Tackling the Gradient Issues in Generative Adversarial Networks

Yanran Li

The Hong Kong Polytechnic University
yanranli.summer@gmail.com

Content

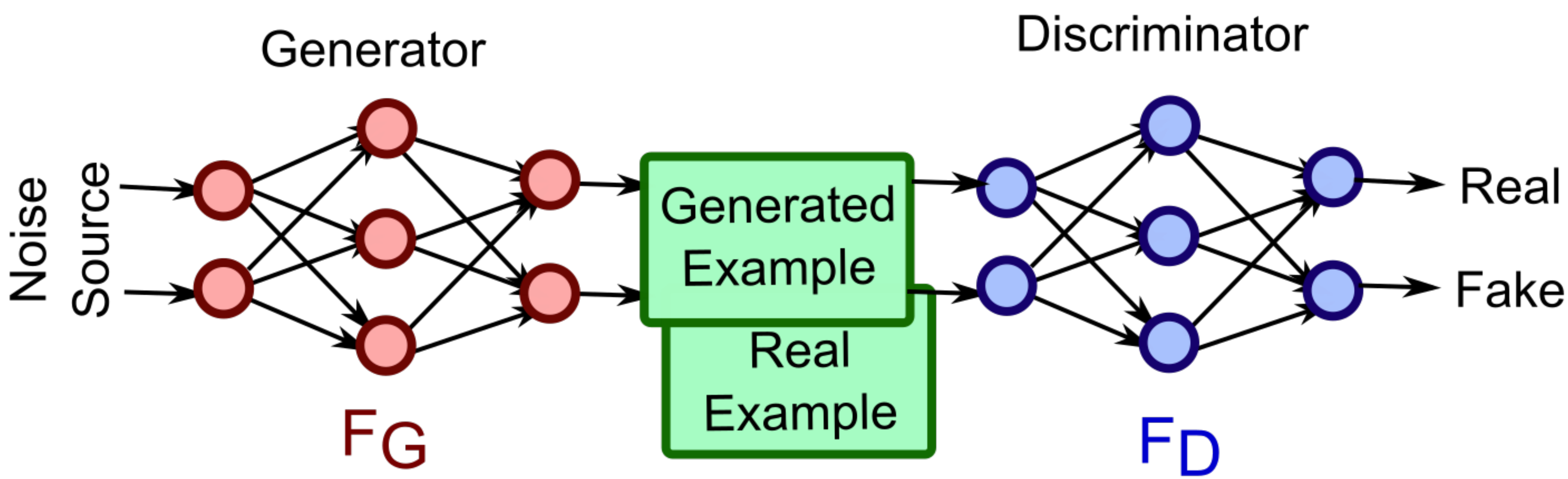
- Generative Adversarial Networks
  - Basics
  - Difficulties
- Solution 1: Encoder-incorporated
  - Mode Regularized GANs
  - Energy-based GANs, InfoGAN, etc.
  - *Noisy Input
- Solution 2: Wasserstein Distance
  - Wasserstein GANs and Improved Training of Wasserstein GANs

Content

- **Generative Adversarial Networks**
  - Basics
  - Difficulties
- Solution 1: Encoder-incorporated
  - Mode Regularized GANs
  - Energy-based GANs, InfoGAN, etc.
  - *Noisy Input
- Solution 2: Wasserstein Distance
  - Wasserstein GANs and Improved Training of Wasserstein GANs

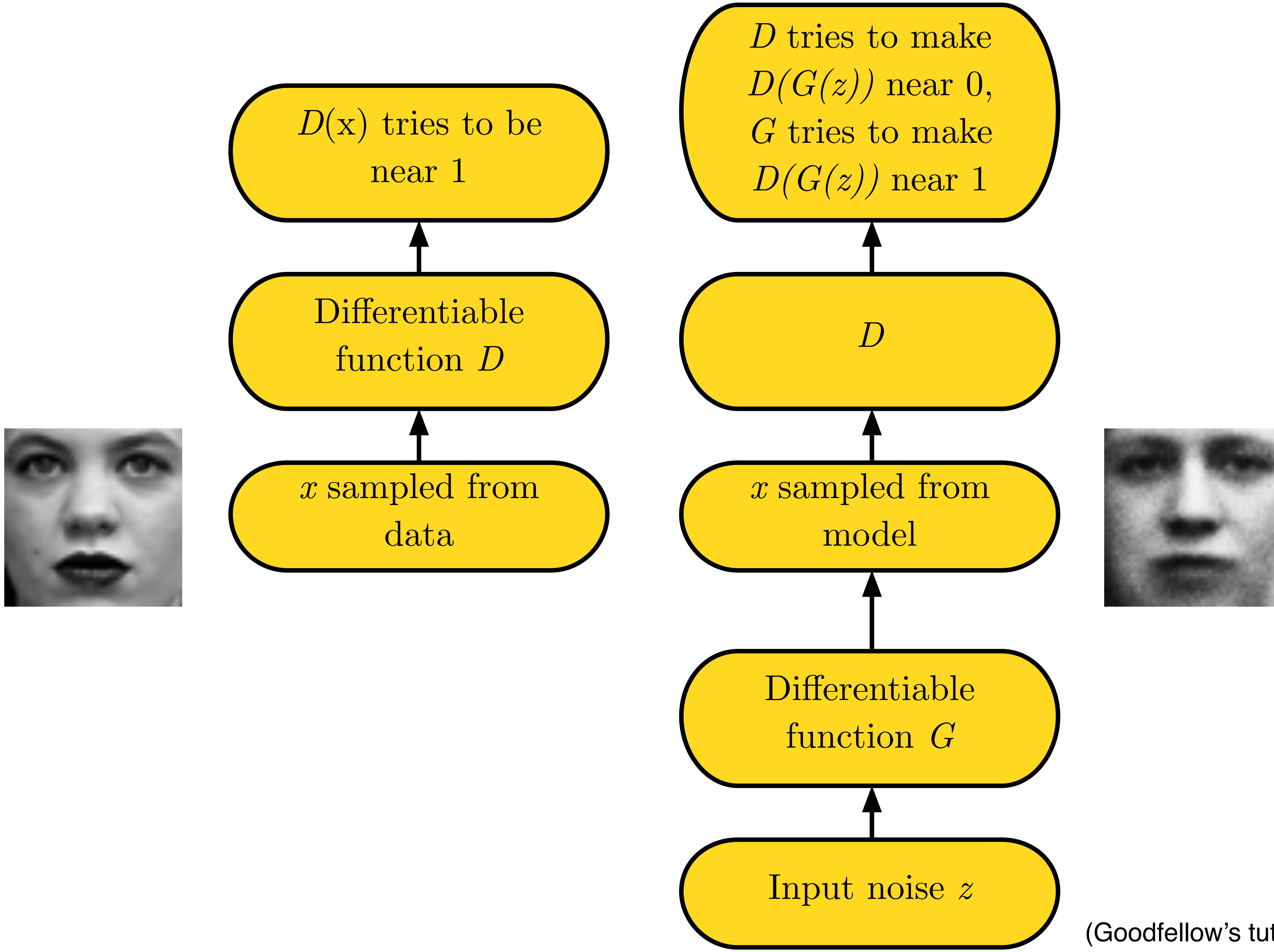
Generative Adversarial Networks

- A min-max game between two components: a generator ***G*** and a discriminator ***D***



(Nicholas Gutenberg's blog)

GANs Framework



(Goodfellow's tutorial)

Objectives for GAN

- The objective of ***D:***

$$L(D, g_\theta) = \mathbb{E}_{x\sim\mathbb{P}_r}[\log D(x)] + \mathbb{E}_{x\sim\mathbb{P}_g}[\log(1 - D(x))]$$

- The objective of ***G:***
  - the original: $\mathbb{E}_{z\sim p(z)}[\log(1 - D(g_\theta(z)))]$
  - the alternative: $\mathbb{E}_{z\sim p(z)}\left[-\log D(g_\theta(z))\right]$
  - *Why alternative?*

Difficulty 1

- using the original form of the objective of ***G***

$$\mathbb{E}_{z\sim p(z)}[\log(1 - D(g_\theta(z)))]$$

will result in gradient vanishing issue of ***D*** for ***G*** because *intuitively*, at the very early phase of training, ***D*** is very easy to be confident in detecting ***G***, so ***D*** will output almost always 0

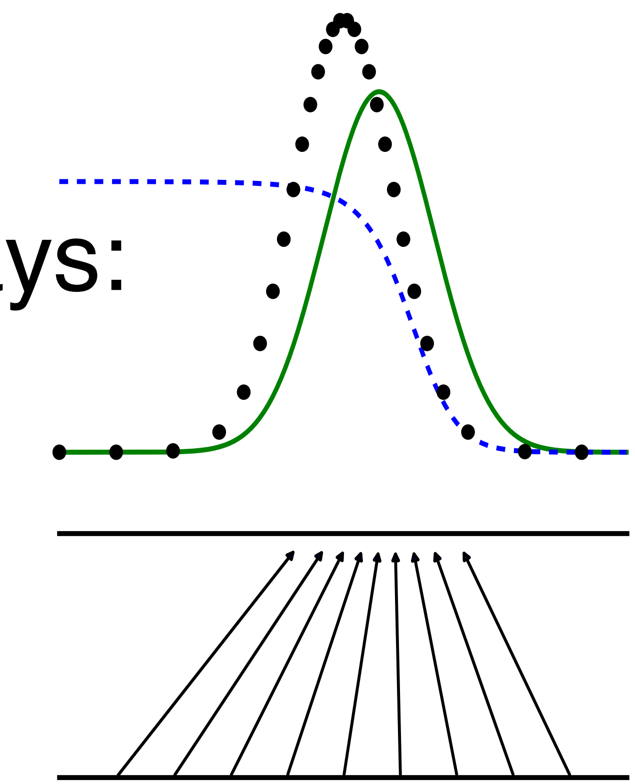
Difficulty 1

- using the original form of the objective of ***G***

$$\mathbb{E}_{z \sim p(z)}[\log(1 - D(g_\theta(z)))]$$

will result in gradient vanishing issue of ***D*** for ***G*** because *theoretically*, when ***D*** is *optimal*, minimizing the loss is equal to minimizing the *JS divergence* (Arjovsky & Bottou, 2017)

Difficulty 1



- The optimal D for any P_r and P_g is always:

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)}$$

and that $$L(D^*, g_\theta) = 2JSD(\mathbb{P}_r \| \mathbb{P}_g) - 2\log 2$$

so, when ***D*** is *optimal*, minimizing the loss is equal to minimizing the *JS divergence* (Arjovsky & Bottou, 2017)

(Goodfellow et al., 2014)

Difficulty 1

- when:

$$L(D^*, g_\theta) = 2JSD(\mathbb{P}_r \| \mathbb{P}_g) - 2\log 2$$

- The JS divergence for the two distributions P_r and P_g is (almost) always log2 because P_r and P_g hardly can overlap (Arjovsky & Bottou, 2017)

- This results in vanishing gradient in theory!

The alternative objective

- The alternative objective of ***G:***

$$\mathbb{E}_{z\sim p(z)}\left[-\log D(g_\theta(z))\right]$$

- Instead of minimizing, let ***G*** maximize the log-probability of the discriminator being mistaken

- It is heuristically motivated that generator can still learn even when discriminator successfully rejects all generator samples, but not theoretically guaranteed

(Goodfellow's tutorial)

Difficulty 2

- using the alternative form of the objective of ***G***

$$\mathbb{E}_{z\sim p(z)}\left[-\log D(g_\theta(z))\right]$$

will result in gradient unstable issue and mode missing problem because *theoretically*, when ***D*** is *optimal*, minimizing the loss is equal to minimizing the *KL divergence* meanwhile maximizing the *JS divergence* (Arjovsky & Bottou, 2017):

$$KL(\mathbb{P}_{g_\theta}\|\mathbb{P}_r) - 2JSD(\mathbb{P}_{g_\theta}\|\mathbb{P}_r)]$$

Difficulty 2

- minimizing the *KL divergence* meanwhile maximizing the *JS divergence* is crazy:

$$KL(\mathbb{P}_{g_\theta}\|\mathbb{P}_r) - 2JSD(\mathbb{P}_{g_\theta}\|\mathbb{P}_r)]$$

- which results in gradient unstable issue

Difficulty 2

- minimizing the *KL divergence* is biased:

$$KL(\mathbb{P}_{g_\theta}||\mathbb{P}_r) - 2JSD(\mathbb{P}_{g_\theta}||\mathbb{P}_r)]$$

- because *KL divergence* is asymmetric, and thus it is not equally treated when ***G*** generates a unreal sample and when ***G*** fails to generate real sample

- Therefore, ***G*** will generate too many few-mode but real samples, a safer strategy

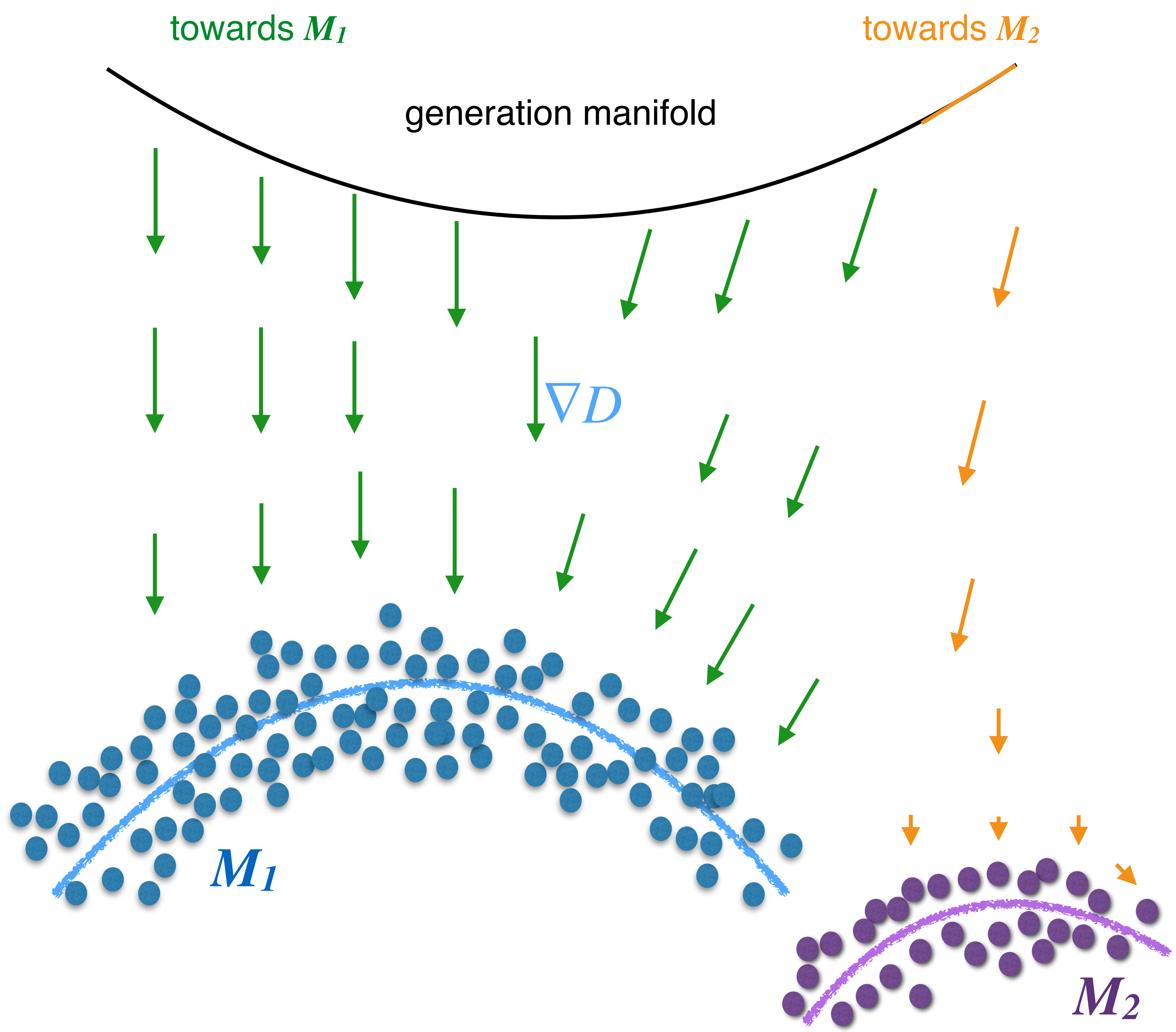
Content

- **Generative Adversarial Networks**
  - Basics
  - Difficulties
- **Solution 1: Encoder-incorporated**
  - Mode Regularized GANs
  - Energy-based GANs, Boundary Equilibrium GANs, etc.
  - *Noisy Input
- Solution 2: Wasserstein Distance
  - Wasserstein GANs and Improved Training of Wasserstein GANs

Solution 1: Encoder-incorporated

- Mode Regularized GANs (Che et al., 2017)
- Tackling the gradient vanishing issue and mode missing problem by incorporating an additional encoder ***E*** to:
  - (1) "enforce" P_r and P_g overlap
  - (2) "build a bridge" between *fake data* and *real data*

Mode Missing Problem

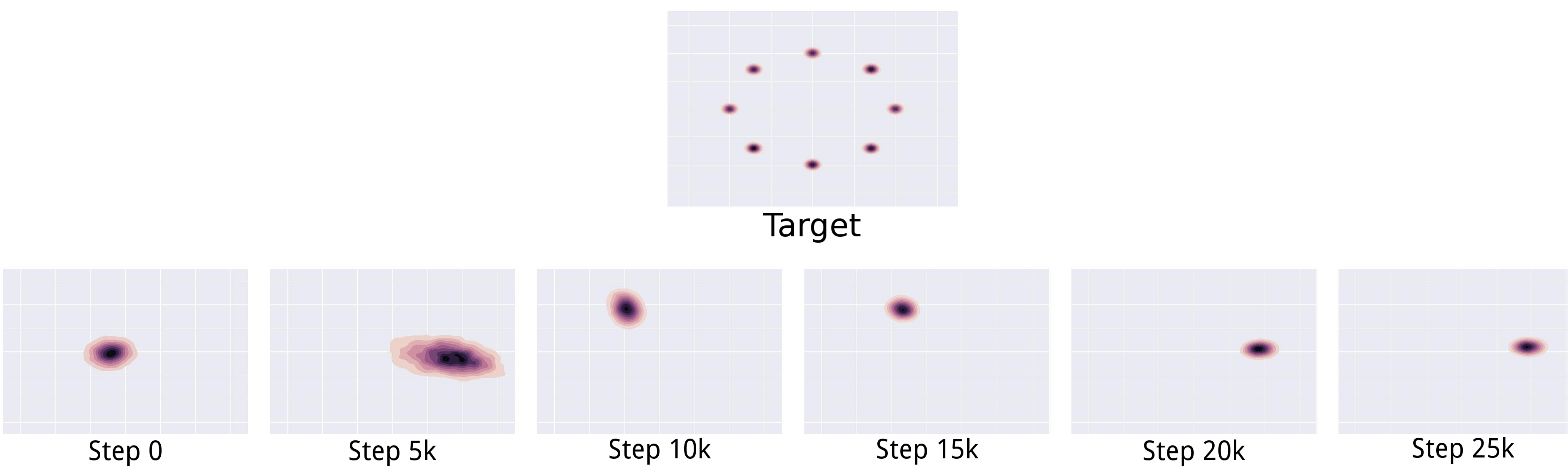


(Che et al., 2017)

Mode Missing Problem

$$\min_G \max_D V(G,D) \neq \max_D \min_G V(G,D)$$

- ***D*** in inner loop: convergence to correct distribution
- ***G*** in inner loop: place all mass on most likely point



(Goodfellow's tutorial)

(Metz et al., 2016)

Mode Regularized GANs

- Regularized GANs
  - for encoder ***E***: $\mathbb{E}_{x \sim p_d}[\lambda_1 d(x, G \circ E(x)) + \lambda_2 \log D(G \circ E(x))]$
  - for generator ***G***:

$$-\mathbb{E}_z[\log D(G(z))] + \mathbb{E}_{x \sim p_d}[\lambda_1 d(x, G \circ E(x)) + \lambda_2 \log D(G \circ E(x))]$$

  - for discriminator ***D***: same as vanilla GAN

Mode Regularized GANs

- Regularized GANs
  - for encoder ***E***: $\mathbb{E}_{x\sim p_d}[\lambda_1 d(x, G\circ E(x)) + \lambda_2 \log D(G\circ E(x))]$
  - for generator ***G***:

$$-\mathbb{E}_z[\log D(G(z))] + \mathbb{E}_{x\sim p_d}[\lambda_1 d(x, G\circ E(x)) + \lambda_2 \log D(G\circ E(x))]$$

  - for discriminator ***D***: same as vanilla GAN

- But it still suffers from gradient vanishing!
- because ***D*** is still comparing between real data and fake data

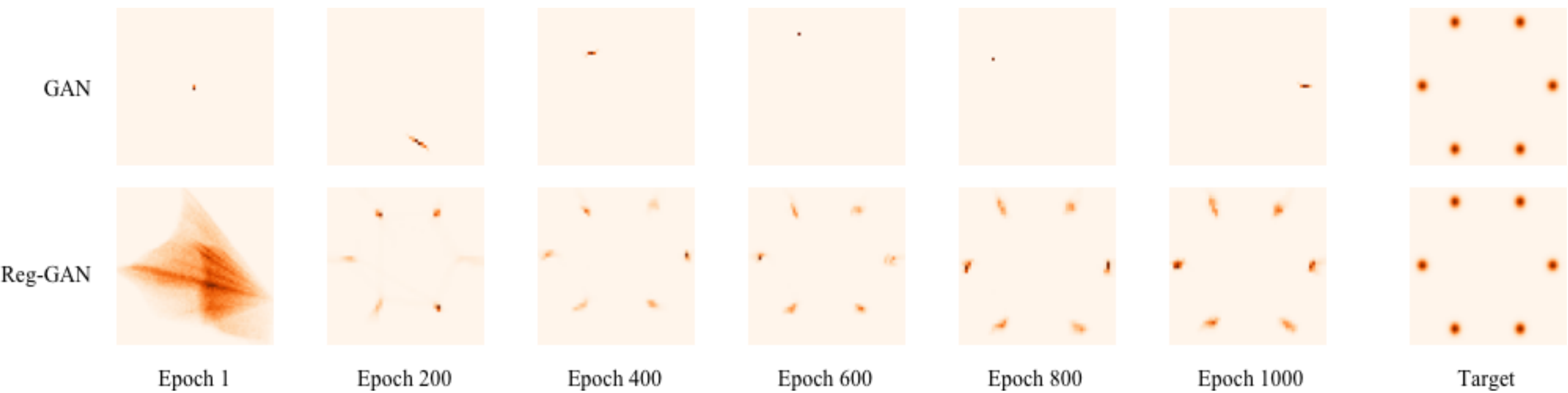
Mode Regularized GANs

- Manifold-Diffusion GANs (MDGAN):
  - for encoder ***E***: $\mathbb{E}_{x\sim p_d}[\lambda_1 d(x, G\circ E(x)) + \lambda_2 \log D(G\circ E(x))]$
  - Manifold-step:
    - for generator ***G***: $\lambda \log D_1(G(E(\mathbf{x}_i))) - ||\mathbf{x}_i - G(E(\mathbf{x}_i))||^2$
    - for discriminator ***D***: $\log D_1(\mathbf{x}_i) + \log(1 - D_1(G(E(\mathbf{x}_i))))$
  - Diffusion-step:
    - for generator ***G***: $\log D_2(G(\mathbf{z}_i))$
    - for discriminator ***D***: $\log D_2(G(E(\mathbf{x}_i))) + \log(1 - D_2(\mathbf{z}_i))$

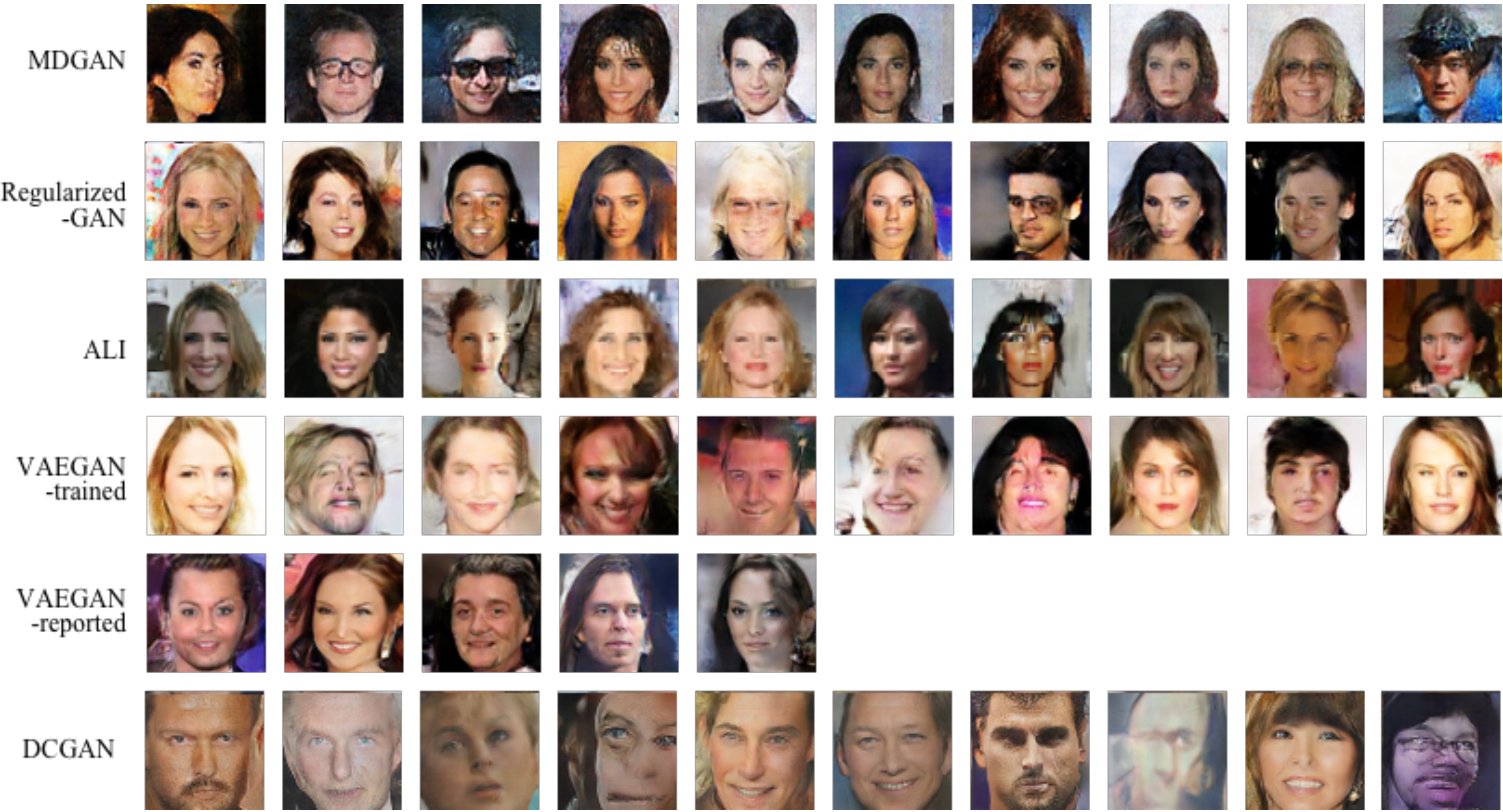
Mode Regularized GANs

- Manifold-Diffusion GANs (MDGAN):
  - for encoder ***E***: $\mathbb{E}_{x\sim p_d}[\lambda_1 d(x, G\circ E(x)) + \lambda_2 \log D(G\circ E(x))]$
  - Manifold-step:
    - for generator ***G***: $\lambda \log D_1(G(E(\mathbf{x}_i))) - ||\mathbf{x}_i - G(E(\mathbf{x}_i))||^2$
    - for discriminator ***D***: $\log D_1(\mathbf{x}_i) + \log(1 - D_1(G(E(\mathbf{x}_i))))$
  - Diffusion-step:
    - for generator ***G***: $\log D_2(G(\mathbf{z}_i))$
    - for discriminator ***D***: $\log D_2(G(E(\mathbf{x}_i))) + \log(1 - D_2(\mathbf{z}_i))$
- ***D*** is firstly comparing between real data and the encoded data — much harder!

Mode Regularized GANs



Mode Regularized GANs

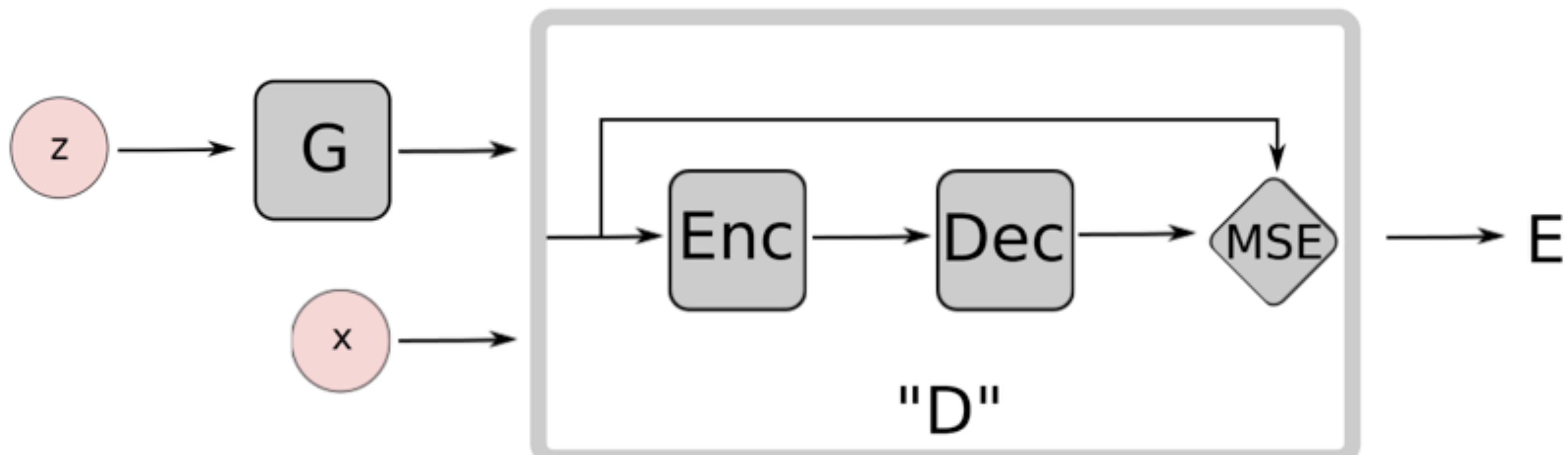


Solution 1: Encoder-incorporated

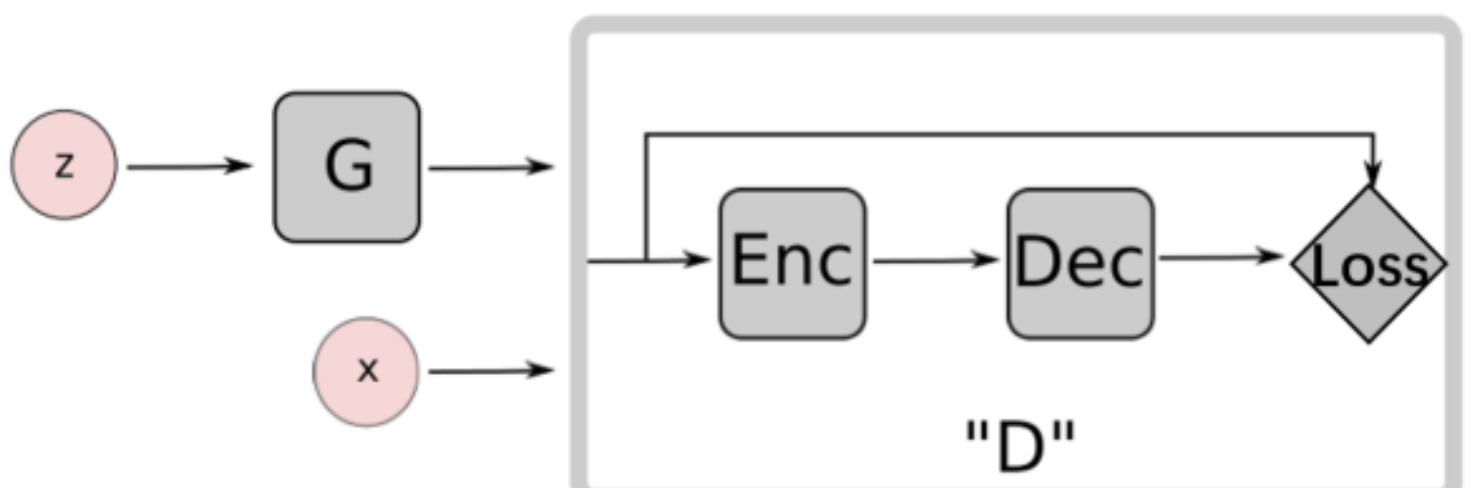
- Mode Regularized GANs (Che et al., 2017)
- Energy-based GANs (Zhao et al., 2017)
- Boundary Equilibrium GANs (Berthelot et al., 2017)
- etc.

Solution 1: Encoder-incorporated

- Energy-based GANs (Zhao et al., 2017)



- Boundary Equilibrium GANs (Berthelot et al., 2017)



Solution 1: **Noisy Input*

- Add noise to input (both real data and fake data) before passing into D (Arjovsky & Bottou, 2017)
- Add noise to layers in D and G (Zhao et al., 2017)
- Instance Noise (Sønderby et al., 2017)
- All these are indeed “enforcing” P_r and P_g to overlap

Content

- **Generative Adversarial Networks**
  - Basics
  - Difficulties
- Solution 1: Encoder-incorporated
  - Mode Regularized GANs
  - Energy-based GANs, Boundary Equilibrium GANs, etc.
  - *Noisy Input
- **Solution 2: Wasserstein Distance**
  - Wasserstein GANs

Solution 2: Wasserstein Distance

- Wasserstein GANs (Arjovsky et al., 2017)
- Wasserstein-1 Distance (Earth-Mover Distance):

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

Solution 2: Wasserstein Distance

- Wasserstein GANs (Arjovsky et al., 2017)
- Wasserstein-1 Distance (Earth-Mover Distance):

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

- *Why is it superior to KL and JS divergence?*

Solution 2: Wasserstein Distance

- Wasserstein-1 Distance (Earth-Mover Distance):

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

where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$. Intuitively, $\gamma(x, y)$ indicates how much "mass" must be transported from x to y in order to transform the distributions $\mathbb{P}_r$ into the distribution $\mathbb{P}_g$. The EM distance then is the "cost" of the optimal transport plan.

(Arjovsky et al., 2017)

Solution 2: Wasserstein Distance

- Wasserstein-1 Distance (Earth-Mover Distance):

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

- The distance is shown to have the desirable property that under mild assumptions
  - it is continuous everywhere and
  - differentiable almost everywhere.

(Arjovsky et al., 2017)

Solution 2: Wasserstein Distance

- Wasserstein-1 Distance (Earth-Mover Distance):

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

- The distance is shown to have the desirable property that under mild assumptions
  - *And most importantly, it can reflect the distance of two distributions even if they do not overlap, and thus can provide meaningful gradients*

(Arjovsky et al., 2017)

Solution 2: Wasserstein Distance

- Wasserstein-1 Distance (Earth-Mover Distance):

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

- By applying the Kantorovich-Rubinstein duality (Villani, 2008), Wasserstein GANs becomes:

$$\min_G \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r} \big[D(\boldsymbol{x})\big] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g} \big[D(\tilde{\boldsymbol{x}}))\big]$$

(Arjovsky et al., 2017)

Wasserstein GANs

- This new value function of WGAN gives rise to the additional requirement that the discriminator must lie within in the space of 1-Lipschitz functions:

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \big[D(\boldsymbol{x})\big] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \big[D(\tilde{\boldsymbol{x}}))\big]$$

- in other words, ***D*** is the set of 1-Lipschitz functions

- To explain Lipschitz continuous is beyond today's topic

(Arjovsky et al., 2017)

Wasserstein GANs

- This new value function of WGAN gives rise to the additional requirement that the discriminator must lie within in the space of 1-Lipschitz functions:

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \big[D(\boldsymbol{x})\big] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \big[D(\tilde{\boldsymbol{x}}))\big]$$

- To satisfy this requirement, WGAN enforces the weights of ***D*** lie within a compact space *[-c, c]* by applying weight clipping

(Arjovsky et al., 2017)

Wasserstein GANs

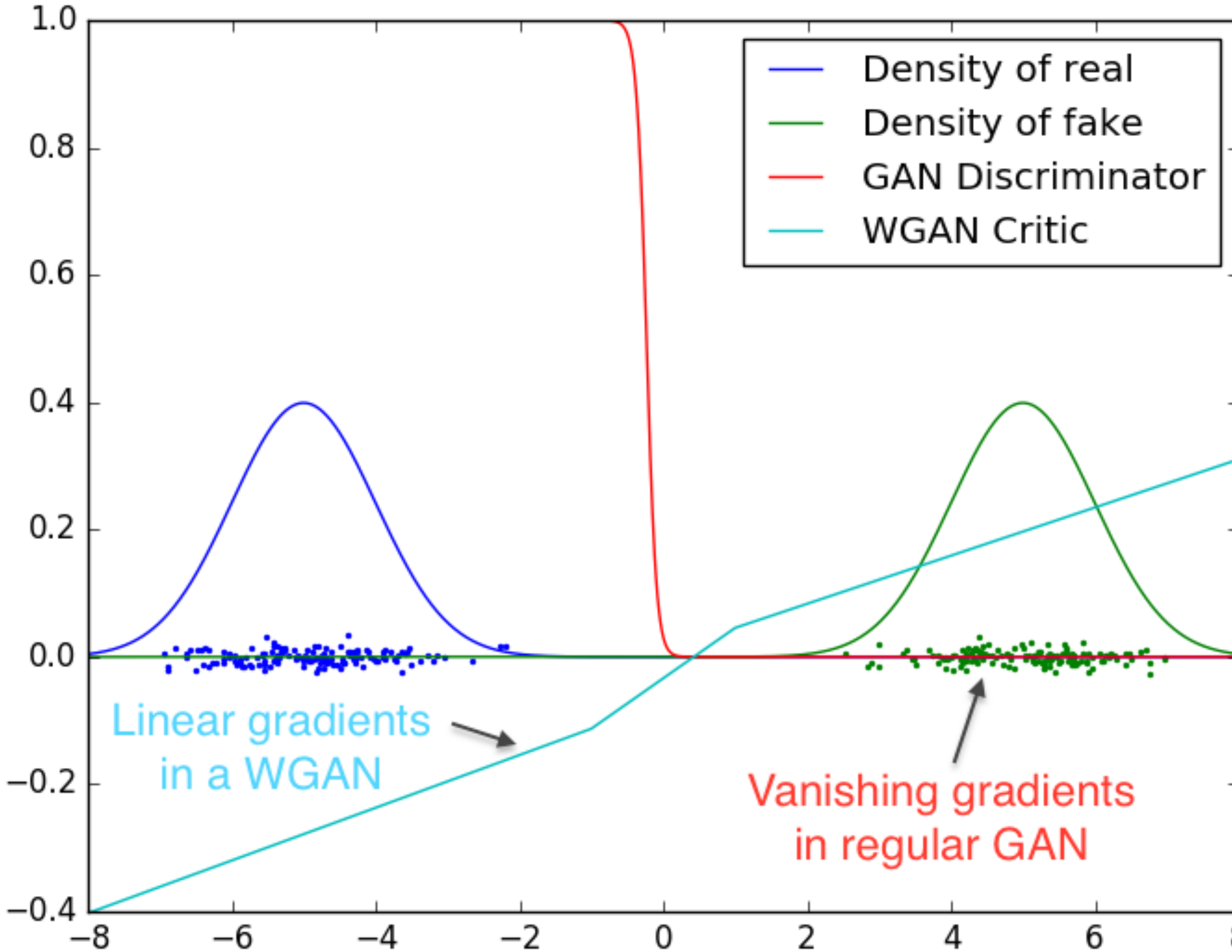
- This new value function of WGAN gives rise to the additional requirement that the discriminator must lie within in the space of 1-Lipschitz functions:

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \big[D(\boldsymbol{x})\big] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \big[D(\tilde{\boldsymbol{x}}))\big]$$

- Also, WGAN removes the sigmoid layer in ***D*** because by using Wasserstein distance, ***D*** in WGAN is doing regression rather than classification

(Arjovsky et al., 2017)

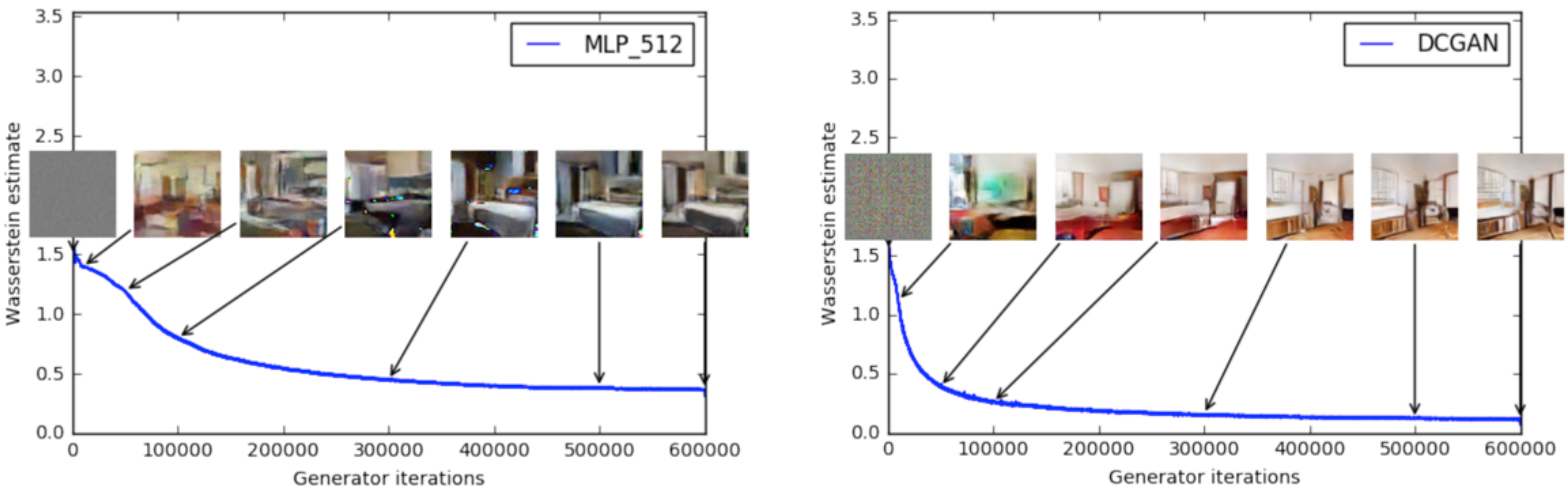
Wasserstein GANs



(Arjovsky et al., 2017)

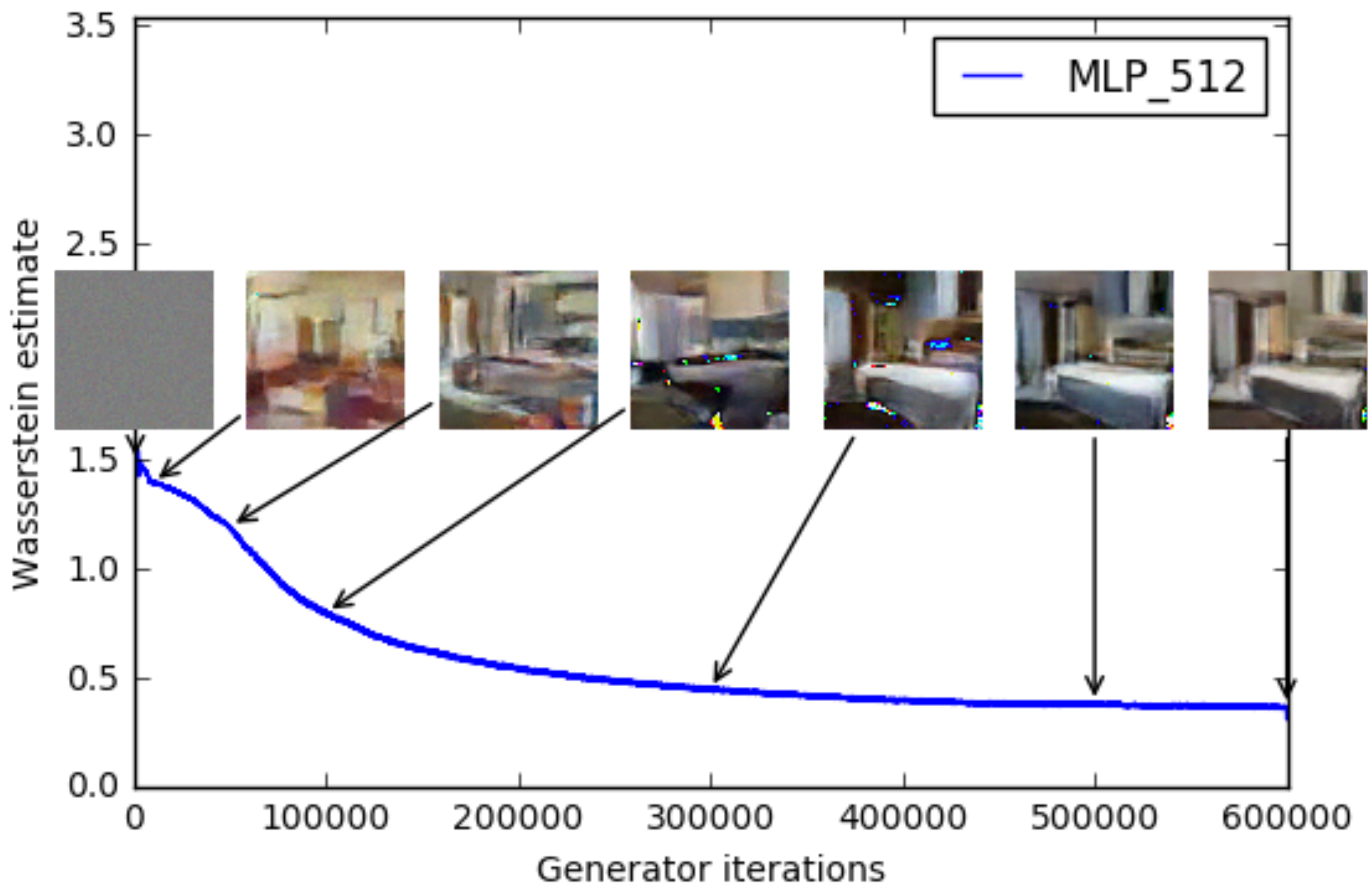
Wasserstein GANs

- This new value function of WGAN seems correlate with the quality of the generated samples:



(Arjovsky et al., 2017)

Wasserstein GANs



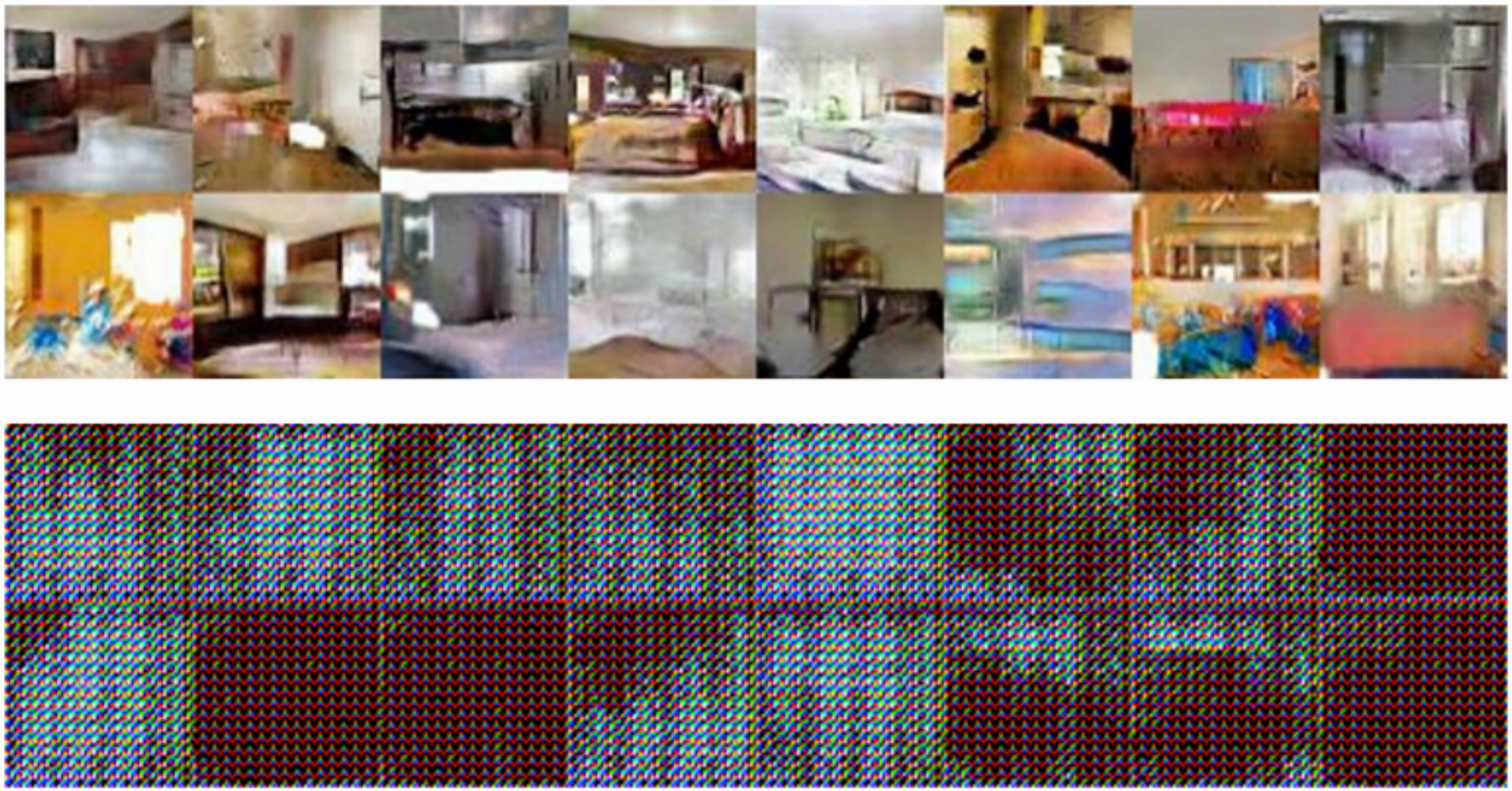
(Arjovsky et al., 2017)

Wasserstein GANs

Top: WGAN with the same DCGAN architecture. *Bottom:* DCGAN

(Arjovsky et al., 2017)

Wasserstein GANs



Top: WGAN with DCGAN architecture, no batch norm. *Bottom:* DCGAN, no batch norm.

(Arjovsky et al., 2017)

Wasserstein GANs

Top: WGAN with MLP architecture. *Bottom:* Standard GAN, same architecture.

(Arjovsky et al., 2017)

Thanks for your attention!
Any questions?

References

- Arjovsky and Bottou, “Towards Principled Methods for Training Generative Adversarial Networks”. ICLR 2017.
- Goodfellow et al., “Generative Adversarial Networks”. ICLR 2014.
- Che et al., “Mode Regularized Generative Adversarial Networks”. ICLR 2017.
- Zhao et al., “Energy-based Generative Adversarial Networks”. ICLR 2017.
- Berthelot et al., “BEGAN: Boundary Equilibrium Generative Adversarial Networks”. arXiv preprint 2017.
- Sønderby, et al., “Amortised MAP Inference for Image Super-Resolution”. ICLR 2017.
- Arjovsky et al., “Wasserstein GANs”. arXiv preprint 2017.
- Villani, Cedric. “Optimal transport: old and new”, volume 338. Springer Science & Business Media, 2008