(W)GANs

Danica J. Sutherland

(from thispersondoesnotexist.com)

MLCC 2019

Generative models

- Start with a bunch of examples: $X_1, \ldots, X_n \sim \mathbb{P}$
- Want a model for the data: $\mathbb{Q} \approx \mathbb{P}$

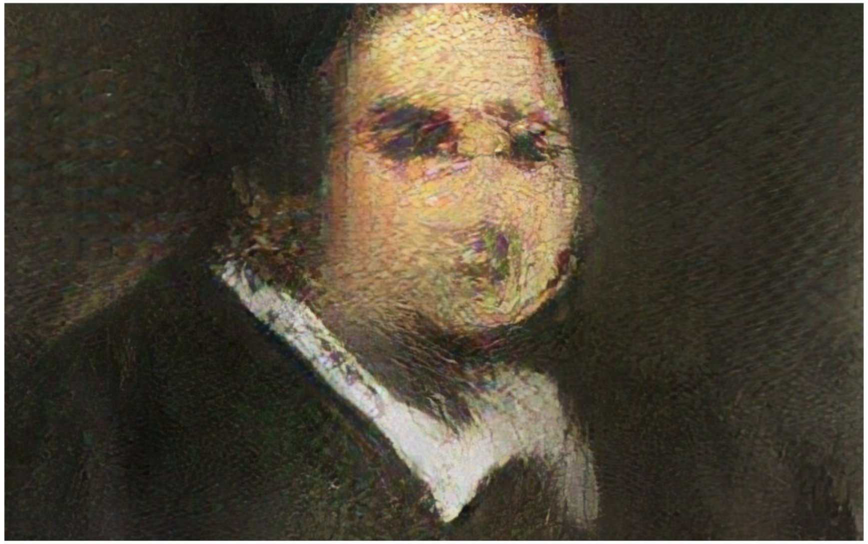
Generative models

- Start with a bunch of examples: $X_1, \ldots, X_n \sim \mathbb{P}$
- Want a model for the data: $\mathbb{Q} \approx \mathbb{P}$
- Might want to do different things with the model:
    - Find most representative data points / modes
    - Find outliers, anomalies, ...
    - Discover underlying structure of the data
    - Impute missing values
    - Use as prior (semi-supervised, machine translation, ...)
    - Produce "more samples"
    - ...

Generative models

- Start with a bunch of examples: $X_1, \ldots, X_n \sim \mathbb{P}$
- Want a model for the data: $\mathbb{Q} \approx \mathbb{P}$
- Might want to do different things with the model:
    - Find most representative data points / modes
    - Find outliers, anomalies, ...
    - Discover underlying structure of the data
    - Impute missing values
    - Use as prior (semi-supervised, machine translation, ...)
    - Produce "more samples"
    - ...

Why produce samples?



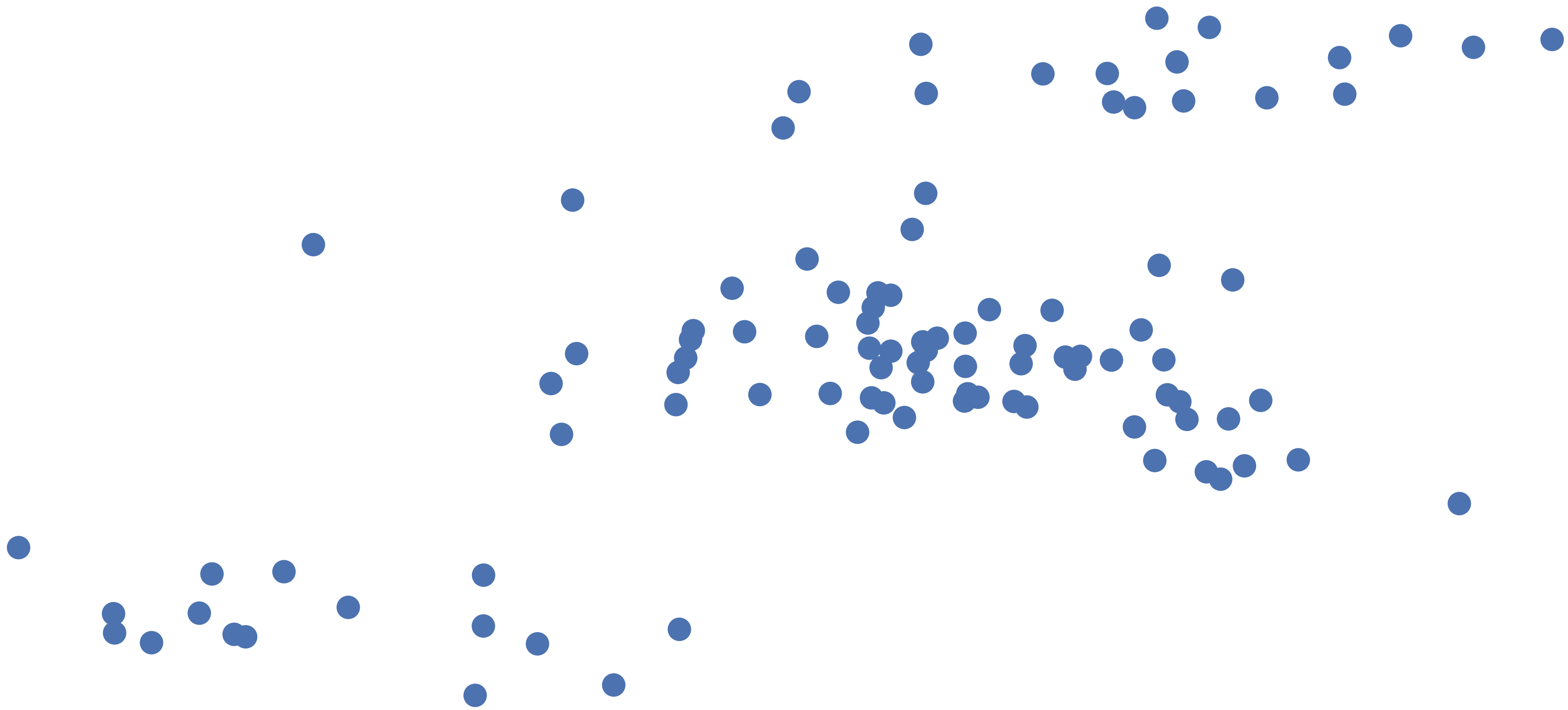
Is artificial intelligence set to become art's next medium?

AI artwork sells for $432,500 — nearly 45 times its high estimate — as Christie's becomes the first auction house to offer a work of art created by an algorithm

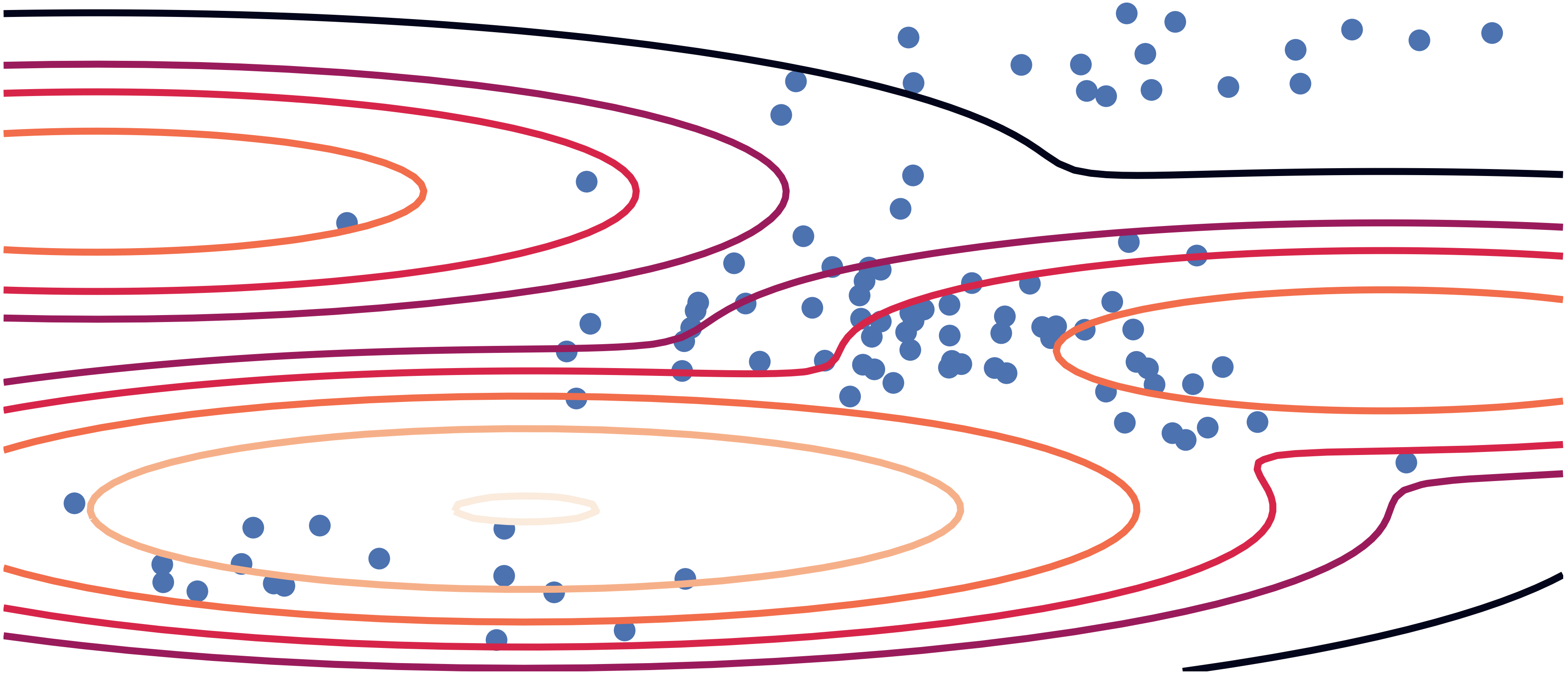
The portrait in its gilt frame depicts a portly gentleman, possibly French and — to judge by his dark frockcoat and plain white collar — a man of the church. The work appears unfinished: the facial features are somewhat indistinct and there are blank areas of canvas. Oddly, the whole composition is displaced slightly to the north-west. A label on the wall states that the sitter is a man named Edmond Belamy, but the giveaway clue as to the origins of the work is the artist's signature at the bottom right. In cursive Gallic script it reads:

$$\min_G \max_D \mathbb{E}_x[\log(D(x))] + \mathbb{E}_z[\log(1 - D(G(z)))]$$

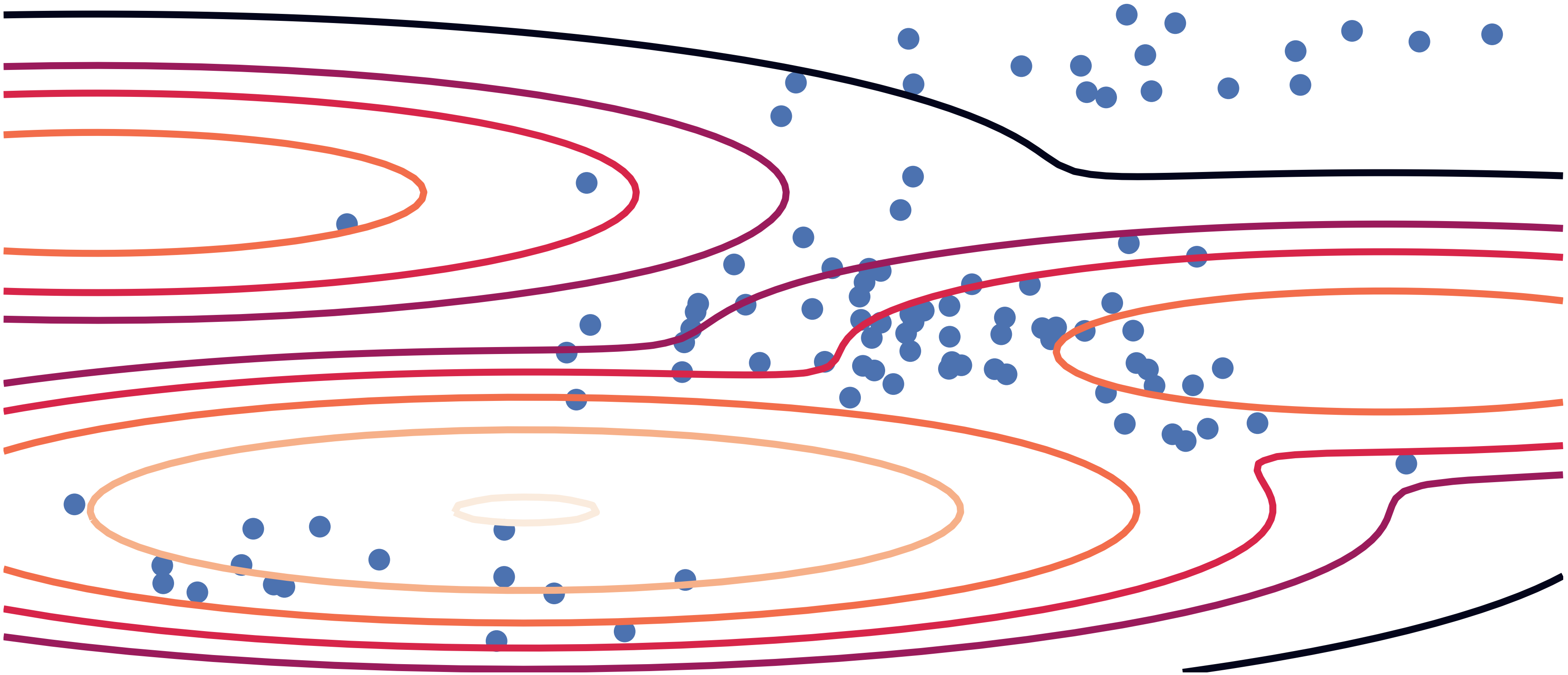
Generative models: a traditional way



Generative models: a traditional way

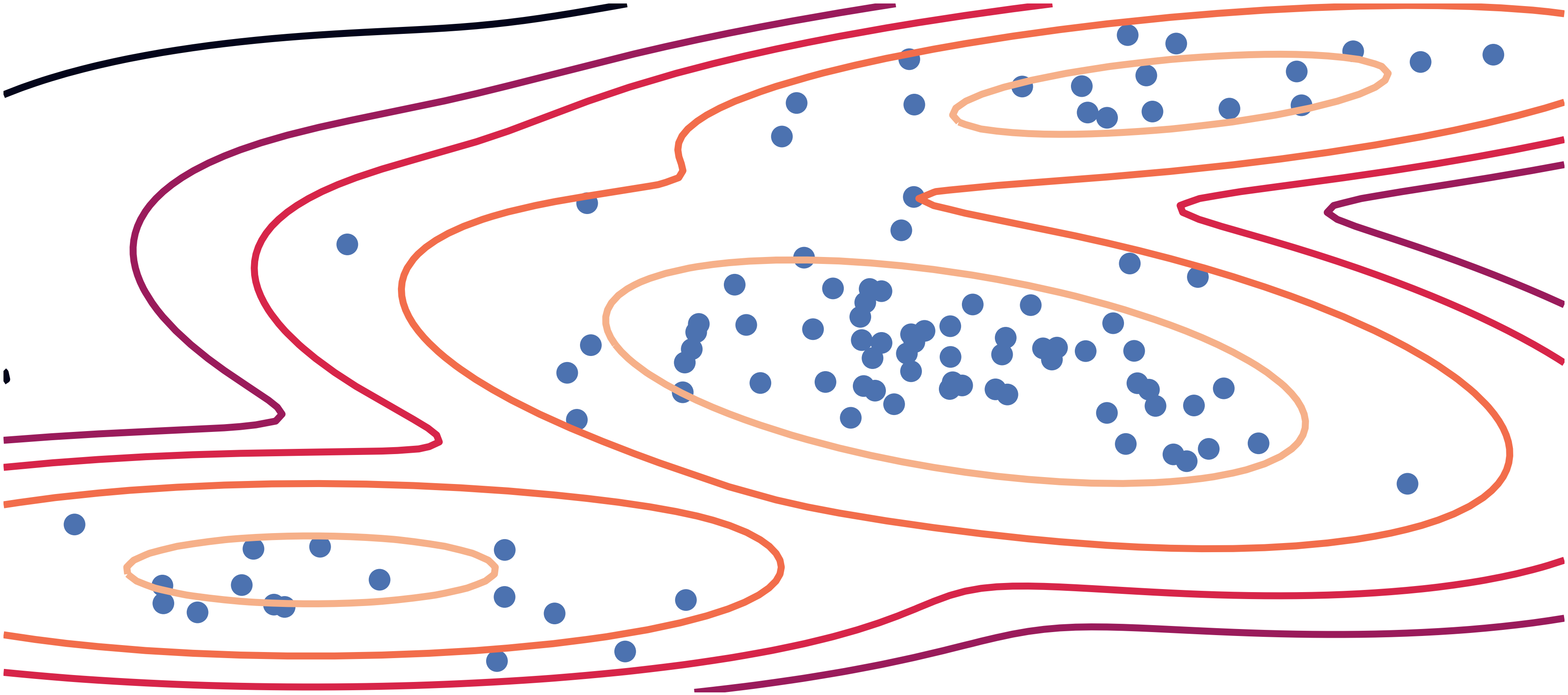


Generative models: a traditional way



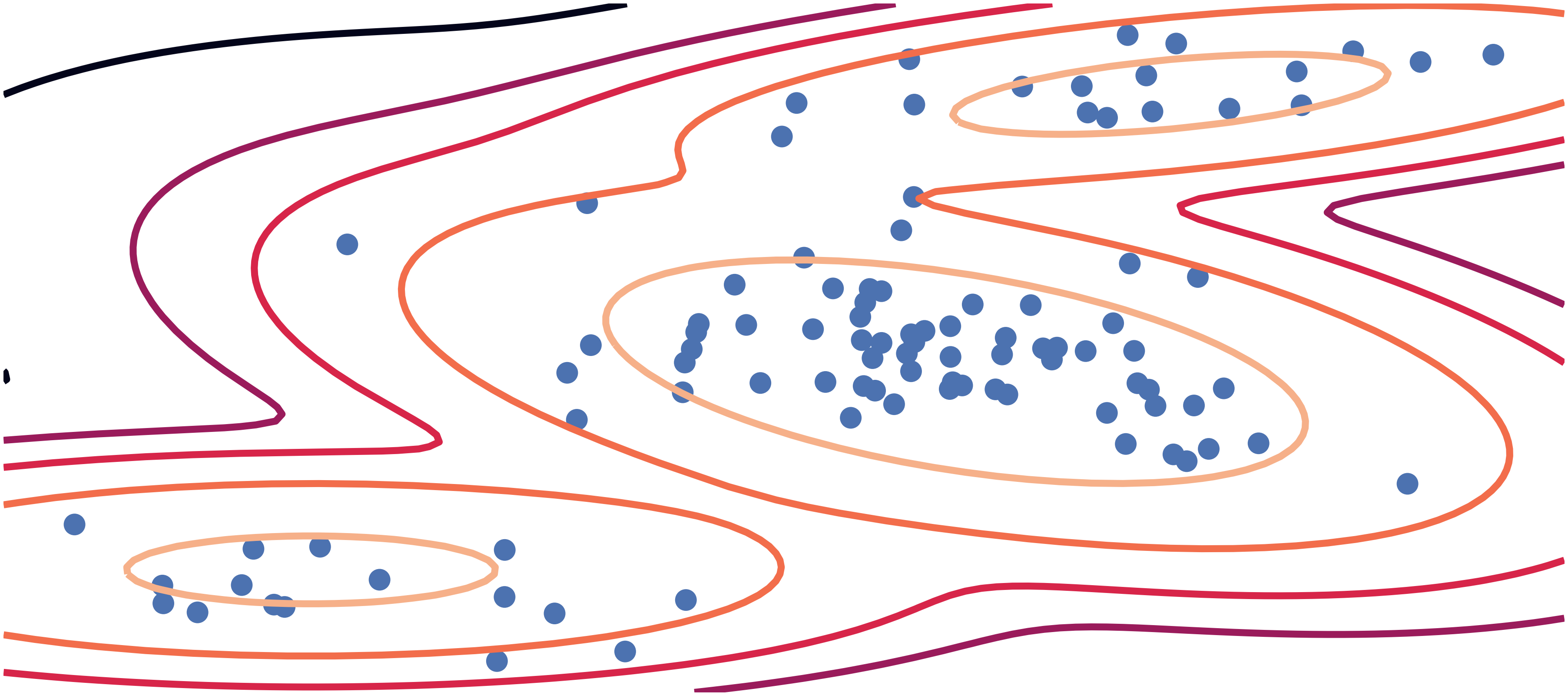
- Maximum likelihood: $\max_\theta \mathbb{E}_{X\sim\mathbb{P}}[\log q_\theta(X)]$

Generative models: a traditional way



- Maximum likelihood: $\max_{\theta} \mathbb{E}_{X \sim \mathbb{P}} [\log q_{\theta}(X)]$

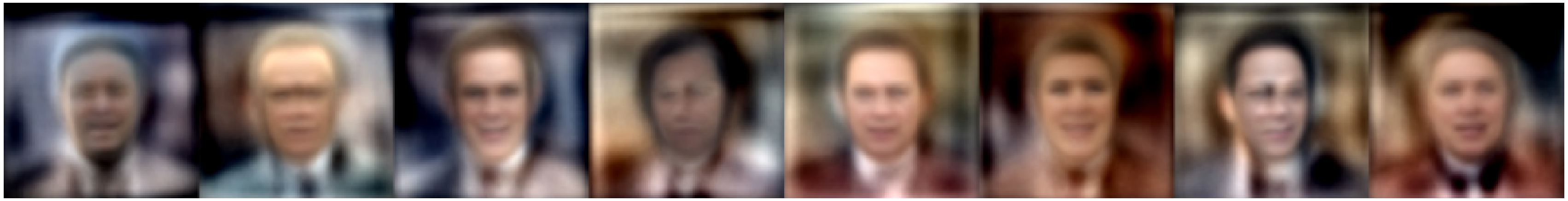
Generative models: a traditional way



- Maximum likelihood: $\max_\theta \mathbb{E}_{X\sim\mathbb{P}}[\log q_\theta(X)]$
- Equivalent: $\min_\theta \mathrm{KL}(\mathbb{P}\|\mathbb{Q}_\theta) = \min_\theta \int p(x) \log \frac{p(x)}{q_\theta(x)} \mathrm{d}x$

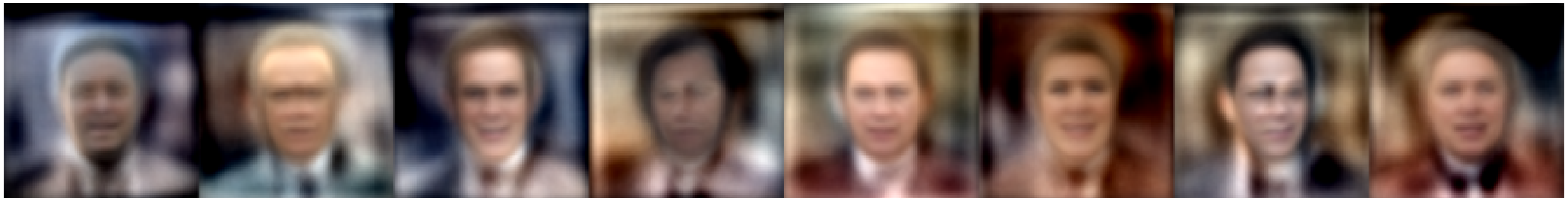
Traditional models for images

- 1987-style generative model of faces (Eigenface via Alex Egg)



Traditional models for images

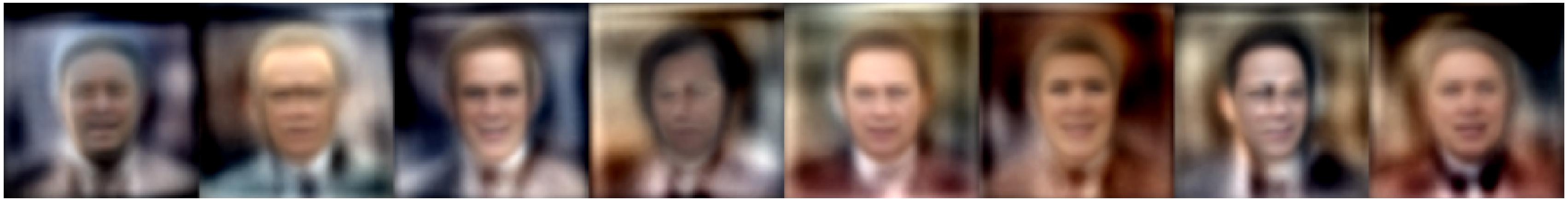
- 1987-style generative model of faces (Eigenface via Alex Egg)



- Can do fancier versions, of course…

Traditional models for images

- 1987-style generative model of faces (Eigenface via Alex Egg)



- Can do fancier versions, of course...
- Usually based on Gaussian noise $\approx L_2$ loss

A hard case for traditional approaches

- One use case of generative models is inpainting [Harry Yang]:

A hard case for traditional approaches

- One use case of generative models is inpainting [Harry Yang]:

A hard case for traditional approaches

- One use case of generative models is inpainting [Harry Yang]:

A hard case for traditional approaches

- One use case of generative models is inpainting [Harry Yang]:

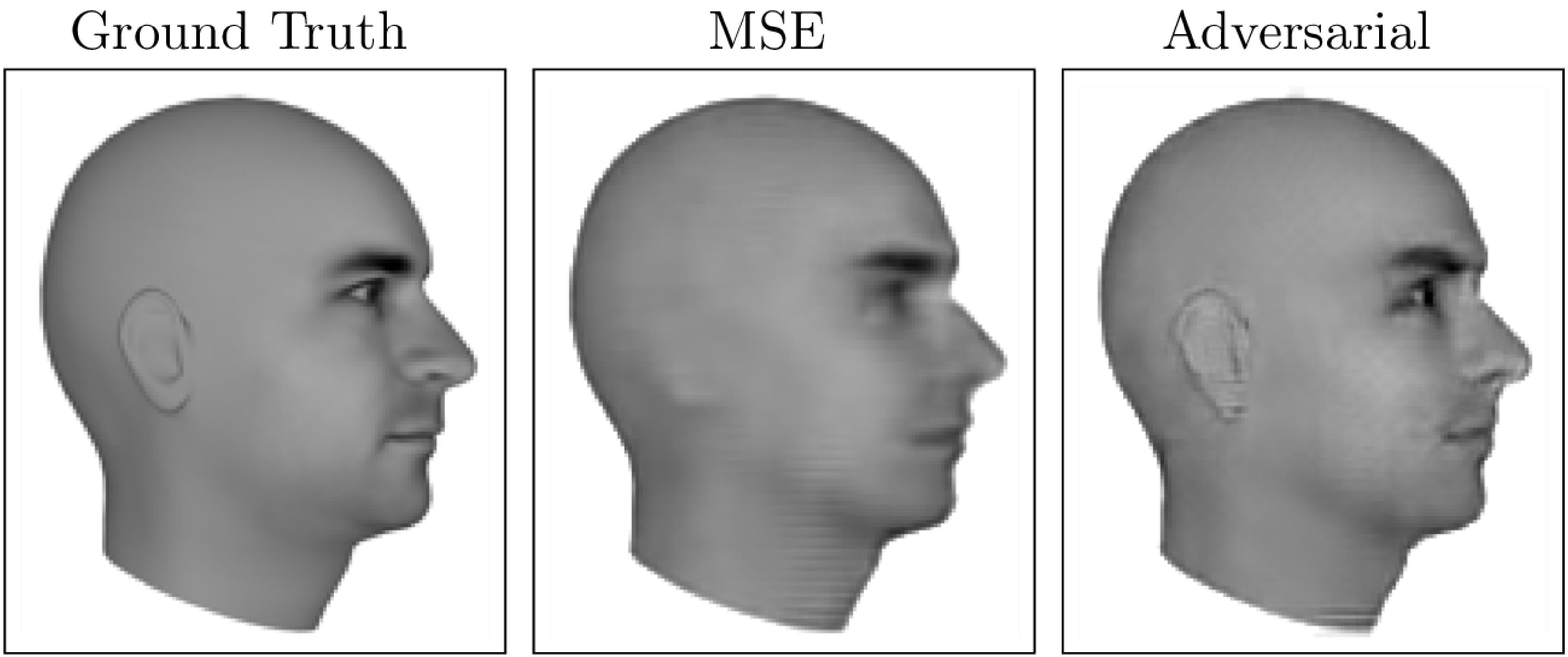
- L_2 loss / Gaussians will pick the *mean* of possibilities

A hard case for traditional approaches

- One use case of generative models is inpainting [Harry Yang]:

- L_2 loss / Gaussians will pick the *mean* of possibilities

Next-frame video prediction



[Lotter+ 2016]

Trick a discriminator [Goodfellow+ NeurIPS-14]

Generator ($\mathbb{Q}_\theta$)

Discriminator

Trick a discriminator [Goodfellow+ NeurIPS-14]

Generator ($\mathbb{Q}_\theta$)

Discriminator

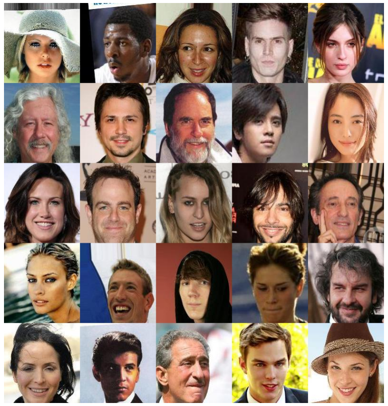
Is this real?

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

Generator ($\mathbb{Q}_\theta$)

Discriminator



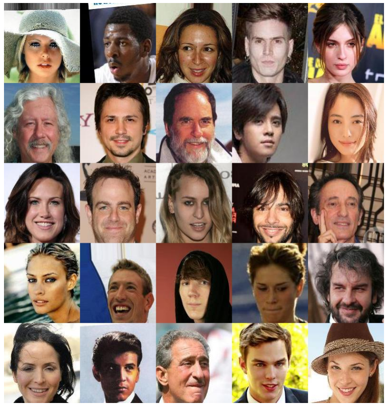
Is this real?

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

Generator ($\mathbb{Q}_\theta$)

Discriminator



Is this real?

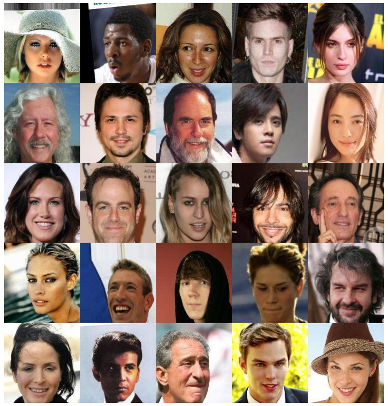
No way! $\mathrm{Pr}(\mathrm{real}) = 0.03$

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

Generator ($\mathbb{Q}_\theta$)

Discriminator



Is this real?

:(I'll try harder...

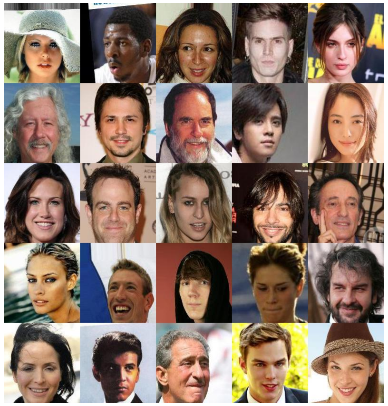
No way! $\mathrm{Pr}(\mathrm{real}) = 0.03$

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

Generator ($\mathbb{Q}_\theta$)

Discriminator



Is this real?

:(I'll try harder...

No way! $\Pr(\text{real}) = 0.03$

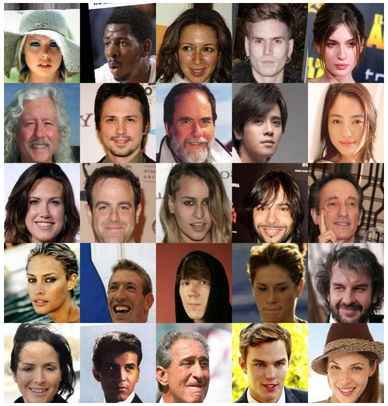
⋮

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

Generator ($\mathbb{Q}_\theta$)

Discriminator



Is this real?

:(I'll try harder...

No way! $\Pr(\text{real}) = 0.03$

⋮

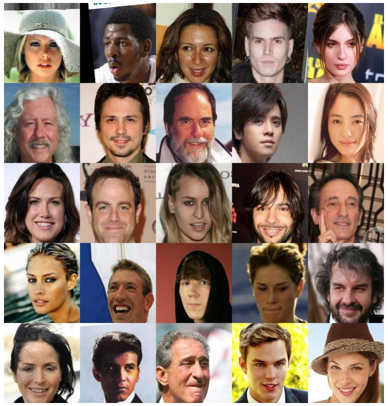
Is this real?

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

Generator ($\mathbb{Q}_\theta$)

Discriminator



Is this real?

No way! $\Pr(\text{real}) = 0.03$

:(I'll try harder...

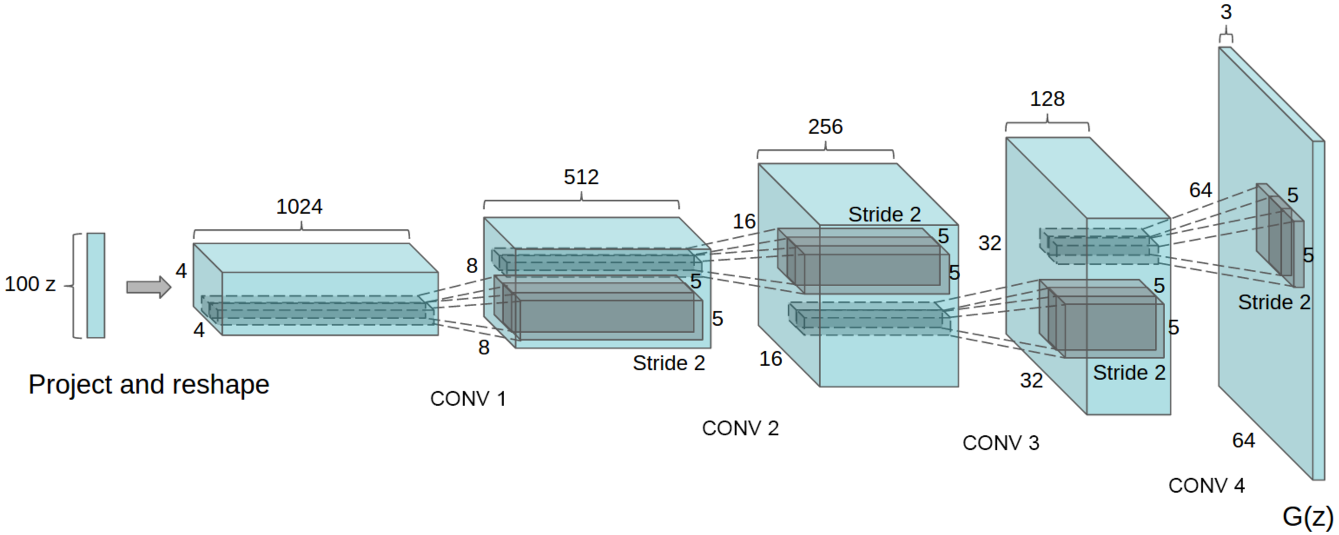
⋮

Is this real?

Umm... $\Pr(\text{real}) = 0.48$

Generator networks

- How to specify $\mathbb{Q}_\theta$?



[Radford+ ICLR-16]

- $Z \sim \mathbb{Z} = \text{Uniform}\left([-1,1]^{100}\right)$
- $G_\theta : [-1,1]^{100} \to \mathcal{X}, G_\theta(Z) \sim \mathbb{Q}_\theta$

GANs in equations

- Tricking the discriminator:

$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathbb{E}_{X \sim \mathbb{P}} [\log D_{\psi}(X)] + \frac{1}{2} \mathbb{E}_{Y \sim \mathbb{Q}_{\theta}} [\log(1 - D_{\psi}(Y))]$$

GANs in equations

- Tricking the discriminator:

$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [\log D_{\psi}(X)] + \frac{1}{2} \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}_{\theta}} [\log(1 - D_{\psi}(Y))]$$

- Using the generator network for $\mathbb{Q}_{\theta}$:

$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [\log D_{\psi}(X)] + \frac{1}{2} \mathop{\mathbb{E}}_{Z \sim \mathbb{Z}} [\log(1 - D_{\psi}(G_{\theta}(Z)))]$$

GANs in equations

- Tricking the discriminator:

$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [\log D_{\psi}(X)] + \frac{1}{2} \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}_{\theta}} [\log(1 - D_{\psi}(Y))]$$

- Using the generator network for $\mathbb{Q}_{\theta}$:

$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [\log D_{\psi}(X)] + \frac{1}{2} \mathop{\mathbb{E}}_{Z \sim \mathbb{Z}} [\log(1 - D_{\psi}(G_{\theta}(Z)))]$$

- Can do alternating gradient descent!

Original paper's results [Goodfellow+ NeurIPS-14]

DCGAN results [Radford+ ICLR-16]

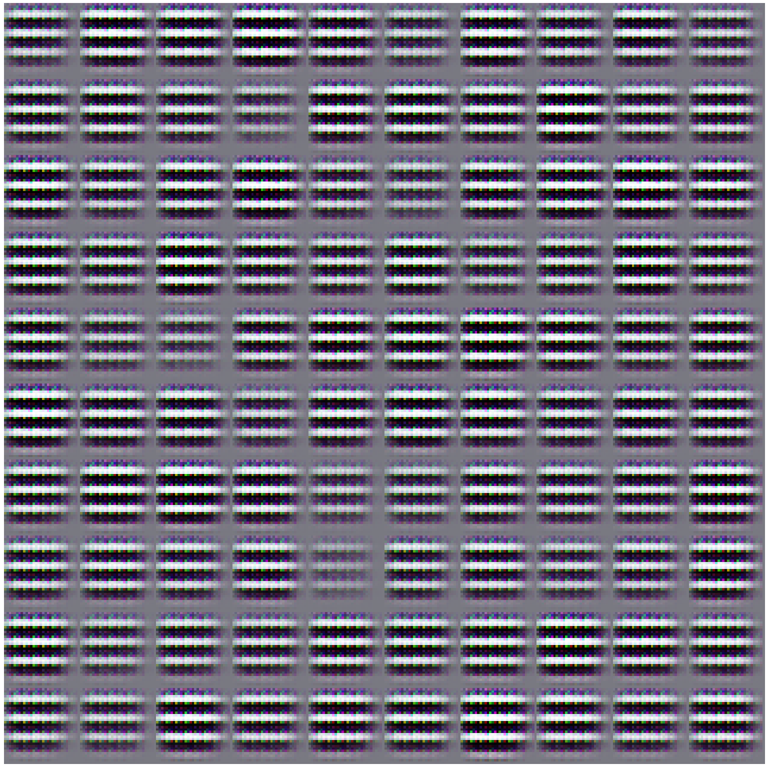
Training instability

Running code from [Salimans+ NeurIPS-16]:

Run 1, epoch 1

Training instability

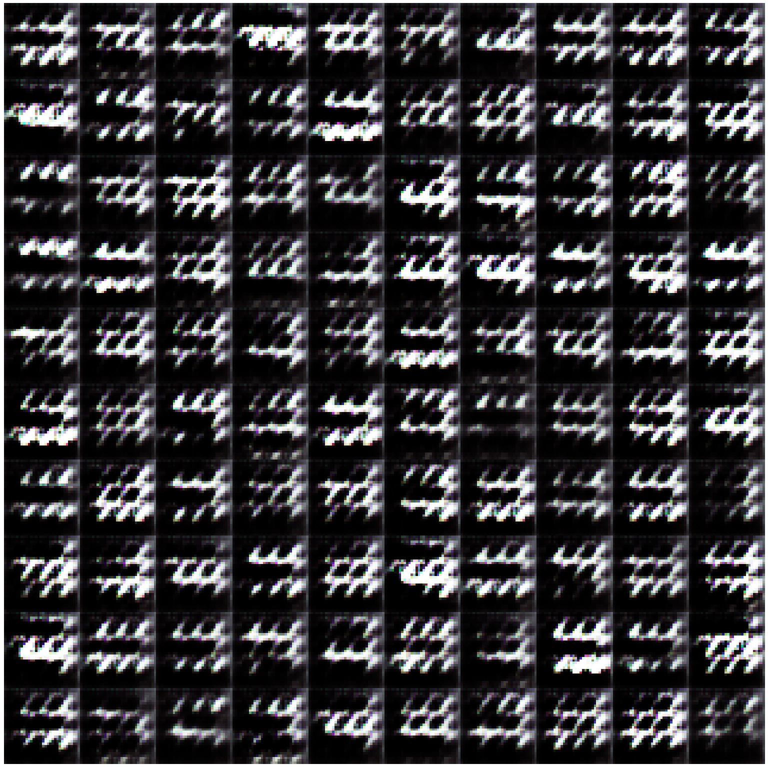
Running code from [Salimans+ NeurIPS-16]:



Run 1, epoch 2

Training instability

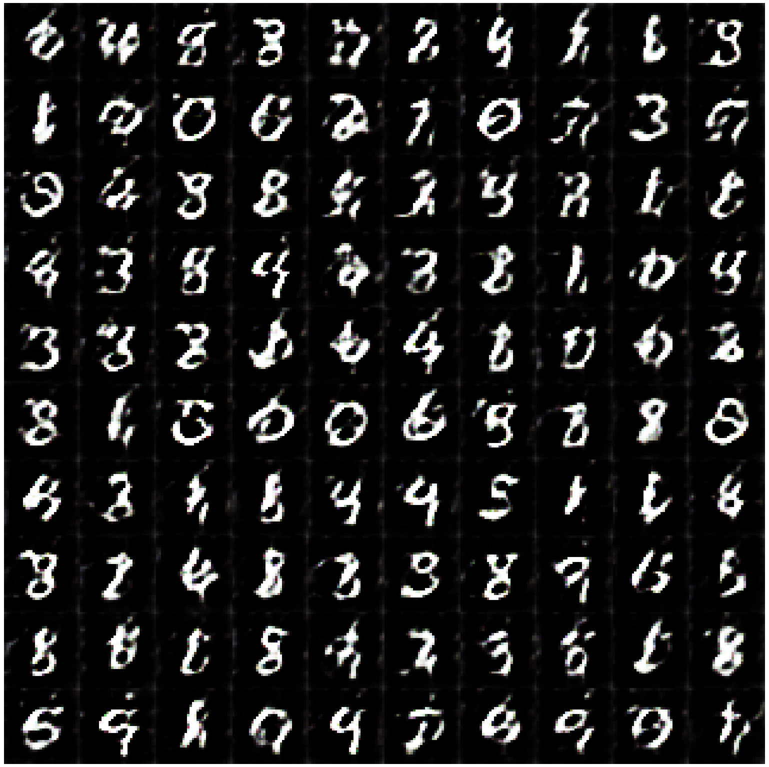
Running code from [Salimans+ NeurIPS-16]:



Run 1, epoch 3

Training instability

Running code from [Salimans+ NeurIPS-16]:



Run 1, epoch 4

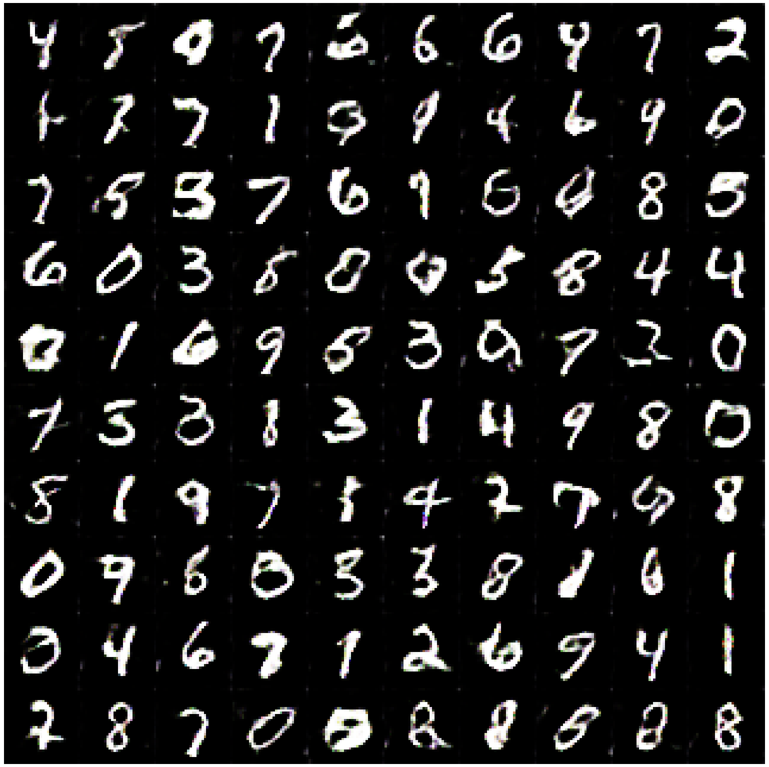
Training instability

Running code from [Salimans+ NeurIPS-16]:

Run 1, epoch 5

Training instability

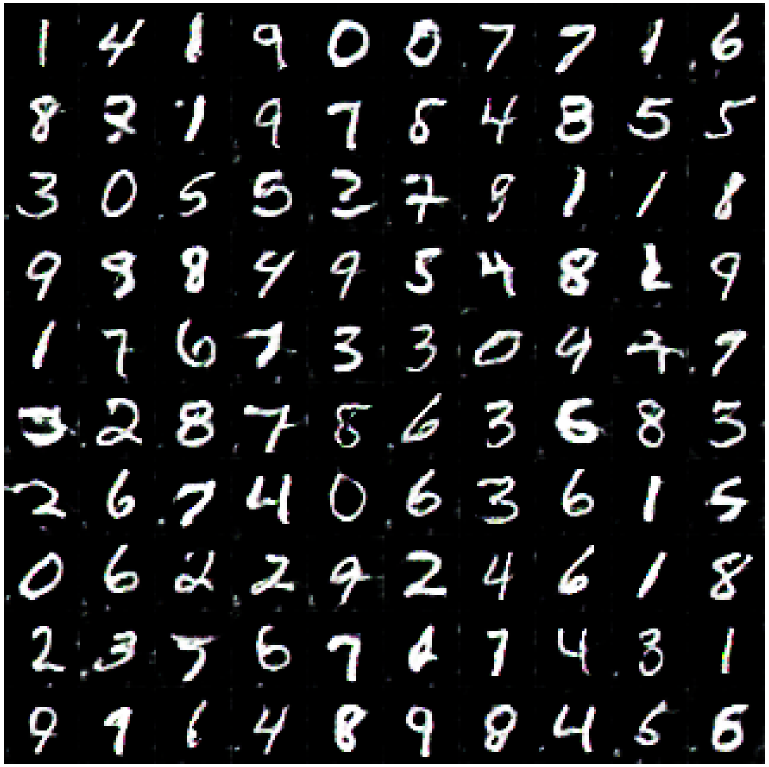
Running code from [Salimans+ NeurIPS-16]:



Run 1, epoch 6

Training instability

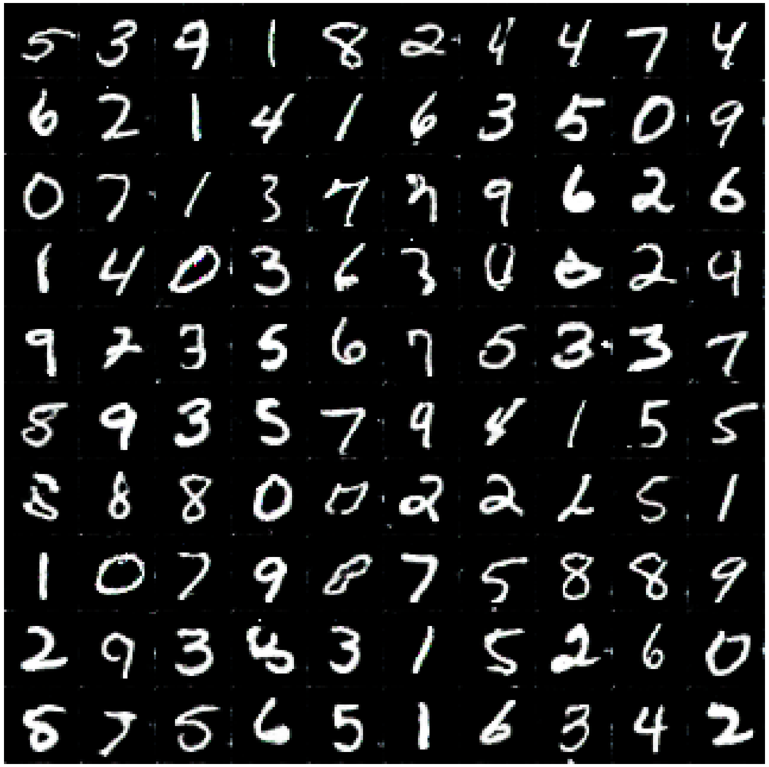
Running code from [Salimans+ NeurIPS-16]:



Run 1, epoch 11

Training instability

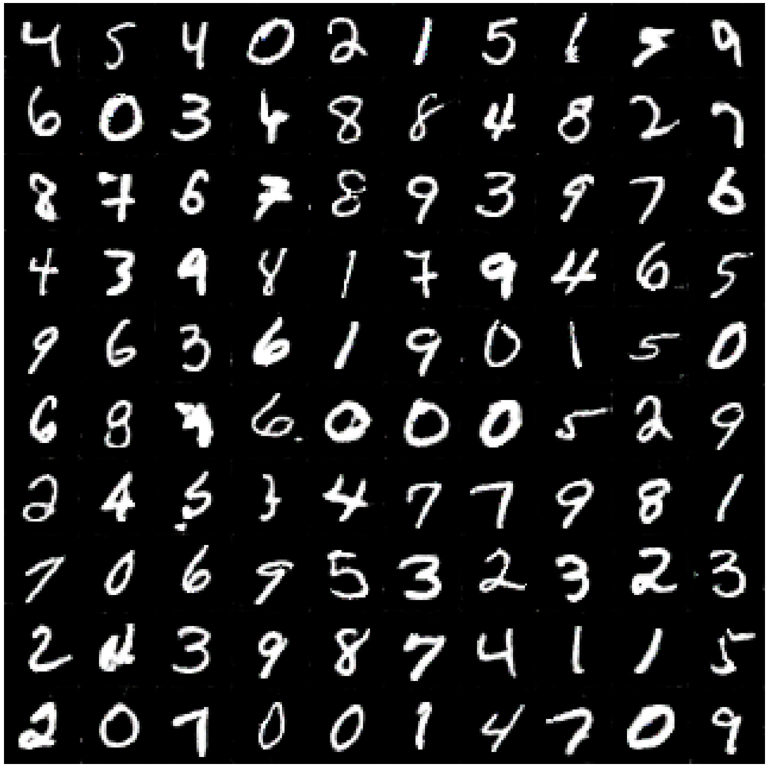
Running code from [Salimans+ NeurIPS-16]:



Run 1, epoch 501

Training instability

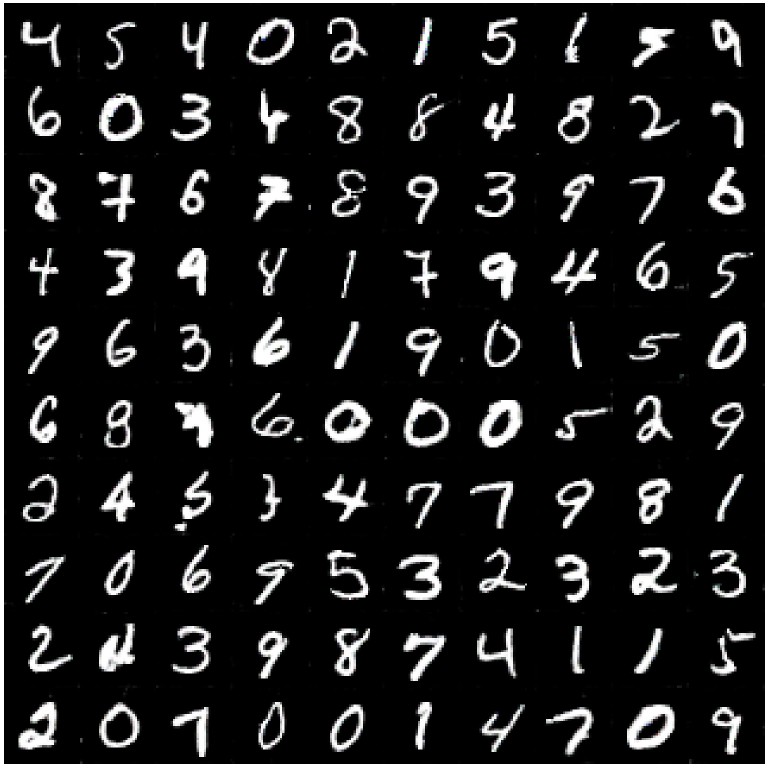
Running code from [Salimans+ NeurIPS-16]:



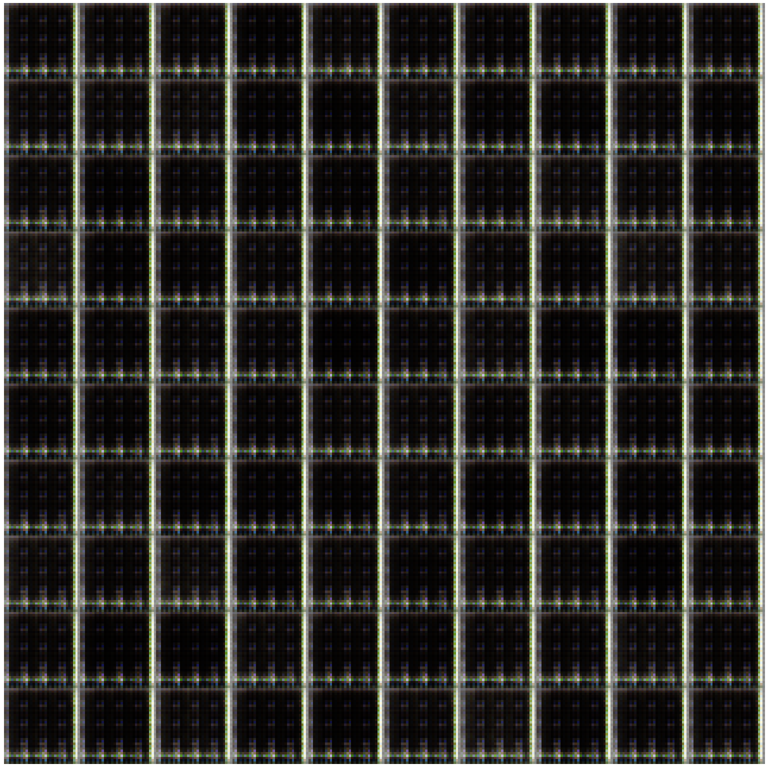
Run 1, epoch 900

Training instability

Running code from [Salimans+ NeurIPS-16]:



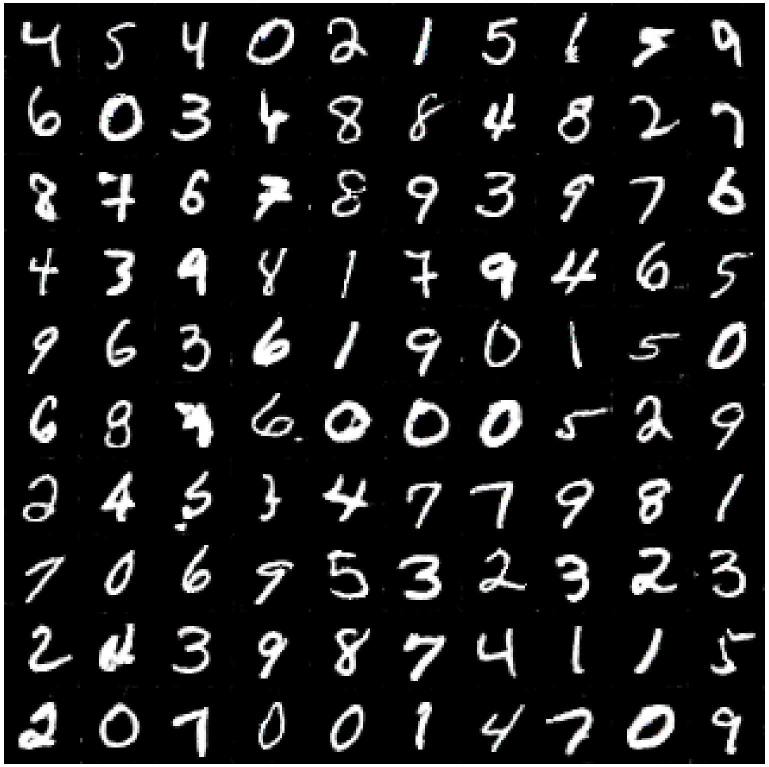
Run 1, epoch 900



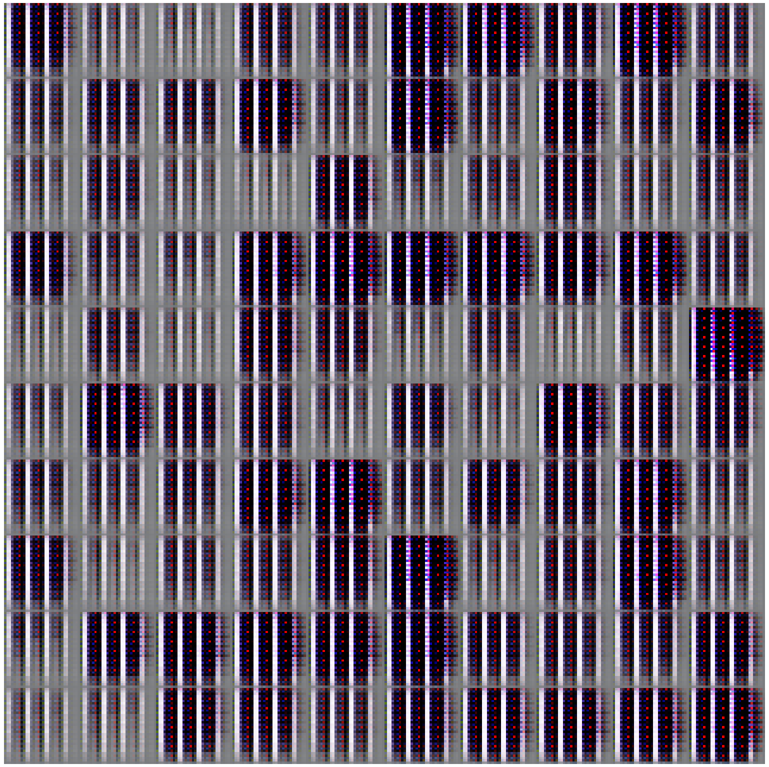
Run 2, epoch 1

Training instability

Running code from [Salimans+ NeurIPS-16]:



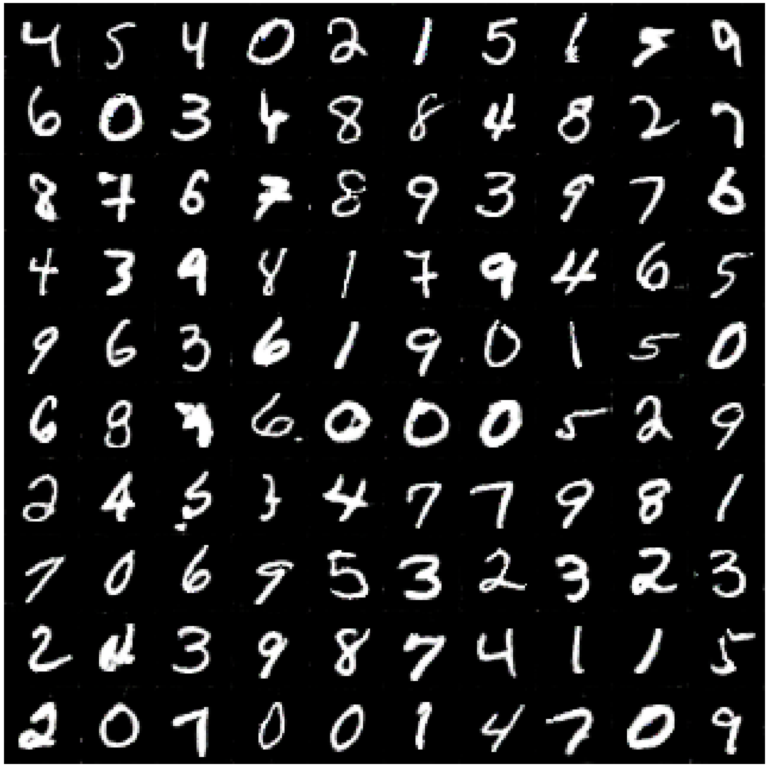
Run 1, epoch 900



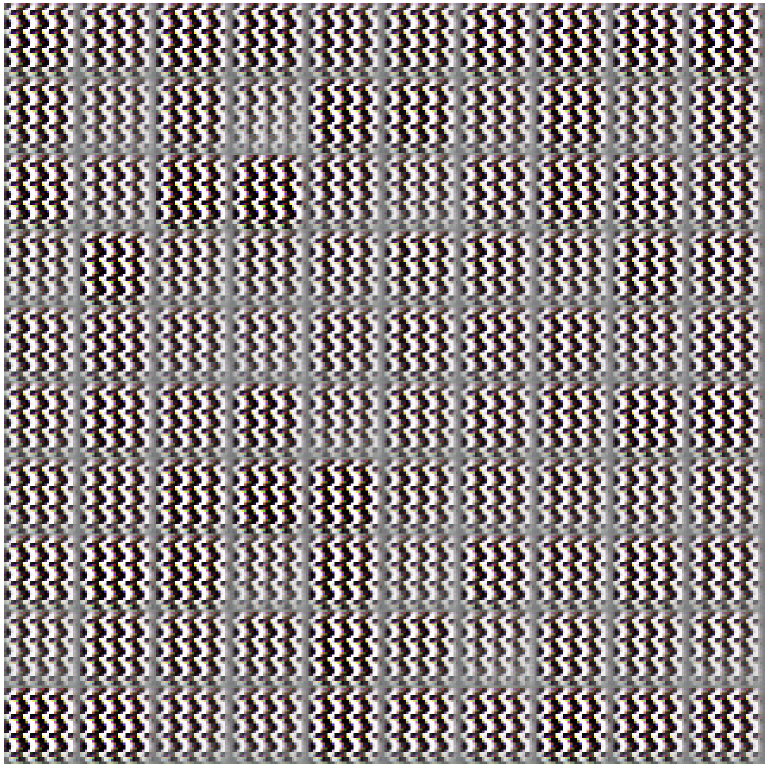
Run 2, epoch 2

Training instability

Running code from [Salimans+ NeurIPS-16]:



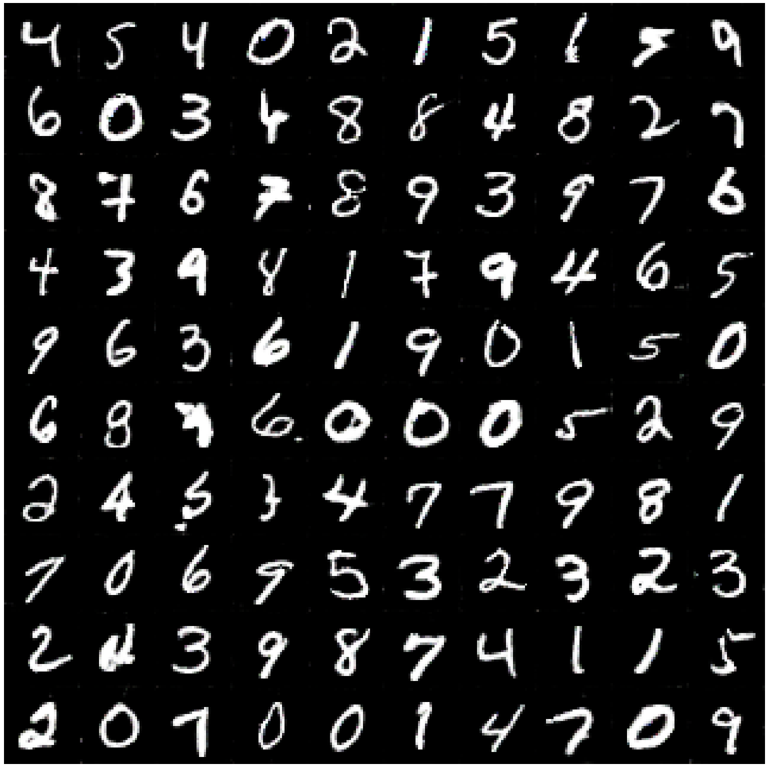
Run 1, epoch 900



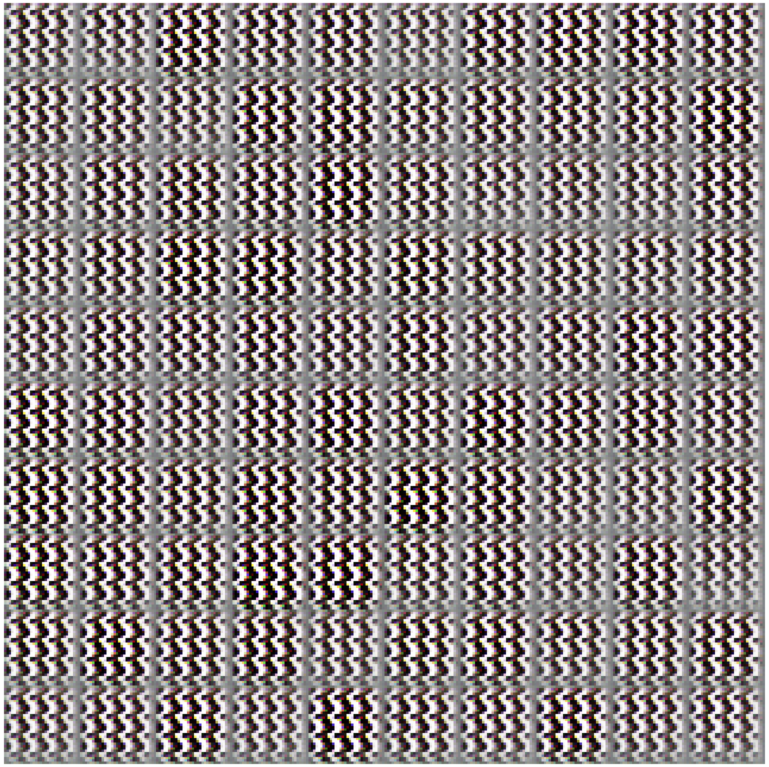
Run 2, epoch 3

Training instability

Running code from [Salimans+ NeurIPS-16]:



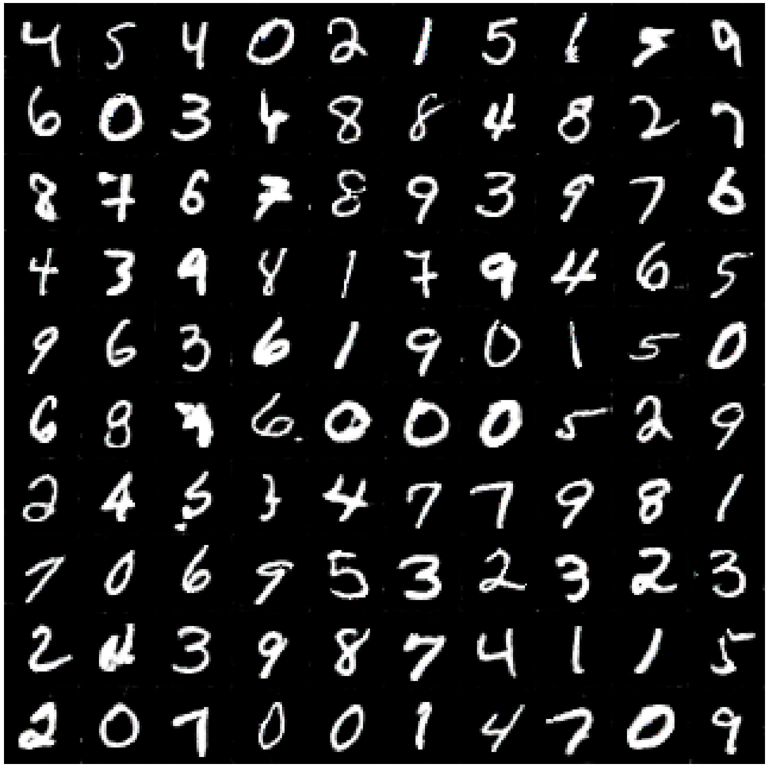
Run 1, epoch 900



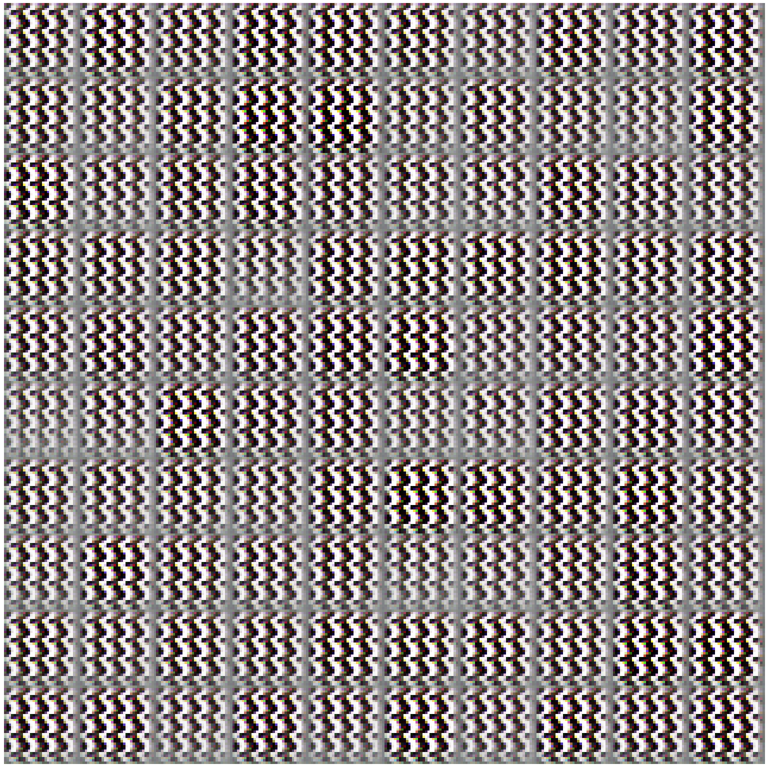
Run 2, epoch 4

Training instability

Running code from [Salimans+ NeurIPS-16]:



Run 1, epoch 900



Run 2, epoch 5

One view: distances between distributions

- What happens when D_ψ is at its optimum?

One view: distances between distributions

- What happens when D_ψ is at its optimum?
- If distributions have densities, $D^*_\psi(x) = \frac{p(x)}{p(x)+q_\theta(x)}$

One view: distances between distributions

- What happens when D_ψ is at its optimum?
- If distributions have densities, $D^*_\psi(x) = \frac{p(x)}{p(x)+q_\theta(x)}$
- If D_ψ stays optimal throughout, θ tries to minimize

$$\frac{1}{2} \mathop{\mathbb{E}}_{X\sim\mathbb{P}}\left[\log \frac{p(X)}{p(X)+q_\theta(X)}\right] + \frac{1}{2} \mathop{\mathbb{E}}_{Y\sim\mathbb{Q}_\theta}\left[\log \frac{q_\theta(X)}{p(X)+q_\theta(X)}\right]$$

which is $\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) - \log 2$

Jensen-Shannon divergence

$$\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) = \frac{1}{2} \int p(x) \log \frac{p(x)}{\frac{1}{2} p(x) + \frac{1}{2} q_\theta(x)} \mathrm{d}x + \frac{1}{2} \int q_\theta(x) \log \frac{q_\theta(x)}{\frac{1}{2} p(x) + \frac{1}{2} q_\theta(x)} \mathrm{d}x$$

Jensen-Shannon divergence

$$
\begin{aligned}
\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) &= \frac{1}{2} \int p(x) \log \frac{p(x)}{\frac{1}{2}p(x) + \frac{1}{2}q_\theta(x)} \mathrm{d}x \\
&\quad + \frac{1}{2} \int q_\theta(x) \log \frac{q_\theta(x)}{\frac{1}{2}p(x) + \frac{1}{2}q_\theta(x)} \mathrm{d}x \\
&= \frac{1}{2} \mathrm{KL}\left(\mathbb{P} \,\middle\|\, \frac{\mathbb{P} + \mathbb{Q}_\theta}{2}\right) + \frac{1}{2} \mathrm{KL}\left(\mathbb{Q}_\theta \,\middle\|\, \frac{\mathbb{P} + \mathbb{Q}_\theta}{2}\right)
\end{aligned}
$$

Jensen-Shannon divergence

$$\begin{aligned}
\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) &= \frac{1}{2}\int p(x)\log\frac{p(x)}{\frac{1}{2}p(x)+\frac{1}{2}q_\theta(x)}\mathrm{d}x \\
&\quad + \frac{1}{2}\int q_\theta(x)\log\frac{q_\theta(x)}{\frac{1}{2}p(x)+\frac{1}{2}q_\theta(x)}\mathrm{d}x \\
&= \frac{1}{2}\mathrm{KL}\left(\mathbb{P}\,\middle\|\,\frac{\mathbb{P}+\mathbb{Q}_\theta}{2}\right) + \frac{1}{2}\mathrm{KL}\left(\mathbb{Q}_\theta\,\middle\|\,\frac{\mathbb{P}+\mathbb{Q}_\theta}{2}\right) \\
&= \mathrm{H}\left[\frac{\mathbb{P}+\mathbb{Q}_\theta}{2}\right] - \frac{\mathrm{H}[\mathbb{P}]+\mathrm{H}[\mathbb{Q}_\theta]}{2}
\end{aligned}$$

JS with disjoint support [Arjovsky/Bottou ICLR-17]

$$\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) = \frac{1}{2} \int p(x) \log \frac{p(x)}{\frac{1}{2}p(x) + \frac{1}{2}q_\theta(x)} \mathrm{d}x$$

$$+ \frac{1}{2} \int q_\theta(x) \log \frac{q_\theta(x)}{\frac{1}{2}p(x) + \frac{1}{2}q_\theta(x)} \mathrm{d}x$$

- If $\mathbb{P}$ and $\mathbb{Q}_\theta$ have (almost) disjoint support

$$\frac{1}{2} \int p(x) \log \frac{p(x)}{\frac{1}{2}p(x)} \mathrm{d}x$$

JS with disjoint support [Arjovsky/Bottou ICLR-17]

$$\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) = \frac{1}{2}\int p(x)\log\frac{p(x)}{\frac{1}{2}p(x)+\frac{1}{2}q_\theta(x)}\mathrm{d}x + \frac{1}{2}\int q_\theta(x)\log\frac{q_\theta(x)}{\frac{1}{2}p(x)+\frac{1}{2}q_\theta(x)}\mathrm{d}x$$

- If $\mathbb{P}$ and $\mathbb{Q}_\theta$ have (almost) disjoint support

$$\frac{1}{2}\int p(x)\log\frac{p(x)}{\frac{1}{2}p(x)}\mathrm{d}x = \frac{1}{2}\int p(x)\log(2)\mathrm{d}x$$

JS with disjoint support [Arjovsky/Bottou ICLR-17]

$$\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) = \frac{1}{2}\int p(x)\log\frac{p(x)}{\frac{1}{2}p(x) + \frac{1}{2}q_\theta(x)}\mathrm{d}x + \frac{1}{2}\int q_\theta(x)\log\frac{q_\theta(x)}{\frac{1}{2}p(x) + \frac{1}{2}q_\theta(x)}\mathrm{d}x$$

- If $\mathbb{P}$ and $\mathbb{Q}_\theta$ have (almost) disjoint support

$$\frac{1}{2}\int p(x)\log\frac{p(x)}{\frac{1}{2}p(x)}\mathrm{d}x = \frac{1}{2}\int p(x)\log(2)\mathrm{d}x = \frac{1}{2}\log 2$$

JS with disjoint support [Arjovsky/Bottou ICLR-17]

$$\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) = \frac{1}{2}\int p(x)\log\frac{p(x)}{\frac{1}{2}p(x)+\frac{1}{2}q_\theta(x)}\mathrm{d}x + \frac{1}{2}\int q_\theta(x)\log\frac{q_\theta(x)}{\frac{1}{2}p(x)+\frac{1}{2}q_\theta(x)}\mathrm{d}x$$

- If $\mathbb{P}$ and $\mathbb{Q}_\theta$ have (almost) disjoint support

$$\frac{1}{2}\int p(x)\log\frac{p(x)}{\frac{1}{2}p(x)}\mathrm{d}x = \frac{1}{2}\int p(x)\log(2)\mathrm{d}x = \frac{1}{2}\log 2$$

so $\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) = \log 2$

Discriminator point of view

Generator ($\mathbb{Q}_\theta$)

Discriminator

Discriminator point of view

Generator ($\mathbb{Q}_\theta$)

Discriminator

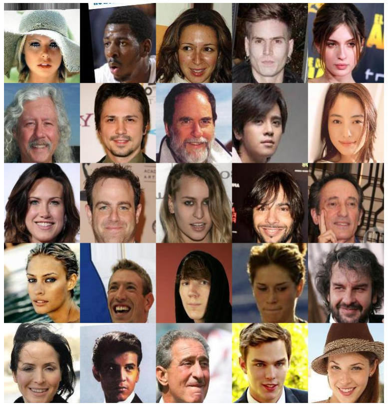
Is this real?

Discriminator point of view

Target ($\mathbb{P}$)

Generator ($\mathbb{Q}_\theta$)

Discriminator



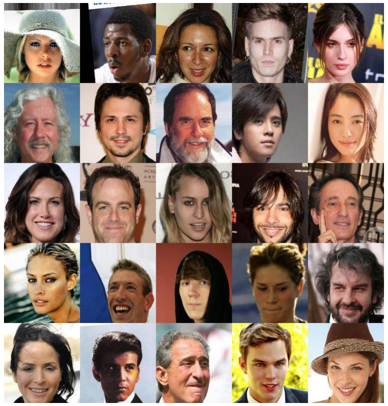
Is this real?

Discriminator point of view

Target ($\mathbb{P}$)

Generator ($\mathbb{Q}_\theta$)

Discriminator



Is this real?

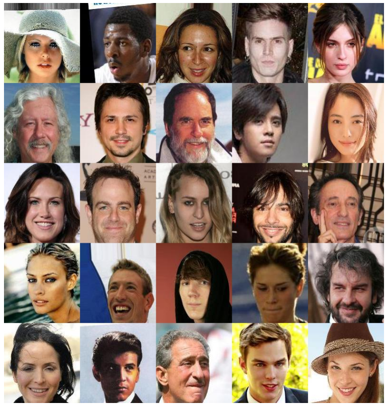
No way! $\mathrm{Pr}(\mathrm{real}) = 0.00$

Discriminator point of view

Target ($\mathbb{P}$)

Generator ($\mathbb{Q}_\theta$)

Discriminator



Is this real?

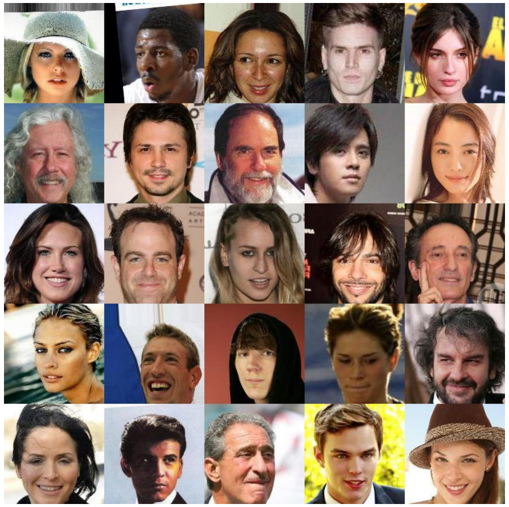
No way! $\mathbf{Pr(real)} = 0.00$

:(I don't know how to do any better...

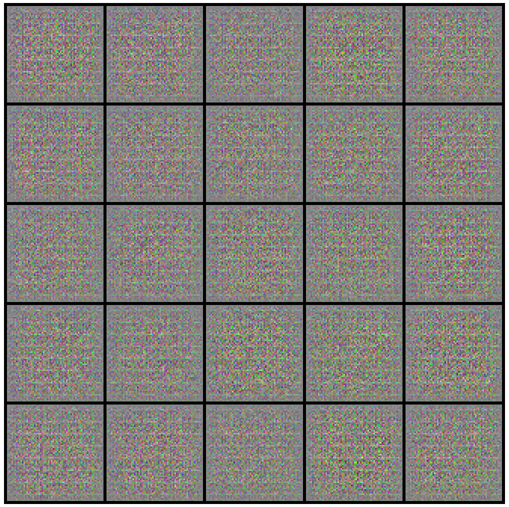
How likely is disjoint support?

- At initialization, pretty reasonable:

$\mathbb{P}$:



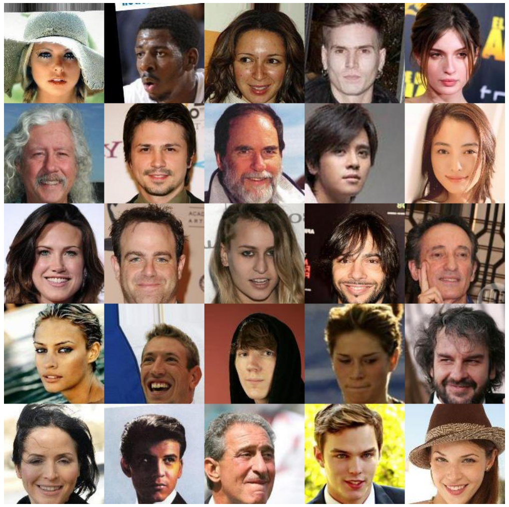
$\mathbb{Q}_\theta$:



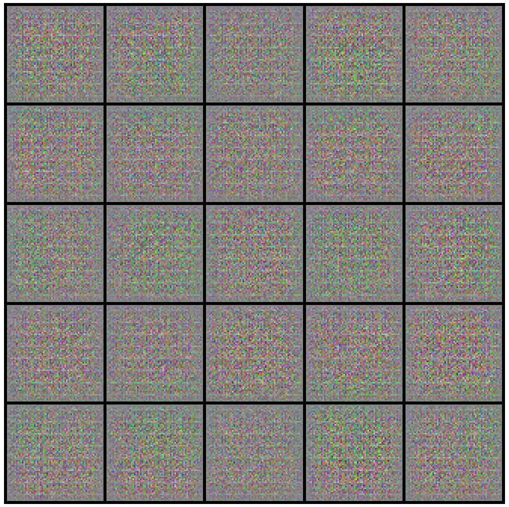
How likely is disjoint support?

- At initialization, pretty reasonable:

$\mathbb{P}$:



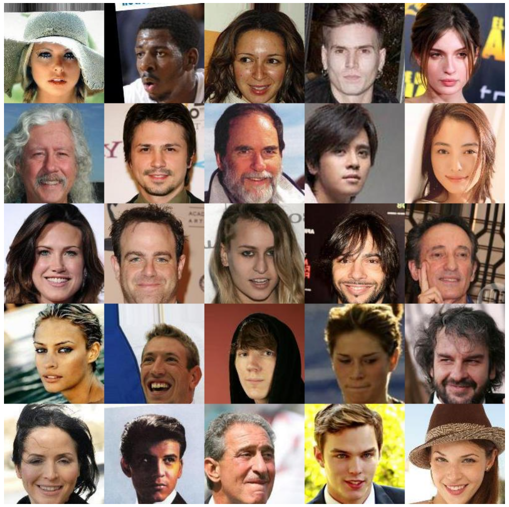
$\mathbb{Q}_\theta$:

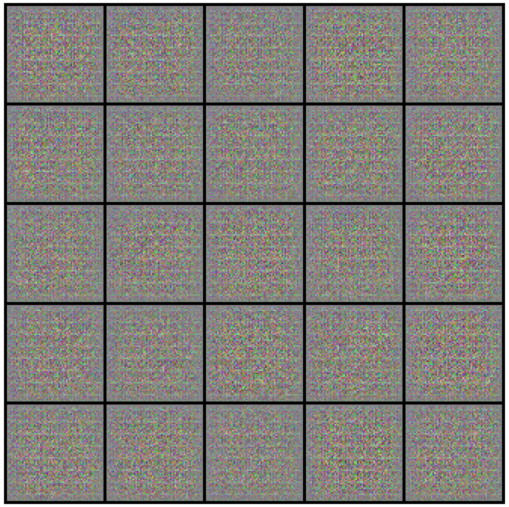


- Remember we might have $G_\theta : \mathbb{R}^{100} \to \mathbb{R}^{64\times64\times3}$

How likely is disjoint support?

- At initialization, pretty reasonable:

$\mathbb{P}$: 

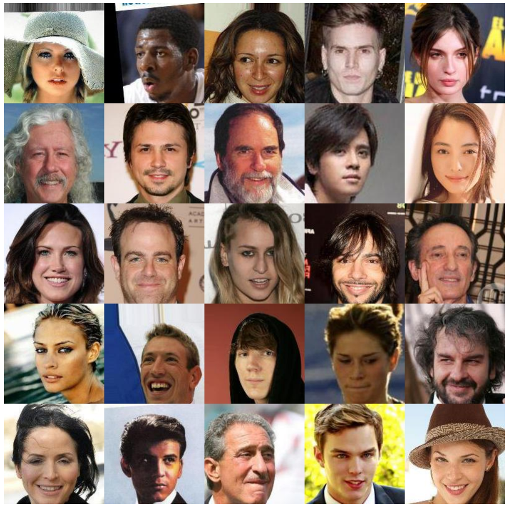
$\mathbb{Q}_\theta$: 

- Remember we might have $G_\theta : \mathbb{R}^{100} \to \mathbb{R}^{64\times64\times3}$
- For usual G_θ, $\mathbb{Q}_\theta$ is supported on a countable union of manifolds with dim ≤ 100

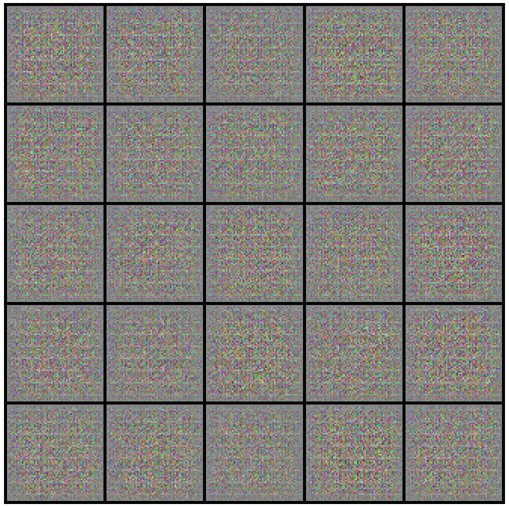
How likely is disjoint support?

- At initialization, pretty reasonable:

$\mathbb{P}$:



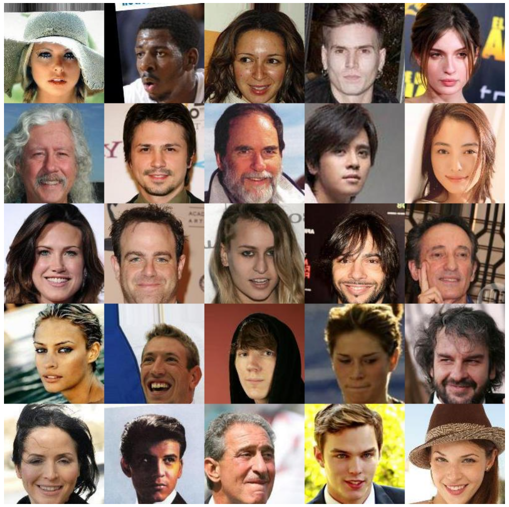
$\mathbb{Q}_\theta$:

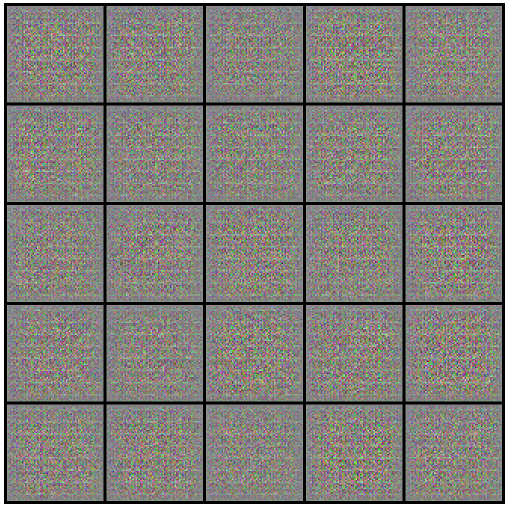


- Remember we might have $G_\theta : \mathbb{R}^{100} \to \mathbb{R}^{64\times64\times3}$
- For usual G_θ, $\mathbb{Q}_\theta$ is supported on a countable union of manifolds with dim ≤ 100
- "Natural image manifold" usually considered low-dim

How likely is disjoint support?

- At initialization, pretty reasonable:

$\mathbb{P}$: 

$\mathbb{Q}_\theta$: 

- Remember we might have $G_\theta : \mathbb{R}^{100} \to \mathbb{R}^{64\times 64\times 3}$
- For usual G_θ, $\mathbb{Q}_\theta$ is supported on a countable union of manifolds with dim ≤ 100
- "Natural image manifold" usually considered low-dim
- No chance that they'd align at init, so $\mathrm{JS}(\mathbb{P}, \mathbb{Q}_\theta) = \log 2$

A heuristic partial workaround

- Original GANs almost never use the minimax game

$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [\log D_\psi(X)] + \frac{1}{2} \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}_\theta} [\log(1 - D_\psi(Y))]$$

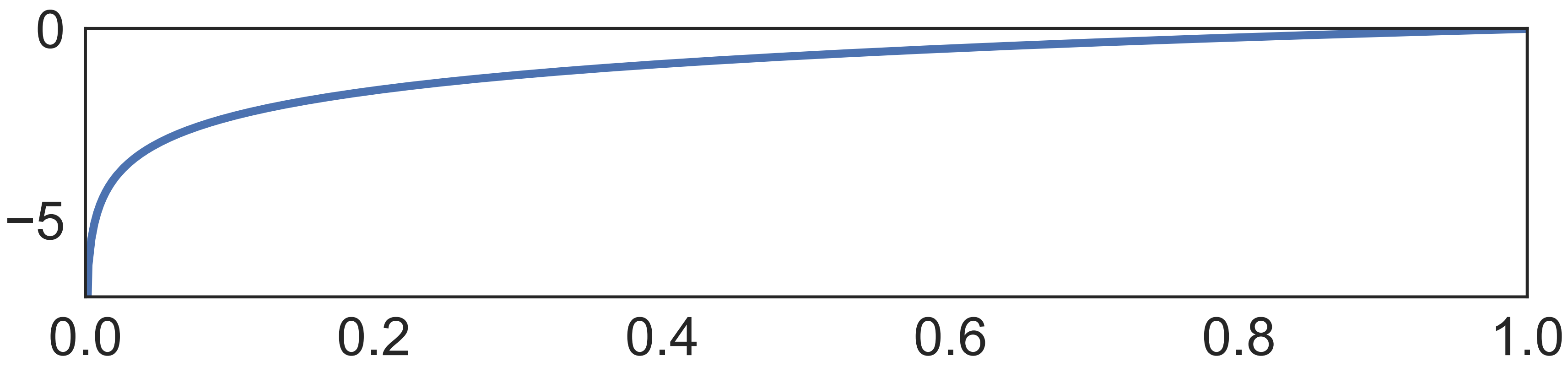
- $\max_\theta \log D_\psi(G_\theta(Z)),\ \text{not}\ \min_\theta \log(1 - D_\psi(G_\theta(Z)))$

A heuristic partial workaround

- Original GANs almost never use the minimax game

$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathbb{E}_{X \sim \mathbb{P}} [\log D_\psi(X)] + \frac{1}{2} \mathbb{E}_{Y \sim \mathbb{Q}_\theta} [\log(1 - D_\psi(Y))]$$

- $\max_\theta \log D_\psi(G_\theta(Z)),\ \text{not}\ \min_\theta \log(1 - D_\psi(G_\theta(Z)))$
- If D_ψ is near-perfect, near $\log 0$ instead of $\log 1$

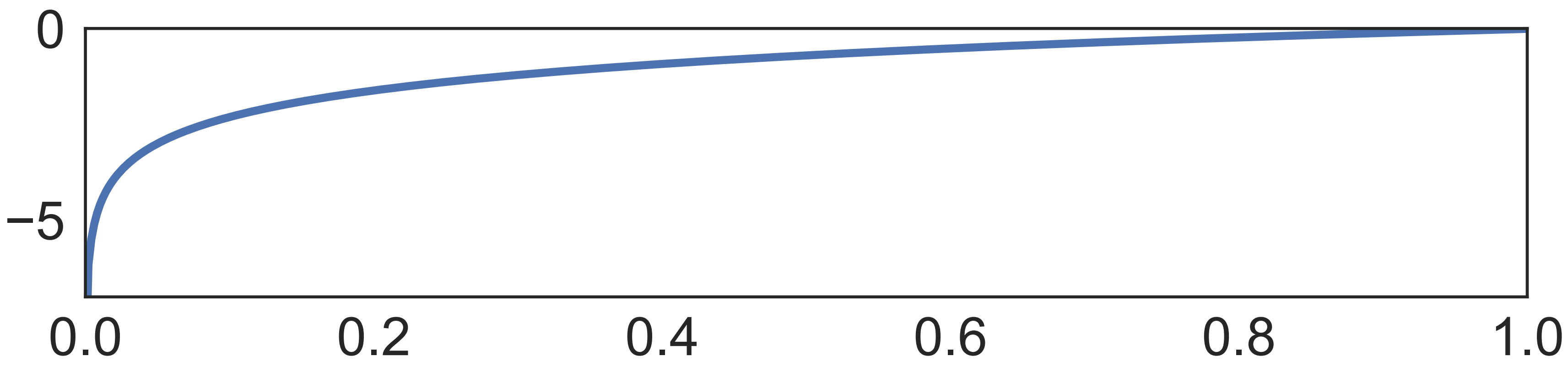


A heuristic partial workaround

- Original GANs almost never use the minimax game

$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [\log D_\psi(X)] + \frac{1}{2} \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}_\theta} [\log(1 - D_\psi(Y))]$$

- $\max_\theta \log D_\psi(G_\theta(Z))$, not $\min_\theta \log(1 - D_\psi(G_\theta(Z)))$
- If D_ψ is near-perfect, near $\log 0$ instead of $\log 1$



- When D_ψ is near-perfect, makes it unstable instead of stuck

Better Solution 1: Optimal Transport to the rescue

- Real problem: JS is not *continuous in the weak topology*

Better Solution 1: Optimal Transport to the rescue

- Real problem: JS is not *continuous in the weak topology*
  - Have $\mathbb{Q}_n \to \mathbb{Q}$ where $\mathrm{JS}(\mathbb{Q}_n, \mathbb{Q}) = \log 2 \nrightarrow 0$

Better Solution 1: Optimal Transport to the rescue

- Real problem: JS is not *continuous in the weak topology*
  - Have $\mathbb{Q}_n \to \mathbb{Q}$ where $\mathrm{JS}(\mathbb{Q}_n, \mathbb{Q}) = \log 2 \nrightarrow 0$
- What distances are?

Better Solution 1: Optimal Transport to the rescue

- Real problem: JS is not *continuous in the weak topology*
  - Have $\mathbb{Q}_n \to \mathbb{Q}$ where $\mathrm{JS}(\mathbb{Q}_n, \mathbb{Q}) = \log 2 \not\to 0$
- What distances are?
- One nice choice: Wasserstein

$$\mathcal{W}_r(\mathbb{P}, \mathbb{Q})^r = \inf_C \mathop{\mathbb{E}}_{(X,Y)\sim C(\mathbb{P},\mathbb{Q})} d(X, Y)^r$$

Better Solution 1: Optimal Transport to the rescue

- Real problem: JS is not *continuous in the weak topology*
  - Have $\mathbb{Q}_n \to \mathbb{Q}$ where $\mathrm{JS}(\mathbb{Q}_n, \mathbb{Q}) = \log 2 \not\to 0$
- What distances are?
- One nice choice: Wasserstein

$$\mathcal{W}_r(\mathbb{P}, \mathbb{Q})^r = \inf_C \mathop{\mathbb{E}}_{(X,Y)\sim C(\mathbb{P},\mathbb{Q})} d(X, Y)^r$$

- Especially because of Kantorovich-Rubinstein duality:

$$\mathcal{W}_1(\mathbb{P}, \mathbb{Q}) = \sup_{f:\|f\|_{\mathrm{Lip}}\le 1} \mathop{\mathbb{E}}_{X\sim\mathbb{P}}[f(X)] - \mathop{\mathbb{E}}_{Y\sim\mathbb{Q}}[f(Y)]$$

The Wasserstein distance

$$\mathcal{W}(\mathbb{P}, \mathbb{Q}) = \sup_{f : \|f\|_{\text{Lip}} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

$f : \mathcal{X} \to \mathbb{R}$ is a 1-Lipschitz *critic function*

$$\|f\|_{\text{Lip}} = \sup_{x,y \in \mathcal{X}} \frac{|f(x) - f(y)|}{\|x - y\|} = \sup_{x \in \mathcal{X}} \|\nabla f(x)\|$$

Turns out $\mathcal{W}$ is *continuous*: if $\mathbb{Q}_\theta \to \mathbb{P}$, then $\mathcal{W}(\mathbb{Q}_\theta, \mathbb{P}) \to 0$

The Wasserstein distance

$$\mathcal{W}(\mathbb{P}, \mathbb{Q}) = \sup_{f : \|f\|_{\mathrm{Lip}} \leq 1} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [f(X)] - \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}} [f(Y)]$$

$f : \mathcal{X} \to \mathbb{R}$ is a 1-Lipschitz *critic function*

$$\|f\|_{\mathrm{Lip}} = \sup_{x,y \in \mathcal{X}} \frac{|f(x) - f(y)|}{\|x - y\|} = \sup_{x \in \mathcal{X}} \|\nabla f(x)\|$$

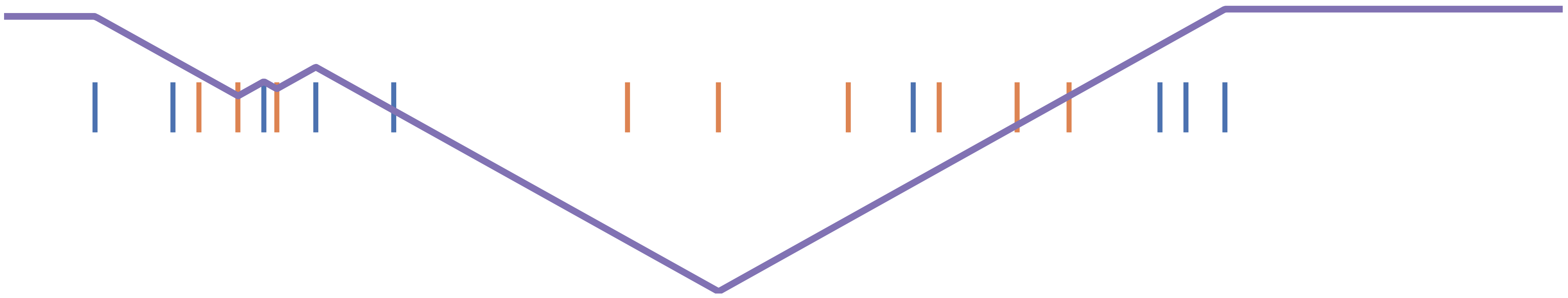
Turns out $\mathcal{W}$ is *continuous*: if $\mathbb{Q}_\theta \to \mathbb{P}$, then $\mathcal{W}(\mathbb{Q}_\theta, \mathbb{P}) \to 0$

The Wasserstein distance

$$\mathcal{W}(\mathbb{P}, \mathbb{Q}) = \sup_{f:\|f\|_{\mathrm{Lip}} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

$f : \mathcal{X} \to \mathbb{R}$ is a 1-Lipschitz *critic function*

$$\|f\|_{\mathrm{Lip}} = \sup_{x,y \in \mathcal{X}} \frac{|f(x) - f(y)|}{\|x - y\|} = \sup_{x \in \mathcal{X}} \|\nabla f(x)\|$$



Turns out $\mathcal{W}$ is *continuous*: if $\mathbb{Q}_\theta \to \mathbb{P}$, then $\mathcal{W}(\mathbb{Q}_\theta, \mathbb{P}) \to 0$

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Idea: turn discriminator D_ψ into a critic f_ψ
- Need to enforce $||f_\psi||_{\mathrm{Lip}} \leq 1$

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Idea: turn discriminator D_ψ into a critic f_ψ
- Need to enforce $\|f_\psi\|_{\text{Lip}} \leq 1$
- $$f_\psi(x) = \sigma_L(b_L + W_L \sigma_{L-1}(b_{L-1} + W_{L-1} \cdots))$$

  so for usual deep nets,

$$\|f_\psi\|_{\text{Lip}} \leq \|\sigma_L\|_{\text{Lip}} \|W_L\| \|\sigma_{L-1}\| \cdots \|W_1\|$$

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Idea: turn discriminator D_ψ into a critic f_ψ
- Need to enforce $\|f_\psi\|_{\mathrm{Lip}} \leq 1$
- $$f_\psi(x) = \sigma_L(b_L + W_L \sigma_{L-1}(b_{L-1} + W_{L-1} \cdots))$$

  so for usual deep nets,

  $$\|f_\psi\|_{\mathrm{Lip}} \leq \|\sigma_L\|_{\mathrm{Lip}} \|W_L\| \|\sigma_{L-1}\| \cdots \|W_1\|$$

- WGANs: just bound $\|W_i\|_\infty \leq C$; if $\|\sigma_i\| \leq 1$, then $\|W_i\| \leq \sqrt{d_i d_{i-1}} C$, and $\|f_\psi\| \leq C^d \sqrt{d_0} \prod_{i=1}^{L-1} d_i$

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Idea: turn discriminator D_ψ into a critic f_ψ
- Need to enforce $\|f_\psi\|_{\mathrm{Lip}} \le 1$
- $$f_\psi(x) = \sigma_L(b_L + W_L \sigma_{L-1}(b_{L-1} + W_{L-1} \cdots))$$

  so for usual deep nets,

  $$\|f_\psi\|_{\mathrm{Lip}} \le \|\sigma_L\|_{\mathrm{Lip}} \|W_L\| \|\sigma_{L-1}\| \cdots \|W_1\|$$

- WGANs: just bound $\|W_i\|_\infty \le C$; if $\|\sigma_i\| \le 1$,
  then $\|W_i\| \le \sqrt{d_i d_{i-1}} C$, and $\|f_\psi\| \le C^d \sqrt{d_0} \prod_{i=1}^{L-1} d_i$
- This turns out not to be a great idea.

WGAN-GP [Gulrajani+ NeurIPS-17]

- Controlling $\|\nabla f(X)\|$ *everywhere* is hard

WGAN-GP [Gulrajani+ NeurIPS-17]

- Controlling $\|\nabla f(X)\|$ *everywhere* is hard
- Instead, control $\|\nabla f(\tilde{X})\|$ *on average, near the data*

WGAN-GP [Gulrajani+ NeurIPS-17]

- Controlling $\|\nabla f(X)\|$ *everywhere* is hard
- Instead, control $\|\nabla f(\tilde{X})\|$ *on average, near the data*

$$\mathop{\mathbb{E}}_{\tilde{X}\sim\mathbb{S}} \left(\|\nabla_{\tilde{X}} f_\psi(\tilde{X})\| - 1\right)^2, \quad \mathbb{S} \text{ between } \mathbb{P} \text{ and } \mathbb{Q}_\theta$$

WGAN-GP [Gulrajani+ NeurIPS-17]

- Controlling $\|\nabla f(X)\|$ *everywhere* is hard
- Instead, control $\|\nabla f(\tilde{X})\|$ *on average, near the data*

$$\mathop{\mathbb{E}}_{\tilde{X}\sim\mathbb{S}} \left(\|\nabla_{\tilde{X}} f_\psi(\tilde{X})\| - 1\right)^2, \quad \mathbb{S} \textbf{ between } \mathbb{P} \textbf{ and } \mathbb{Q}_\theta$$

- Specifically: $\tilde{X} = \theta X + (1-\theta) Y, \theta \sim \mathrm{Uniform}([0,1])$

WGAN-GP [Gulrajani+ NeurIPS-17]

- Controlling $\|\nabla f(X)\|$ *everywhere* is hard
- Instead, control $\|\nabla f(\tilde{X})\|$ *on average, near the data*

$$\mathop{\mathbb{E}}_{\tilde{X}\sim\mathbb{S}} \left(\|\nabla_{\tilde{X}} f_\psi(\tilde{X})\| - 1\right)^2, \quad \mathbb{S} \text{ between } \mathbb{P} \text{ and } \mathbb{Q}_\theta$$

- Specifically: $\tilde{X} = \theta X + (1-\theta) Y, \theta \sim \text{Uniform}([0,1])$
- Works well! But...does it really estimate Wasserstein?

Solution 2: add noise

- Can keep JS if we make the problem harder
- Use $X + \varepsilon$, $Y + \varepsilon'$ for some independent, full-dim noise ε

Solution 2: add noise

- Can keep JS if we make the problem harder
- Use $X + \varepsilon$, $Y + \varepsilon'$ for some independent, full-dim noise ε
- But...how much noise ε to add? Also need more samples.

Solution 2: add noise

- Can keep JS if we make the problem harder
- Use $X + \varepsilon$, $Y + \varepsilon'$ for some independent, full-dim noise ε
- But…how much noise ε to add? Also need more samples.
- If $\varepsilon \sim \mathrm{N}(0, \gamma I)$ and $\gamma \to 0$, get [Mescheder+ NeurIPS-17]

$$\gamma \mathop{\mathbb{E}}_{\mathbb{P}}\left[(1 - D_\psi)^2 \|\nabla \log(D_\psi)\|^2\right] + \gamma \mathop{\mathbb{E}}_{\mathbb{Q}_\theta}\left[D_\psi^2 \|\nabla \log(D_\psi)\|^2\right]$$

Solution 2: add noise

- Can keep JS if we make the problem harder
- Use $X + \varepsilon$, $Y + \varepsilon'$ for some independent, full-dim noise ε
- But...how much noise ε to add? Also need more samples.
- If $\varepsilon \sim \mathrm{N}(0, \gamma I)$ and $\gamma \to 0$, get [Mescheder+ NeurIPS-17]

$$\gamma \mathop{\mathbb{E}}_{\mathbb{P}} \left[(1 - D_\psi)^2 \|\nabla \log(D_\psi)\|^2 \right] + \gamma \mathop{\mathbb{E}}_{\mathbb{Q}_\theta} \left[D_\psi^2 \|\nabla \log(D_\psi)\|^2 \right]$$

- Same kind of gradient penalty!

Solution 2: add noise

- Can keep JS if we make the problem harder
- Use $X + \varepsilon$, $Y + \varepsilon'$ for some independent, full-dim noise ε
- But…how much noise ε to add? Also need more samples.
- If $\varepsilon \sim \mathrm{N}(0, \gamma I)$ and $\gamma \to 0$, get [Mescheder+ NeurIPS-17]

$$\gamma \mathop{\mathbb{E}}_{\mathbb{P}}\left[(1 - D_\psi)^2 \|\nabla \log(D_\psi)\|^2\right] + \gamma \mathop{\mathbb{E}}_{\mathbb{Q}_\theta}\left[D_\psi^2 \|\nabla \log(D_\psi)\|^2\right]$$

- Same kind of gradient penalty!
- Can also simplify to e.g. [Mescheder+ ICML-18]

$$\gamma \mathop{\mathbb{E}}_{X \sim \mathbb{P}}\left[\|\nabla D_\psi(X)\|^2\right]$$

Solution 3: Spectral norm [Miyato+ ICLR-18]

- Regular deep nets: $f_\ell = \sigma\left(W_\ell f_{\ell-1}(x) + b_\ell\right)$
- Spectral normalization: $f_\ell = \sigma\left(\frac{1}{\|W_\ell\|_2} W_\ell f_{\ell-1}(x) + b_\ell\right)$
- $\|W\|_2 := \sup_{x \neq 0} \frac{\|Wx\|_2}{\|x\|_2} = \sigma_{\max}(W)$ is the *spectral norm*
- Guarantees $\|f\|_{\text{Lip}} \leq 1$
- Faster to evaluate than gradient penalties
- Not as well understood yet

Solution 3: Spectral norm [Miyato+ ICLR-18]

- Regular deep nets: $f_\ell = \sigma\left(W_\ell f_{\ell-1}(x) + b_\ell\right)$
- Spectral normalization: $f_\ell = \sigma\left(\frac{1}{\|W_\ell\|_2} W_\ell f_{\ell-1}(x) + b_\ell\right)$
- $\|W\|_2 := \mathbf{sup}_{x \neq 0} \frac{\|Wx\|_2}{\|x\|_2} = \sigma_{\mathbf{max}}(W)$ is the *spectral norm*
- Guarantees* $\|f\|_{\mathbf{Lip}} \leq 1$
- Faster to evaluate than gradient penalties
- Not as well understood yet

New samples [Mescheder+ ICML-18]

How to evaluate?

FID [Heusel+ NeurIPS-17] and KID [Bińkowski+ ICLR-18]

- Consider distance between distributions of image features
- Features $\phi(x)$ from a pretrained ImageNet classifier

FID [Heusel+ NeurIPS-17] and KID [Bińkowski+ ICLR-18]

- Consider distance between distributions of image features
- Features $\phi(x)$ from a pretrained ImageNet classifier
- FID: $\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}_\theta}\|^2 + \mathrm{Tr}\left(\Sigma_{\mathbb{P}} + \Sigma_{\mathbb{Q}_\theta} - 2\left(\Sigma_{\mathbb{P}}\Sigma_{\mathbb{Q}_\theta}\right)^{\frac{1}{2}}\right)$

FID [Heusel+ NeurIPS-17] and KID [Bińkowski+ ICLR-18]

- Consider distance between distributions of image features
- Features $\phi(x)$ from a pretrained ImageNet classifier
- FID: $\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}_\theta}\|^2 + \mathbf{Tr}\left(\Sigma_{\mathbb{P}} + \Sigma_{\mathbb{Q}_\theta} - 2\left(\Sigma_{\mathbb{P}}\Sigma_{\mathbb{Q}_\theta}\right)^{\frac{1}{2}}\right)$
    - Estimator very biased, small variance

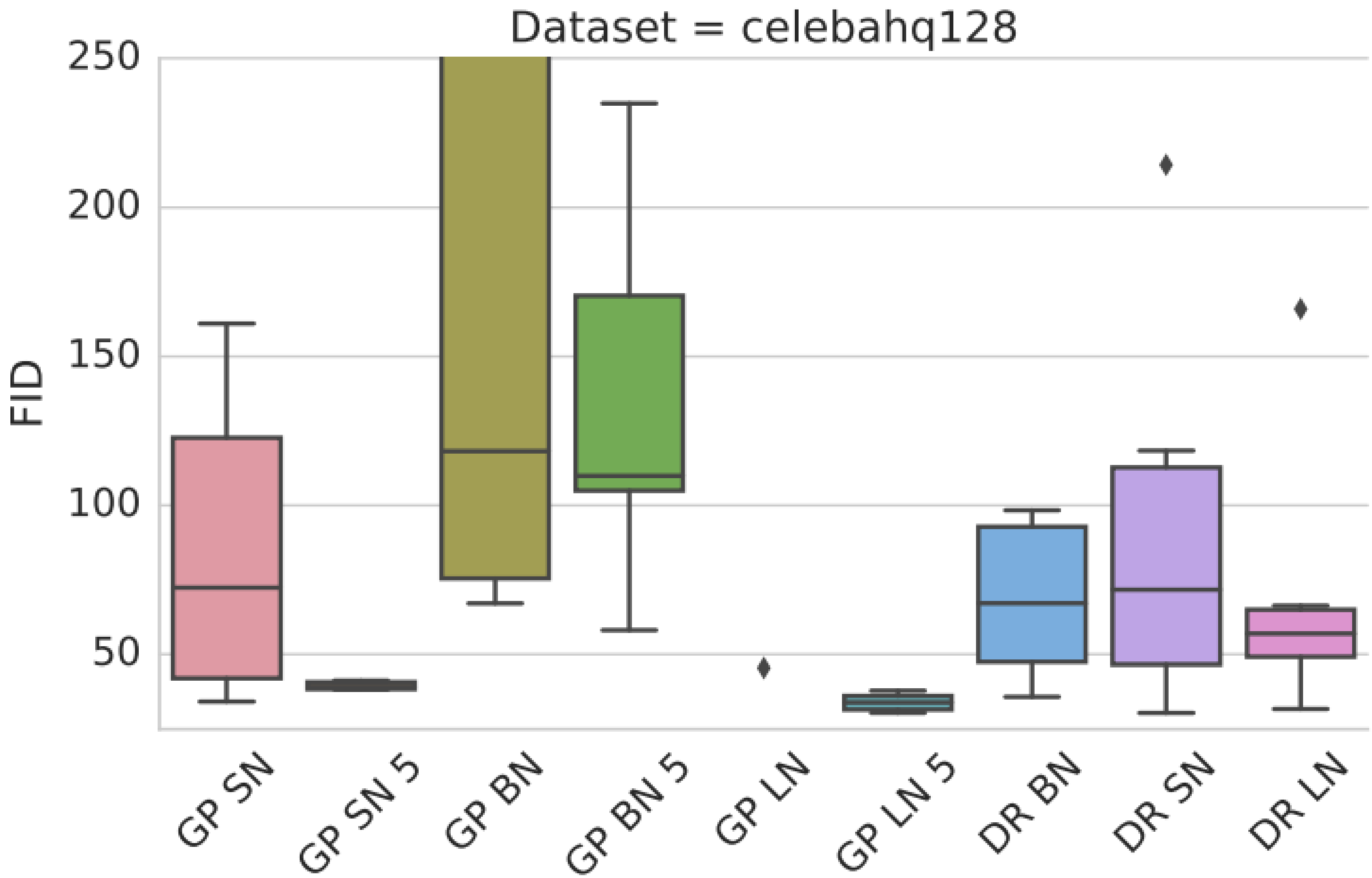
FID [Heusel+ NeurIPS-17] and KID [Bińkowski+ ICLR-18]

- Consider distance between distributions of image features
- Features $\phi(x)$ from a pretrained ImageNet classifier
- FID: $\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}_\theta}\|^2 + \mathbf{Tr}\left(\Sigma_{\mathbb{P}} + \Sigma_{\mathbb{Q}_\theta} - 2\left(\Sigma_{\mathbb{P}}\Sigma_{\mathbb{Q}_\theta}\right)^{\frac{1}{2}}\right)$
  - Estimator very biased, small variance
- KID: use Maximum Mean Discrepancy instead

FID [Heusel+ NeurIPS-17] and KID [Bińkowski+ ICLR-18]

- Consider distance between distributions of image features
- Features $\phi(x)$ from a pretrained ImageNet classifier
- FID: $\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}_\theta}\|^2 + \mathbf{Tr}\left(\Sigma_{\mathbb{P}} + \Sigma_{\mathbb{Q}_\theta} - 2\left(\Sigma_{\mathbb{P}}\Sigma_{\mathbb{Q}_\theta}\right)^{\frac{1}{2}}\right)$
    - Estimator very biased, small variance
- KID: use Maximum Mean Discrepancy instead
    - Similar distance with unbiased, ~normal estimator!

Comparing approaches [Kurach+ ICML-19]



Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f : \|f\|_{\mathcal{H}_k} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

Maximum Mean Discrepancy

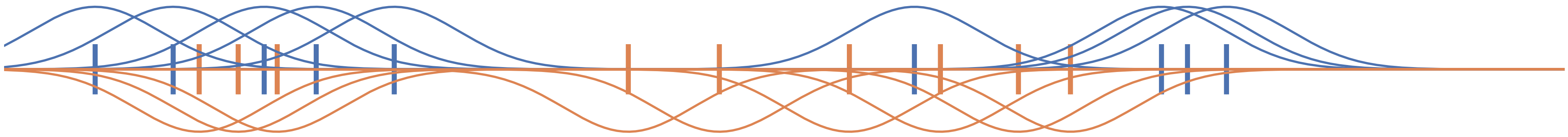
$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f:\|f\|_{\mathcal{H}_k} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f : \|f\|_{\mathcal{H}_k} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

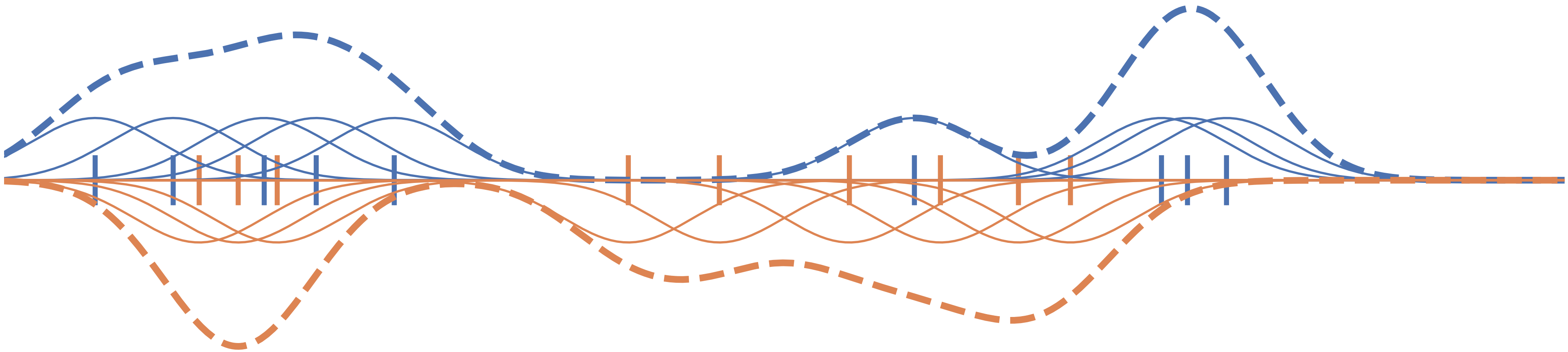
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f : \|f\|_{\mathcal{H}_k} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

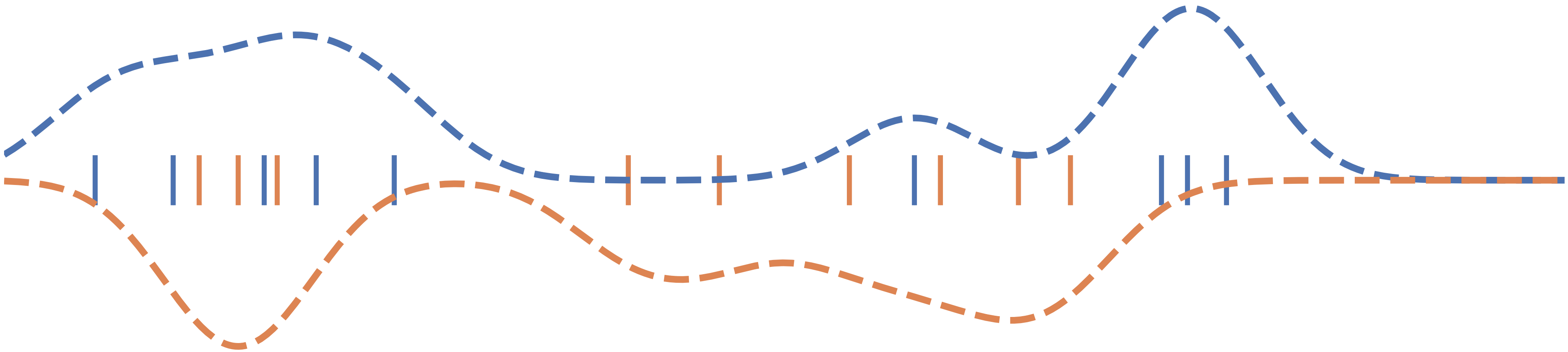
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f:\|f\|_{\mathcal{H}_k} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

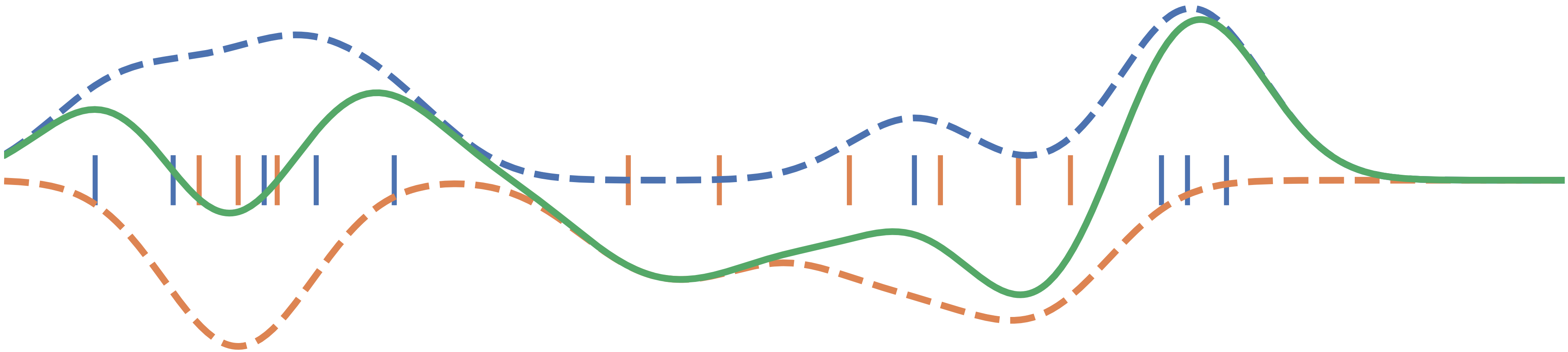
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$



Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f:\|f\|_{\mathcal{H}_k} \leq 1} \mathbb{E}_{X\sim\mathbb{P}}[f(X)] - \mathbb{E}_{Y\sim\mathbb{Q}}[f(Y)]$$

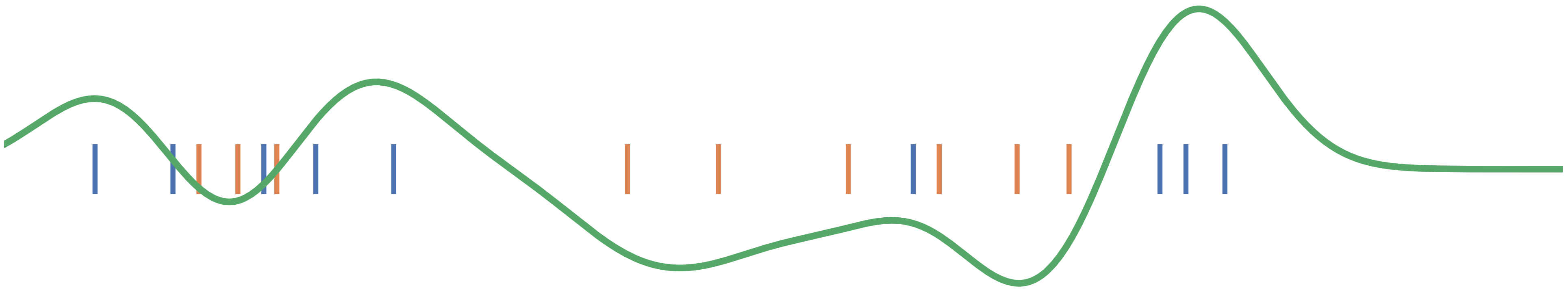
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f : \|f\|_{\mathcal{H}_k} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

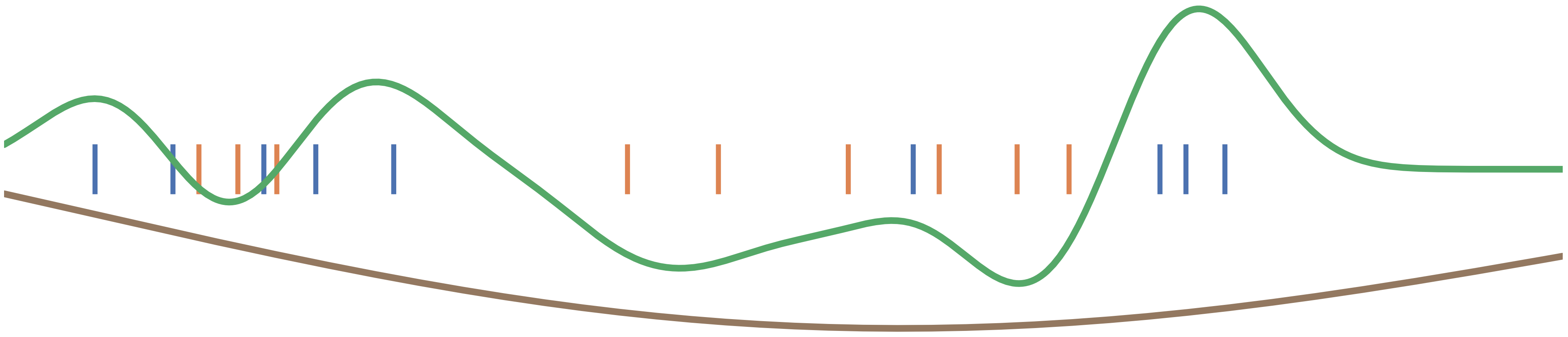
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f : \|f\|_{\mathcal{H}_k} \leq 1} \mathop{\mathbb{E}}_{X \sim \mathbb{P}}[f(X)] - \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}}[f(Y)]$$

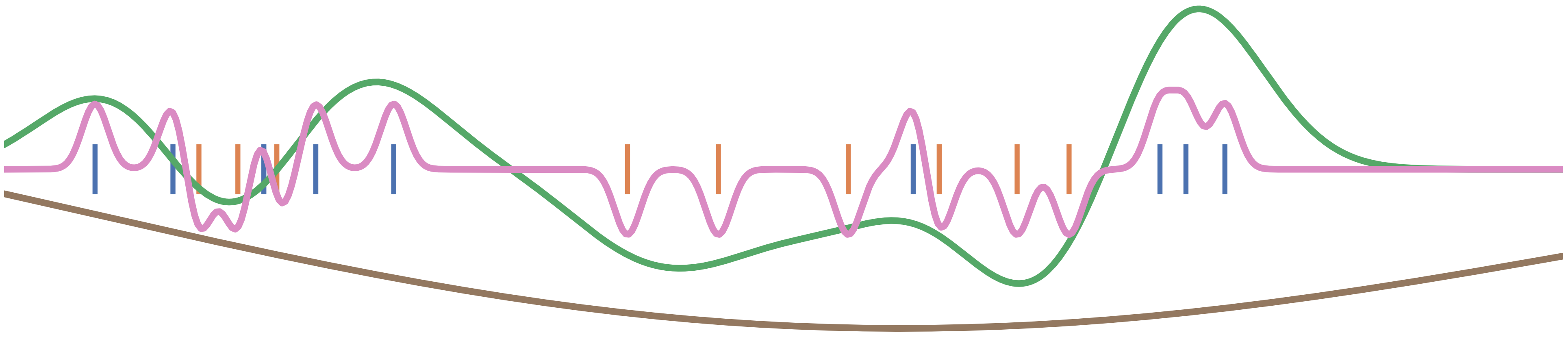
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f : \|f\|_{\mathcal{H}_k} \leq 1} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [f(X)] - \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}} [f(Y)]$$

$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

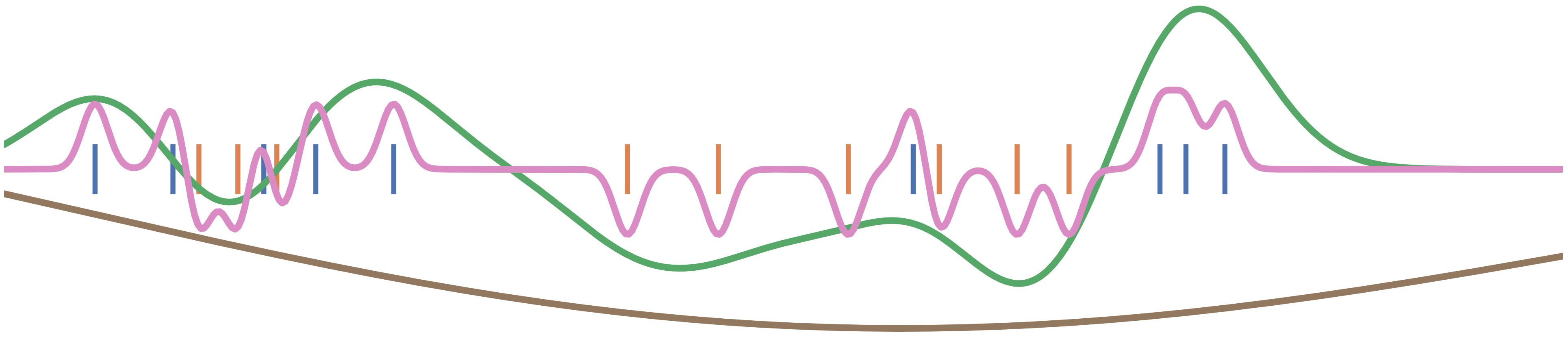


Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f : \|f\|_{\mathcal{H}_k} \leq 1} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [f(X)] - \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}} [f(Y)]$$

$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

Optimal f *analytically*: $f^*(t) \propto \mathbb{E}_{X \sim \mathbb{P}}\, k(t, X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, k(t, Y)$



Estimating MMD

$$\mathrm{MMD}_k^2(\mathbb{P}, \mathbb{Q}) = \mathop{\mathbb{E}}_{X, X' \sim \mathbb{P}}[k(X, X')] + \mathop{\mathbb{E}}_{Y, Y' \sim \mathbb{Q}}[k(Y, Y')] - 2 \mathop{\mathbb{E}}_{\substack{X \sim \mathbb{P} \\ Y \sim \mathbb{Q}}}[k(X, Y)]$$

Estimating MMD

$$\mathrm{MMD}_k^2(\mathbb{P}, \mathbb{Q}) = \mathop{\mathbb{E}}_{X, X' \sim \mathbb{P}}[k(X, X')] + \mathop{\mathbb{E}}_{Y, Y' \sim \mathbb{Q}}[k(Y, Y')] - 2 \mathop{\mathbb{E}}_{\substack{X \sim \mathbb{P} \\ Y \sim \mathbb{Q}}}[k(X, Y)]$$

$$\widehat{\mathrm{MMD}}_k^2(X, Y) = \mathrm{mean}(K_{XX}) + \mathrm{mean}(K_{YY}) - 2\,\mathrm{mean}(K_{XY})$$

Estimating MMD

$$\mathrm{MMD}_k^2(\mathbb{P}, \mathbb{Q}) = \mathop{\mathbb{E}}_{X, X' \sim \mathbb{P}}[k(X, X')] + \mathop{\mathbb{E}}_{Y, Y' \sim \mathbb{Q}}[k(Y, Y')] - 2 \mathop{\mathbb{E}}_{\substack{X \sim \mathbb{P} \\ Y \sim \mathbb{Q}}}[k(X, Y)]$$

$$\widehat{\mathrm{MMD}}_k^2(X, Y) = \mathrm{mean}(K_{XX}) + \mathrm{mean}(K_{YY}) - 2\,\mathrm{mean}(K_{XY})$$

$$K_{XX}$$

1.0	0.2	0.6
0.2	1.0	0.5
0.6	0.5	1.0

Estimating MMD

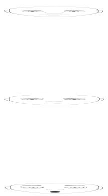
$$\mathrm{MMD}_k^2(\mathbb{P}, \mathbb{Q}) = \mathbb{E}_{X, X' \sim \mathbb{P}}[k(X, X')] + \mathbb{E}_{Y, Y' \sim \mathbb{Q}}[k(Y, Y')] - 2 \mathbb{E}_{\substack{X \sim \mathbb{P} \\ Y \sim \mathbb{Q}}}[k(X, Y)]$$

$$\widehat{\mathrm{MMD}}_k^2(X, Y) = \mathrm{mean}(K_{XX}) + \mathrm{mean}(K_{YY}) - 2\,\mathrm{mean}(K_{XY})$$

K_{XX}

1.0	0.2	0.6
0.2	1.0	0.5
0.6	0.5	1.0

K_{YY}



1.0	0.8	0.7
0.8	1.0	0.6
0.7	0.6	1.0

Estimating MMD

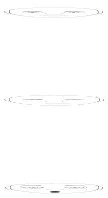
$$\mathrm{MMD}_k^2(\mathbb{P},\mathbb{Q}) = \mathop{\mathbb{E}}_{X,X'\sim\mathbb{P}}[k(X,X')] + \mathop{\mathbb{E}}_{Y,Y'\sim\mathbb{Q}}[k(Y,Y')] - 2\mathop{\mathbb{E}}_{\substack{X\sim\mathbb{P}\\ Y\sim\mathbb{Q}}}[k(X,Y)]$$

$$\widehat{\mathrm{MMD}}_k^2(X,Y) = \mathrm{mean}(K_{XX}) + \mathrm{mean}(K_{YY}) - 2\,\mathrm{mean}(K_{XY})$$

K_{XX}

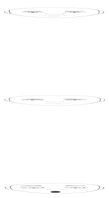
1.0	0.2	0.6
0.2	1.0	0.5
0.6	0.5	1.0

K_{YY}



1.0	0.8	0.7
0.8	1.0	0.6
0.7	0.6	1.0

K_{XY}



0.3	0.1	0.2
0.2	0.3	0.3
0.2	0.1	0.4

MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

- No need for a discriminator – just minimize $\widehat{\mathrm{MMD}}_k$!
- Continuous loss

Generator ($\mathbb{Q}_\theta$)

Critic

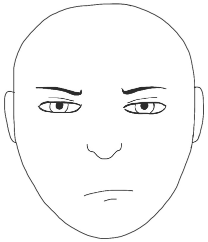
MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

- No need for a discriminator – just minimize $\widehat{\mathrm{MMD}}_k$!
- Continuous loss

Generator ($\mathbb{Q}_\theta$)

Critic

How are these?



MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

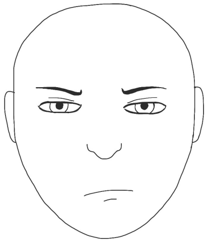
- No need for a discriminator – just minimize $\widehat{\mathrm{MMD}}_k$!
- Continuous loss

Generator ($\mathbb{Q}_\theta$)

Critic

Target ($\mathbb{P}$)

How are these?



MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

- No need for a discriminator – just minimize $\widehat{\mathrm{MMD}}_k$!
- Continuous loss

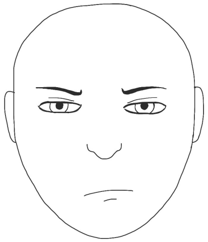
Target ($\mathbb{P}$)

Critic

Generator ($\mathbb{Q}_\theta$)

Not great! $\widehat{\mathrm{MMD}}(\mathbb{Q}_\theta, \mathbb{P}) = 0.75$

How are these?



MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

- No need for a discriminator – just minimize $\widehat{\mathrm{MMD}}_k$!
- Continuous loss

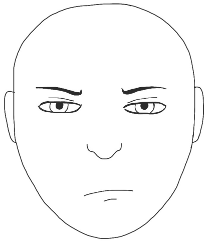
Generator ($\mathbb{Q}_\theta$)

Critic

Target ($\mathbb{P}$)

Not great! $\widehat{\mathrm{MMD}}(\mathbb{Q}_\theta, \mathbb{P}) = 0.75$

How are these?



:(I'll try harder...

MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

- No need for a discriminator – just minimize $\widehat{\mathrm{MMD}}_k$!
- Continuous loss

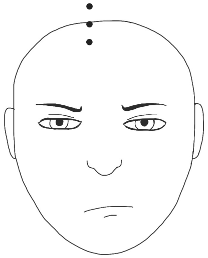
Generator ($\mathbb{Q}_\theta$)

Critic

Target ($\mathbb{P}$)

Not great! $\widehat{\mathrm{MMD}}(\mathbb{Q}_\theta, \mathbb{P}) = 0.75$

How are these?



:(I'll try harder...

MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

MNIST, mix of Gaussian kernels

$\mathbb{P}$

MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

MNIST, mix of Gaussian kernels

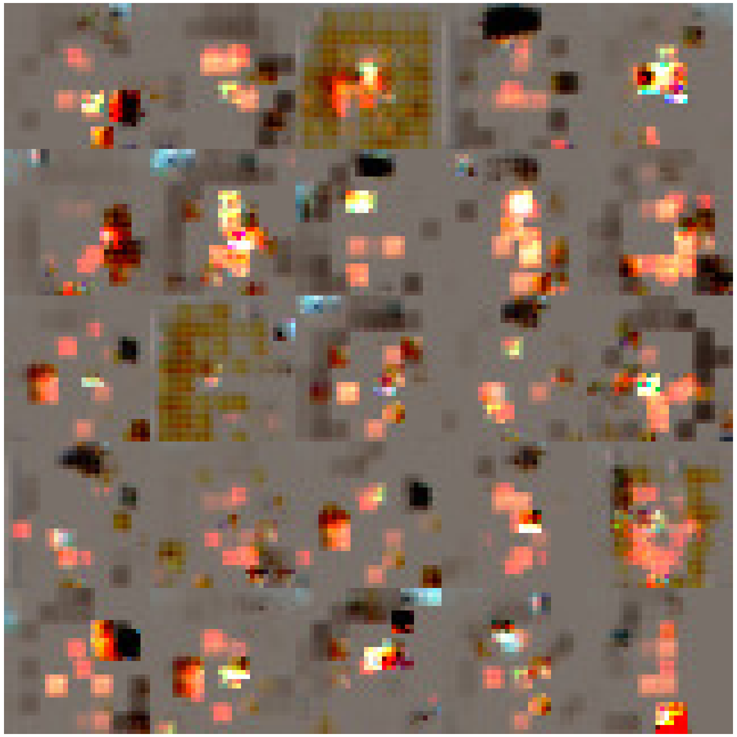
$\mathbb{P}$

$\mathbb{Q}_\theta$

Celeb-A, mix of rational quadratic + linear kernels

Celeb-A, mix of rational quadratic + linear kernels

$\mathbb{P}$



$\mathbb{Q}_\theta$

MMD loss with a smarter kernel

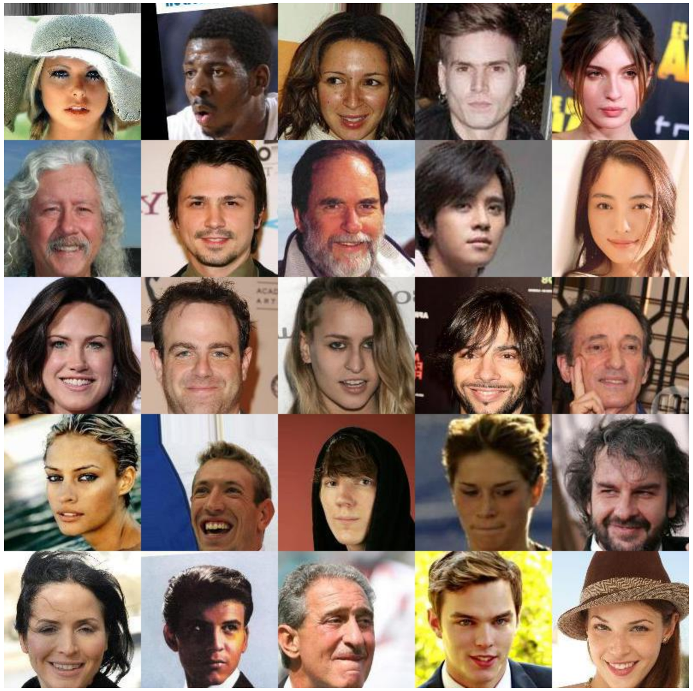
$$k(x, y) = k_{\text{top}}(\phi(x), \phi(y))$$

- $\phi : \mathcal{X} \to \mathbb{R}^{2048}$ from pretrained Inception net
- k_{top} simple: exponentiated quadratic or polynomial

MMD loss with a smarter kernel

$$k(x, y) = k_{\mathrm{top}}(\phi(x), \phi(y))$$

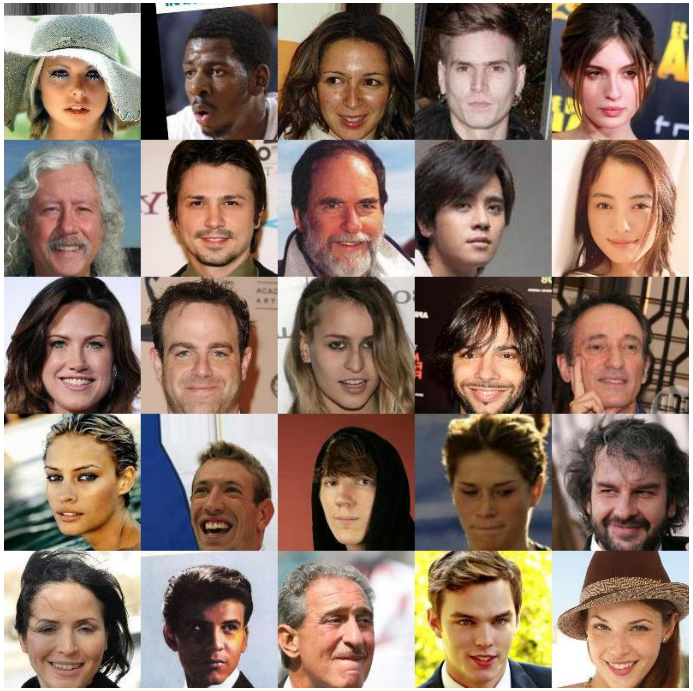
- $\phi : \mathcal{X} \to \mathbb{R}^{2048}$ from pretrained Inception net
- k_{top} simple: exponentiated quadratic or polynomial



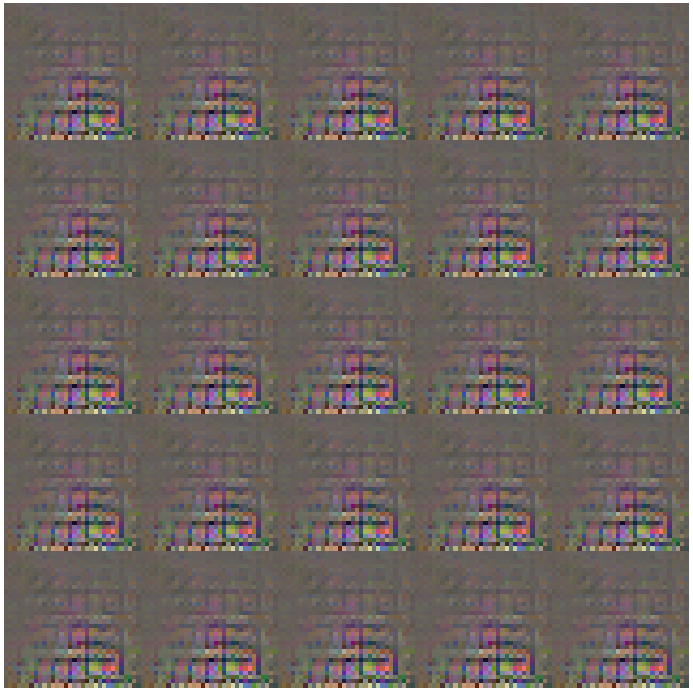
MMD loss with a smarter kernel

$$k(x, y) = k_{\text{top}}(\phi(x), \phi(y))$$

- $\phi : \mathcal{X} \to \mathbb{R}^{2048}$ from pretrained Inception net
- k_{top} simple: exponentiated quadratic or polynomial



$\mathbb{P}$



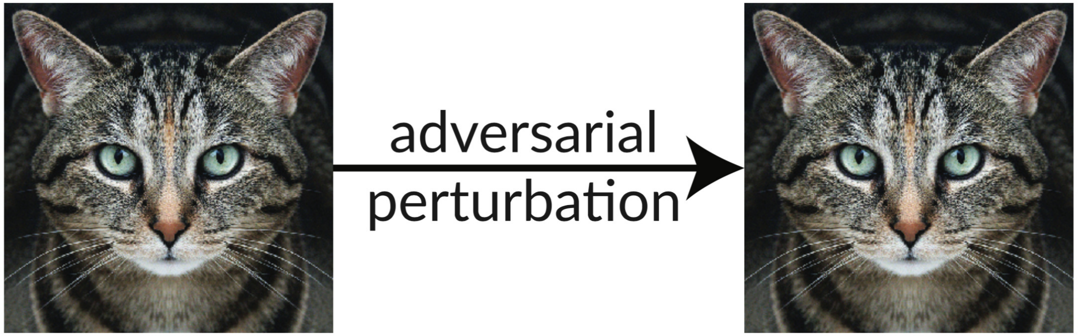
$\mathbb{Q}_\theta$

MMD loss with a smarter kernel

$$k(x, y) = k_{\text{top}}(\phi(x), \phi(y))$$

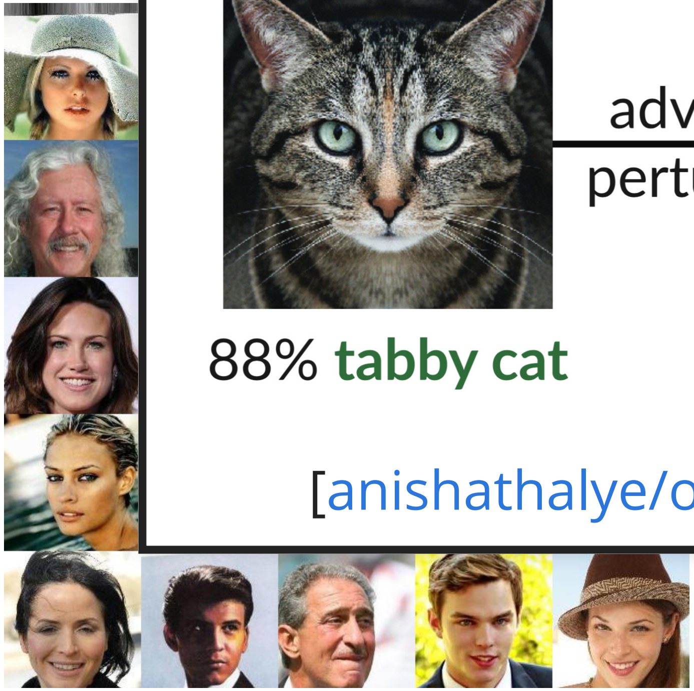
- ϕ : ...
- k_{top} ... omial

We just got adversarial examples!

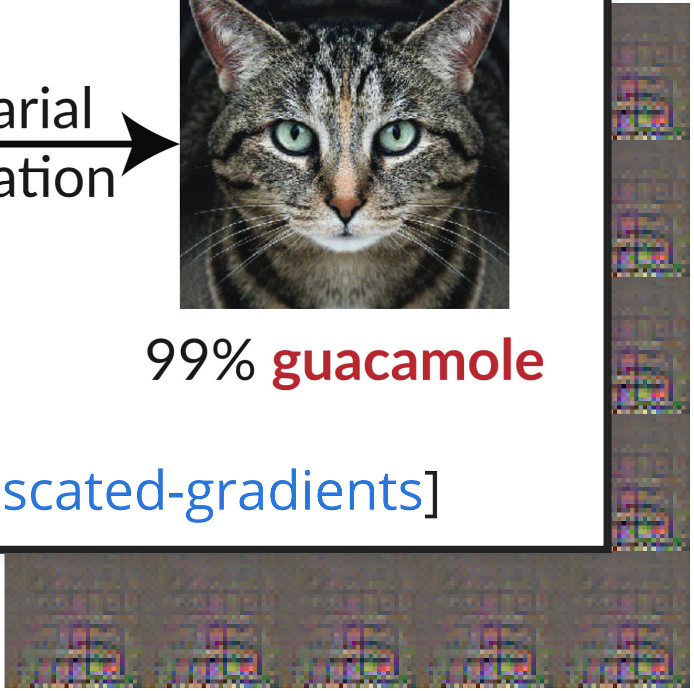


88% **tabby cat** | 99% **guacamole**

[anishathalye/obfuscated-gradients]



$\mathbb{P}$



$\mathbb{Q}_\theta$

Optimized MMD: MMD GANs [Li+ NeurIPS-17]

- Don't just use one kernel, use a *class* parameterized by ψ:

$$k_\psi(x, y) = k_{\text{top}}(\phi_\psi(x), \phi_\psi(y))$$

Optimized MMD: MMD GANs [Li+ NeurIPS-17]

- Don't just use one kernel, use a *class* parameterized by ψ:

$$k_\psi(x, y) = k_{\text{top}}(\phi_\psi(x), \phi_\psi(y))$$

- New distance based on *all* these kernels:

$$\mathcal{D}_{\text{MMD}}(\mathbb{P}, \mathbb{Q}) = \sup_{\psi \in \Psi} \text{MMD}_\psi(\mathbb{P}, \mathbb{Q})$$

Optimized MMD: MMD GANs [Li+ NeurIPS-17]

- Don't just use one kernel, use a *class* parameterized by ψ:

$$k_\psi(x, y) = k_{\text{top}}(\phi_\psi(x), \phi_\psi(y))$$

- New distance based on *all* these kernels:

$$\mathcal{D}_{\text{MMD}}(\mathbb{P}, \mathbb{Q}) = \sup_{\psi \in \Psi} \text{MMD}_\psi(\mathbb{P}, \mathbb{Q})$$

- Turns out that $\mathcal{D}_{\text{MMD}}$ *isn't* continuous: have $\mathbb{Q}_\theta \to \mathbb{P}$ but $\mathcal{D}_{\text{MMD}}(\mathbb{Q}_\theta, \mathbb{P}) \not\to 0$

Optimized MMD: MMD GANs [Li+ NeurIPS-17]

- Don't just use one kernel, use a *class* parameterized by ψ:

$$k_\psi(x, y) = k_{\text{top}}(\phi_\psi(x), \phi_\psi(y))$$

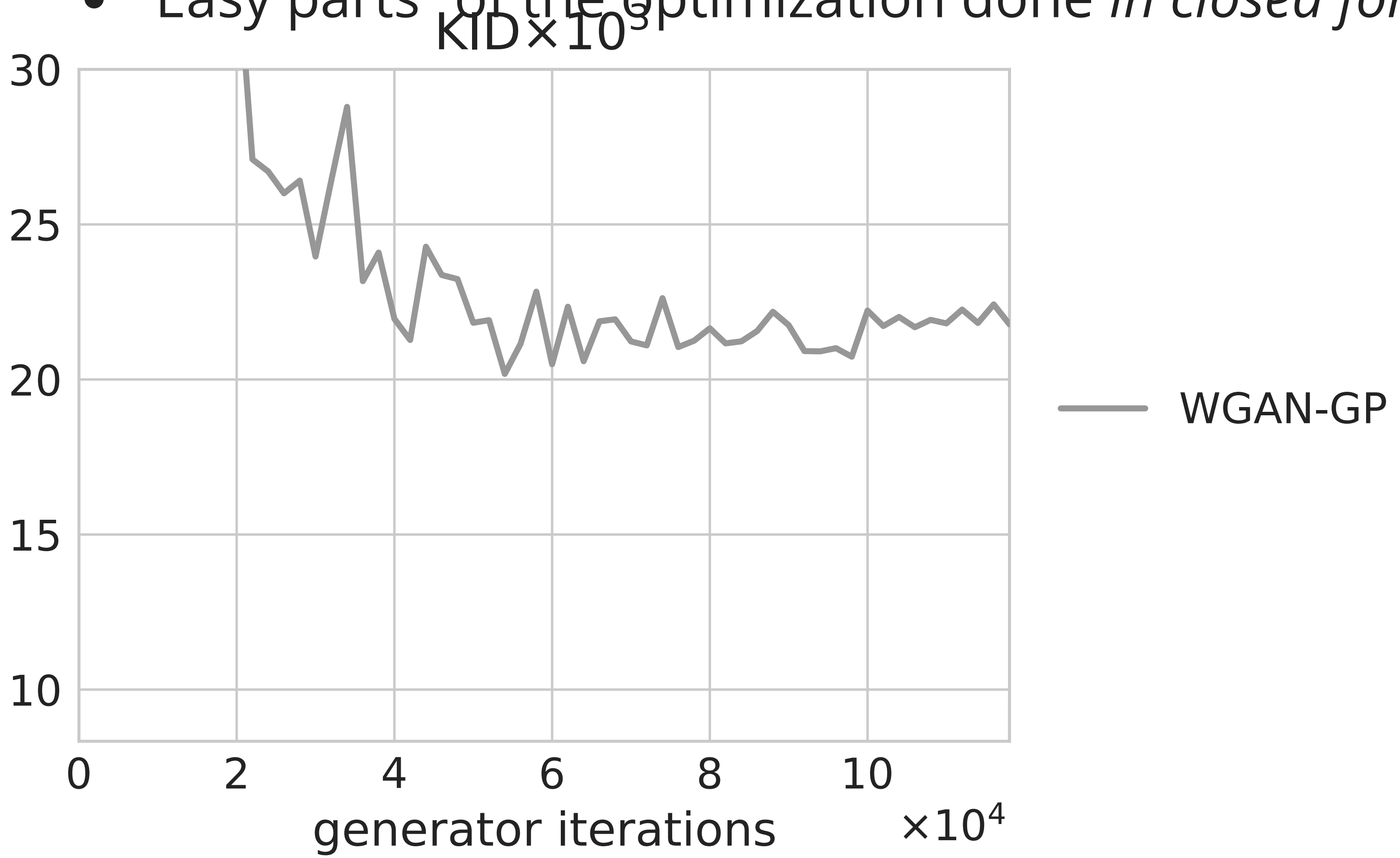
- New distance based on *all* these kernels:

$$\mathcal{D}_{\text{MMD}}(\mathbb{P}, \mathbb{Q}) = \sup_{\psi \in \Psi} \text{MMD}_\psi(\mathbb{P}, \mathbb{Q})$$

- Turns out that $\mathcal{D}_{\text{MMD}}$ *isn't* continuous: have $\mathbb{Q}_\theta \to \mathbb{P}$ but $\mathcal{D}_{\text{MMD}}(\mathbb{Q}_\theta, \mathbb{P}) \not\to 0$
- Scaled MMD GANs [Arbel+ NeurIPS-18] correct $\mathcal{D}_{\text{MMD}}$ with a gradient penalty to make it continuous

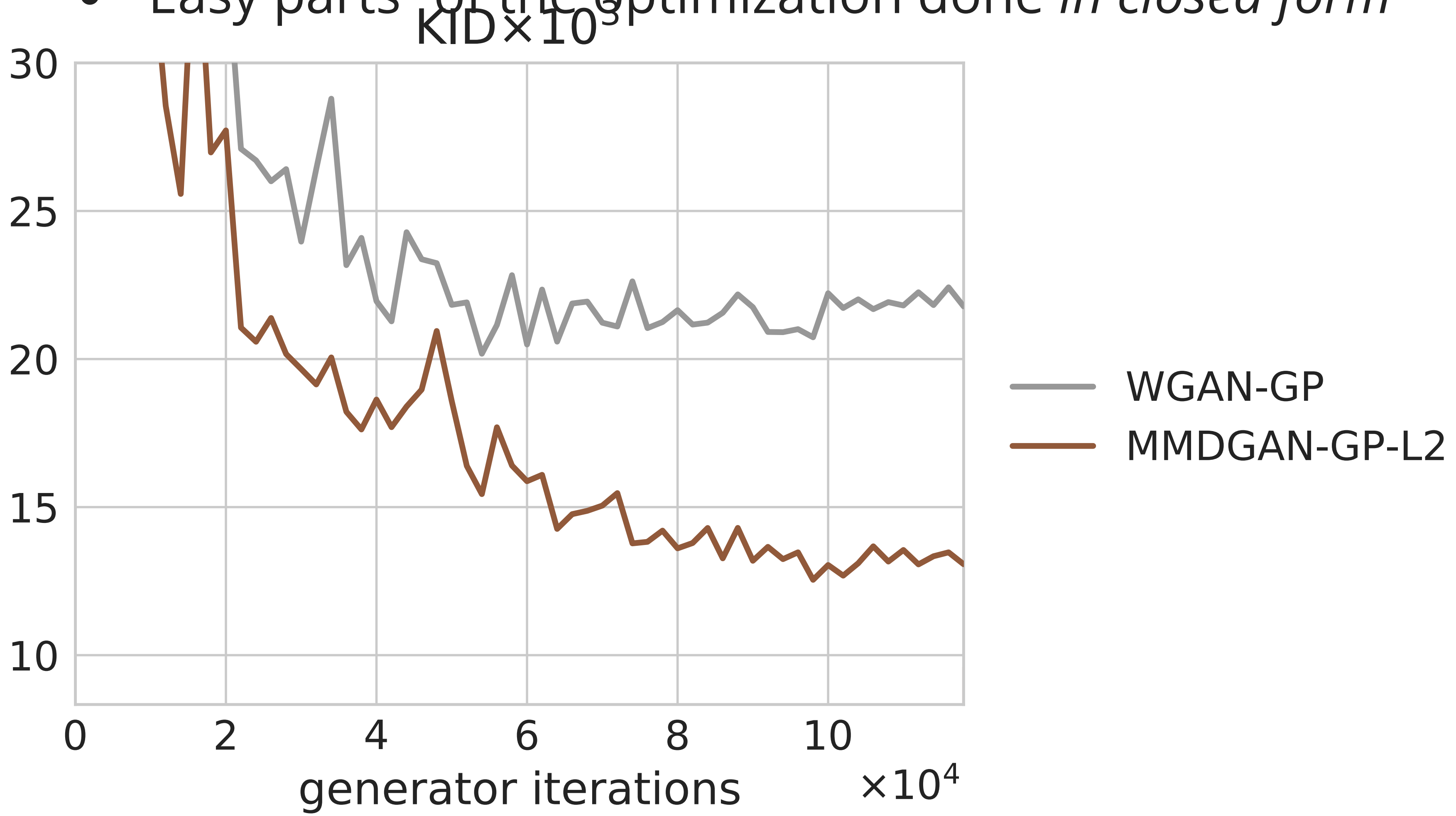
Why MMD GANs?

- “Easy parts” of the optimization done *in closed form*



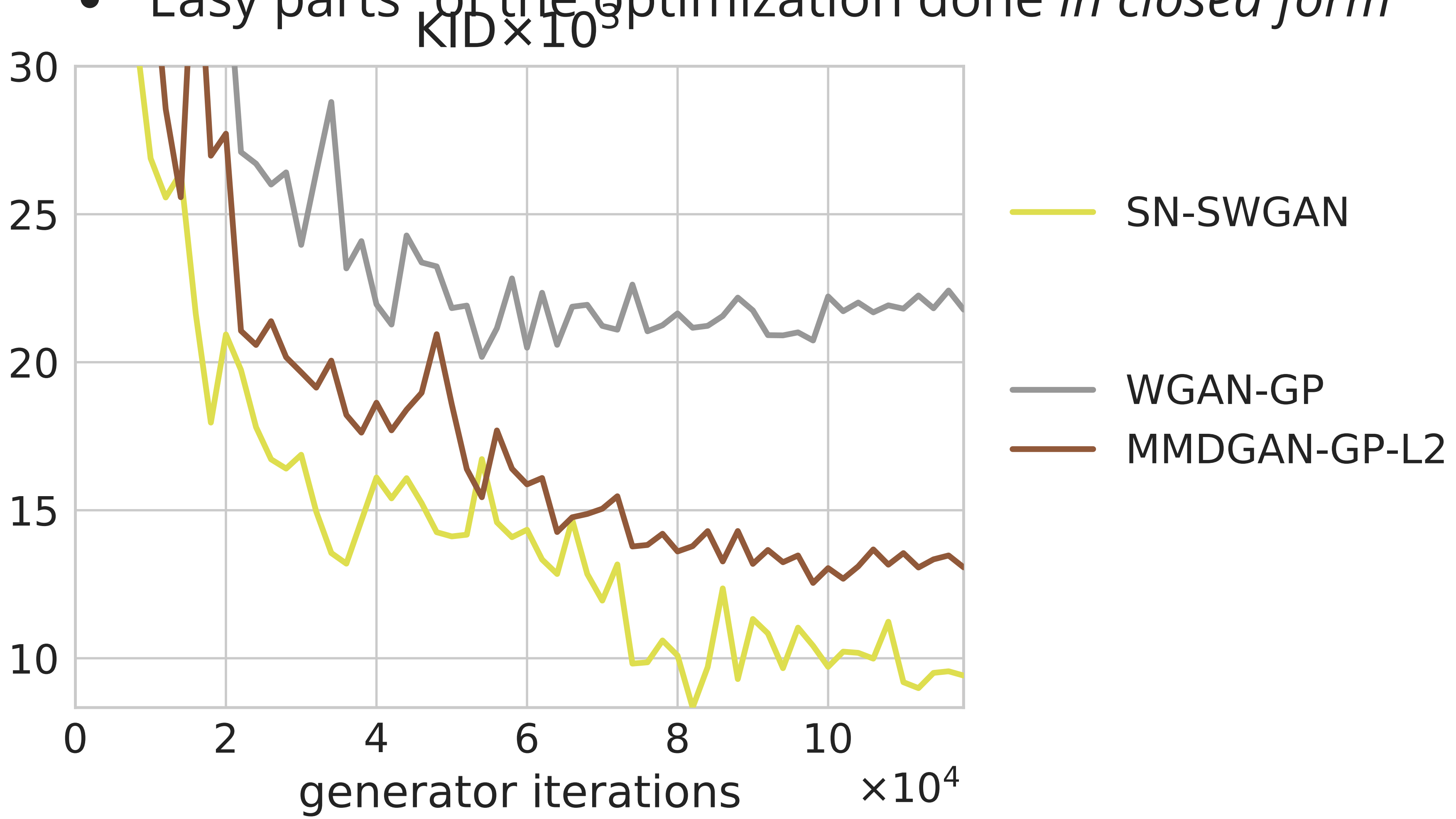
Why MMD GANs?

- "Easy parts" of the optimization done *in closed form*



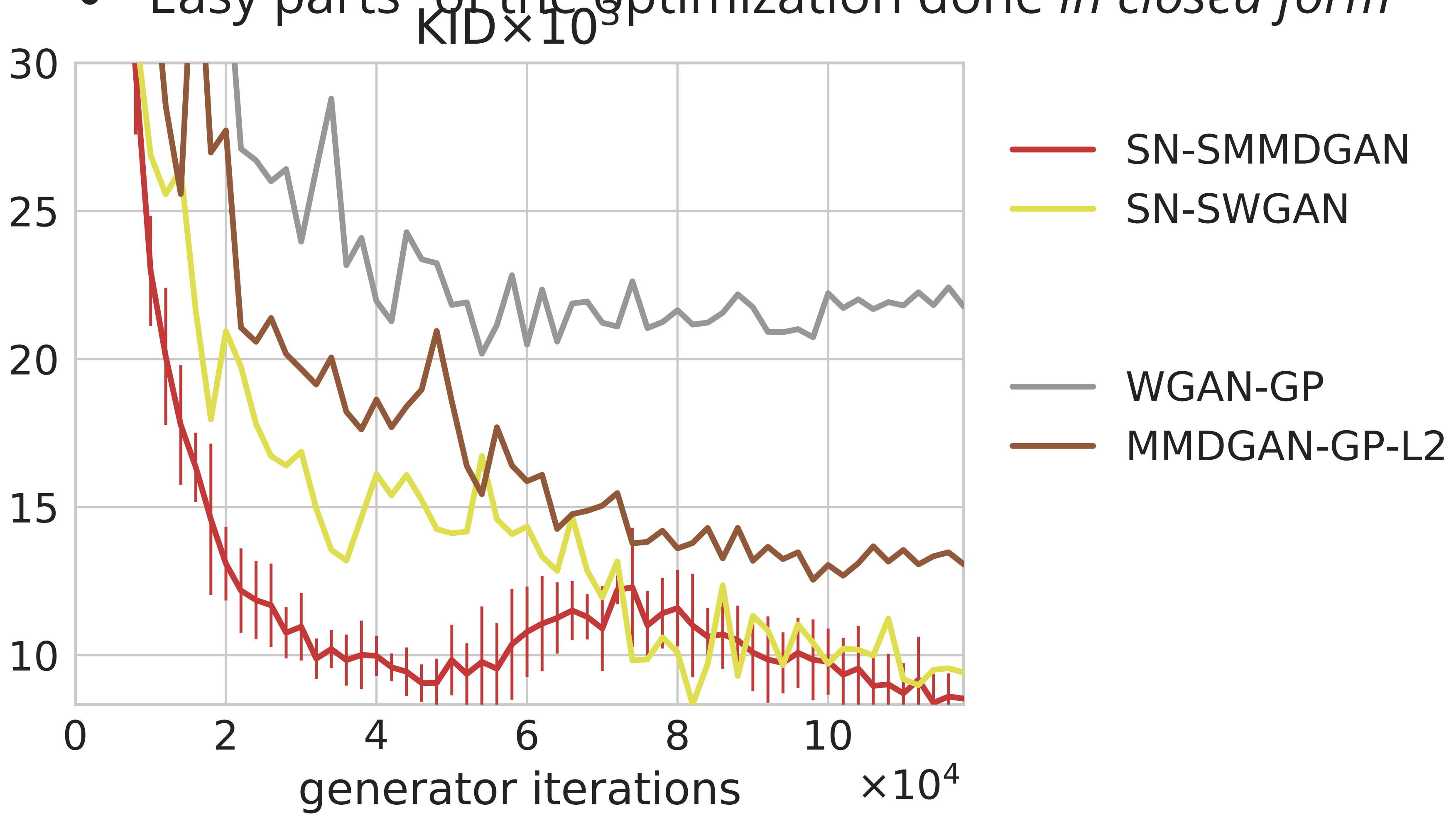
Why MMD GANs?

- “Easy parts” of the optimization done *in closed form*



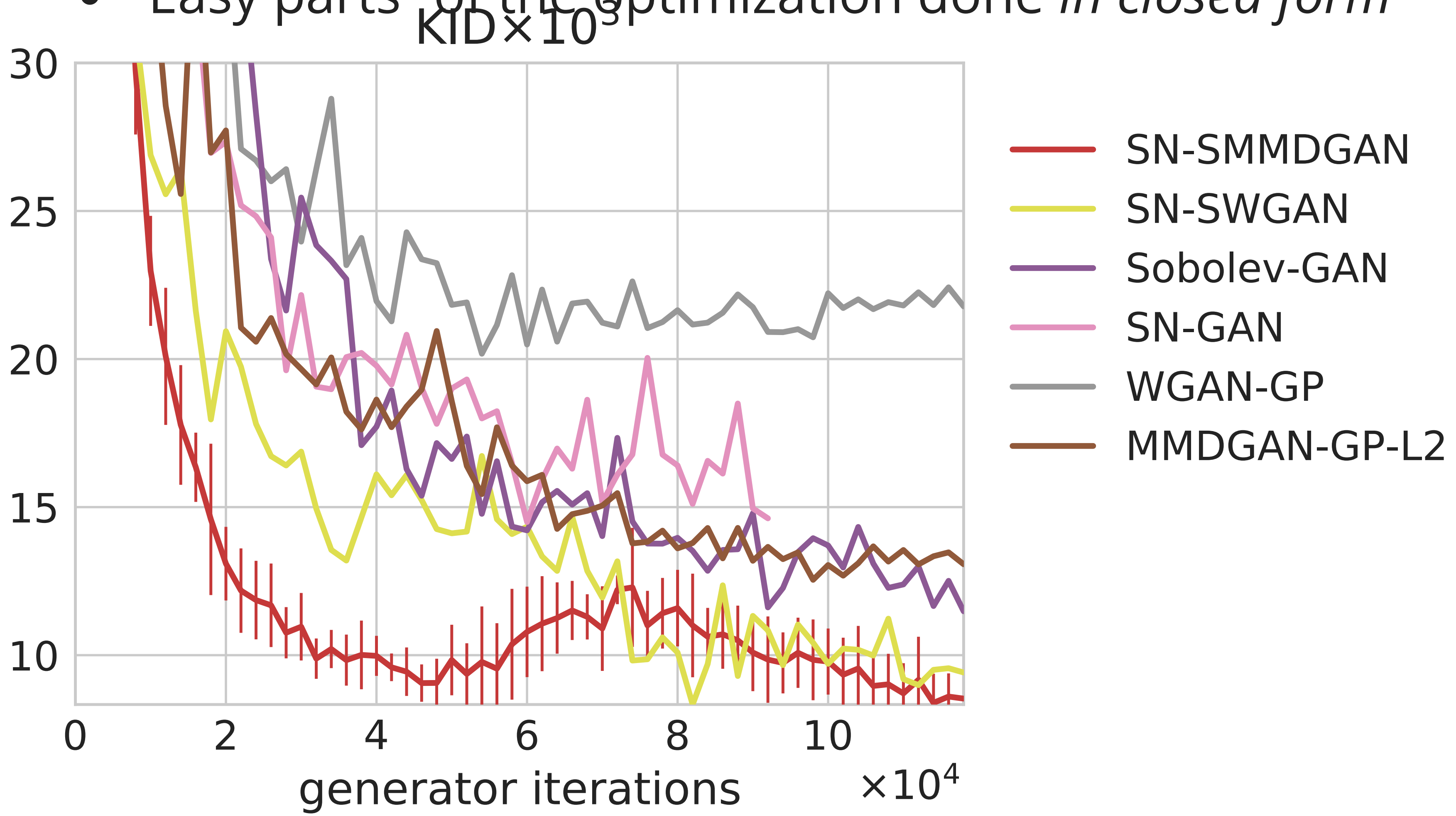
Why MMD GANs?

- "Easy parts" of the optimization done *in closed form*

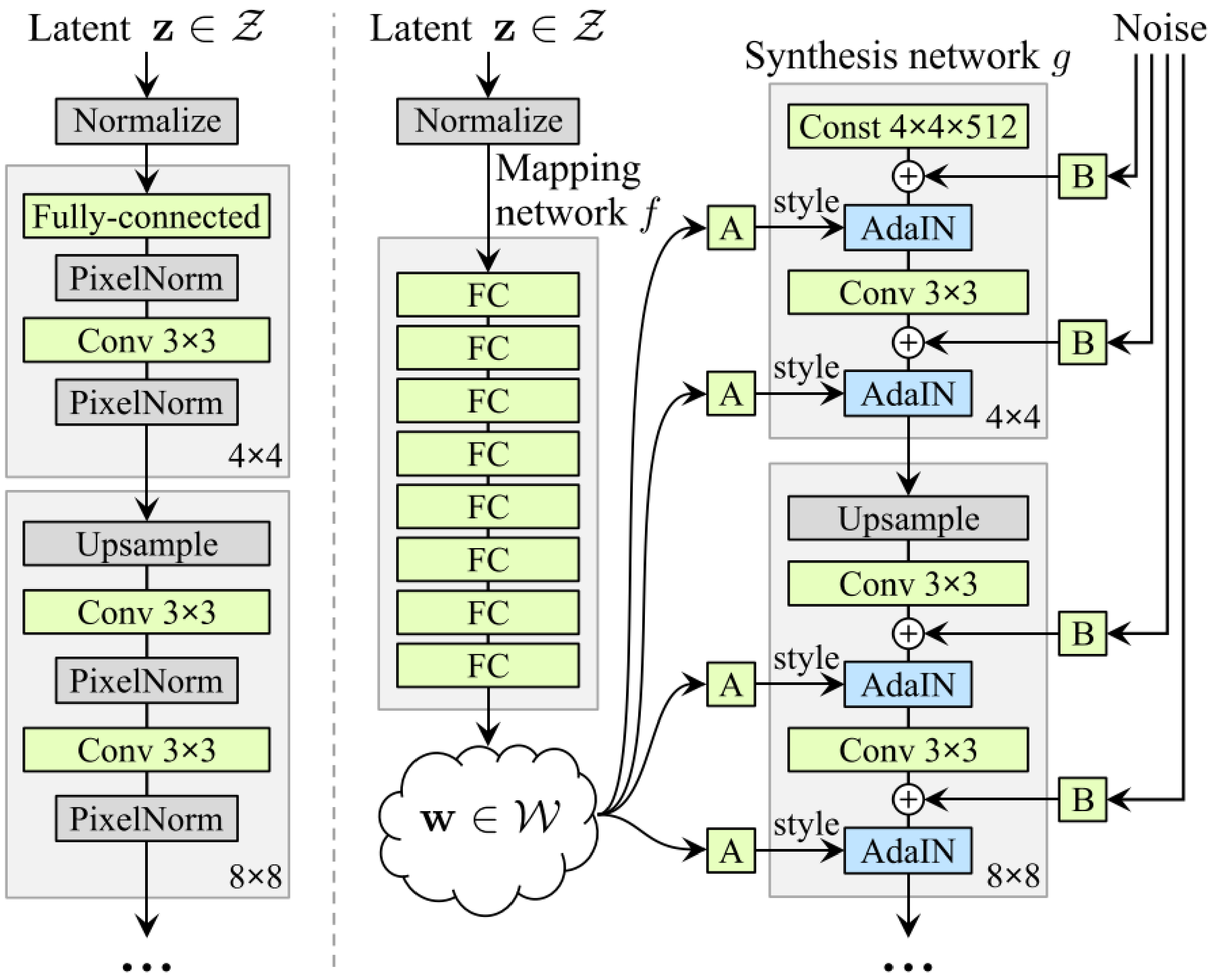


Why MMD GANs?

- “Easy parts” of the optimization done *in closed form*



StyleGANs [Karras+ 2018]



(a) Traditional (b) Style-based generator

StyleGAN: latent structure

StyleGAN: local noise

(a) Generated image (b) Stochastic variation (c) Standard deviation

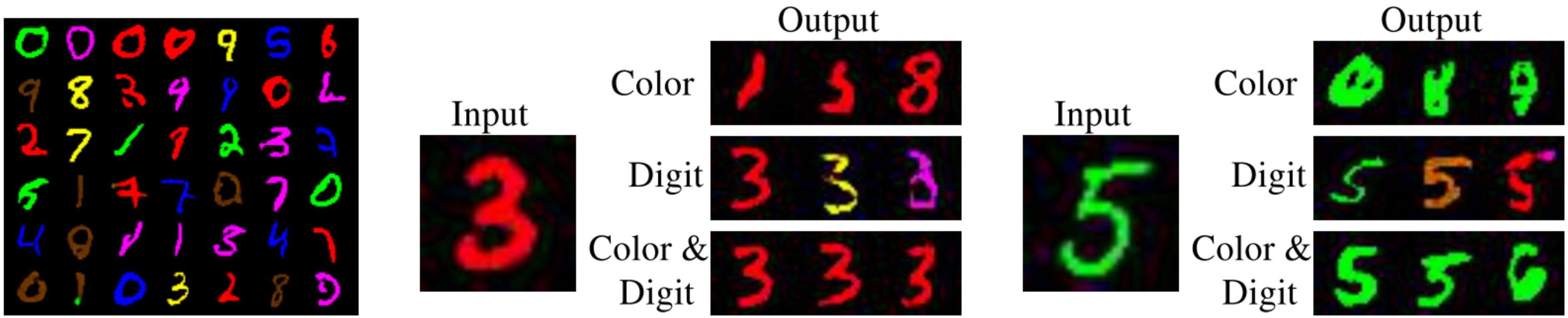
StyleGANs on a different domain [@roadrunning01]

▪

Finding samples you want [Jitkrittum+ ICML-19]

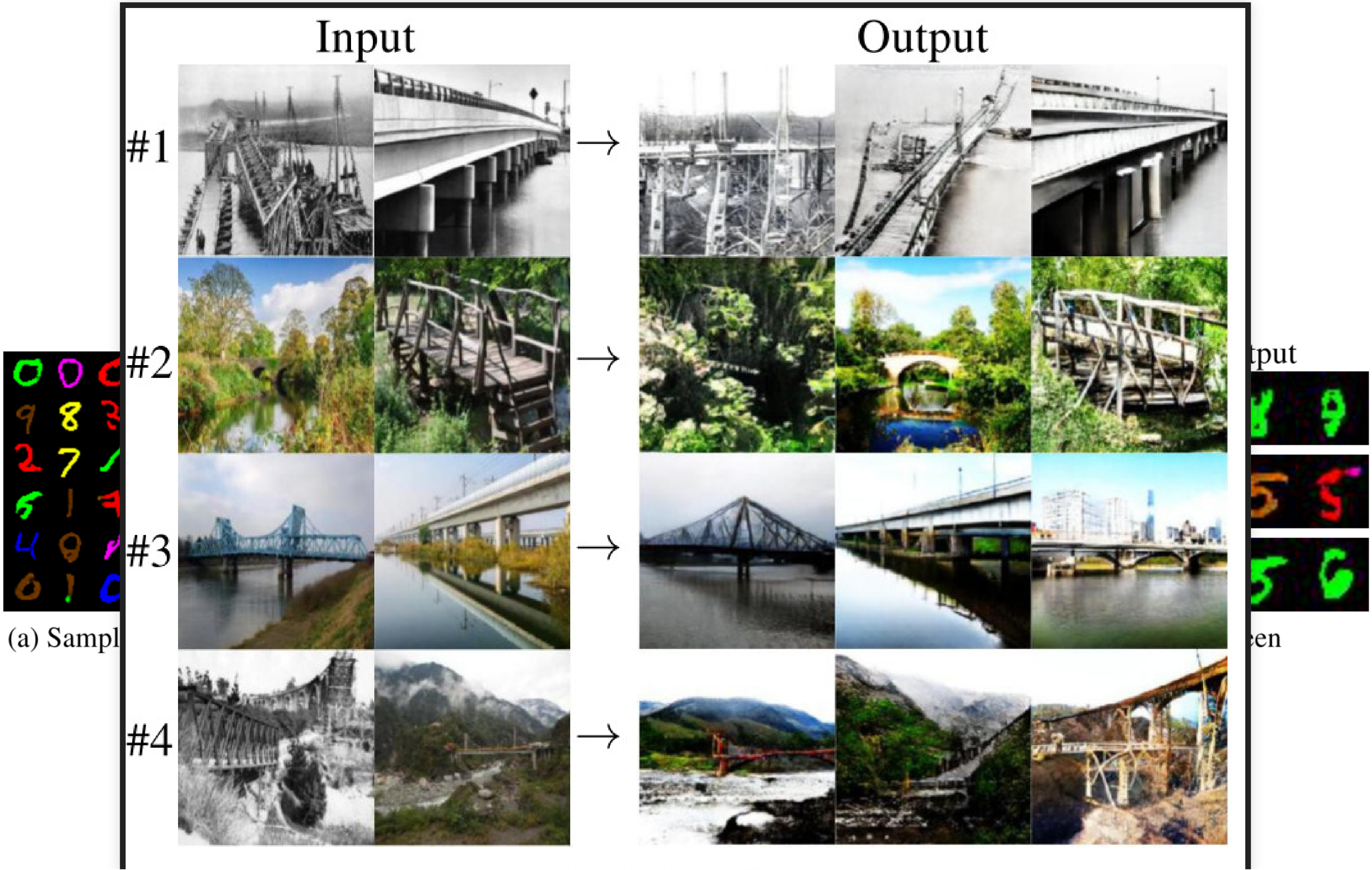
If we want to find "more samples like $\{X\}$":

$$\min_{\{Z_1,\ldots,Z_n\}} \widehat{\mathrm{MMD}}_k^2\left(\{X_i\}_{i=1}^m, \{G_\theta(Z_i)\}_{i=1}^n\right)$$



(a) Samples from DCGAN (b) Input: digit 3 in red (c) Input: digit 5 in green

Finding samples you want [Jitkrittum+ ICML-19]



Conditional GANs and BigGAN

- Conditional GANs: [Mirza+ 2014]
  - Just add a class label as input to G_θ and D_ψ
- BigGAN [Brock+ ICLR-19]: a bunch of tricks to make it huge

Image-to-image translation [Isola+ CVPR-17]

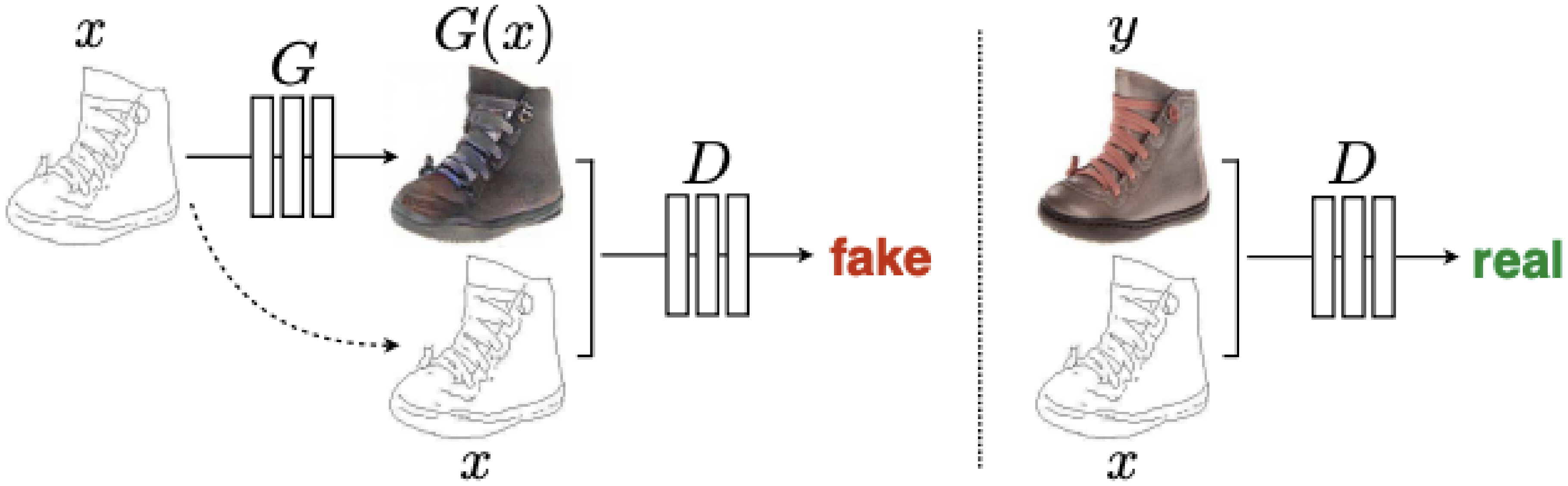
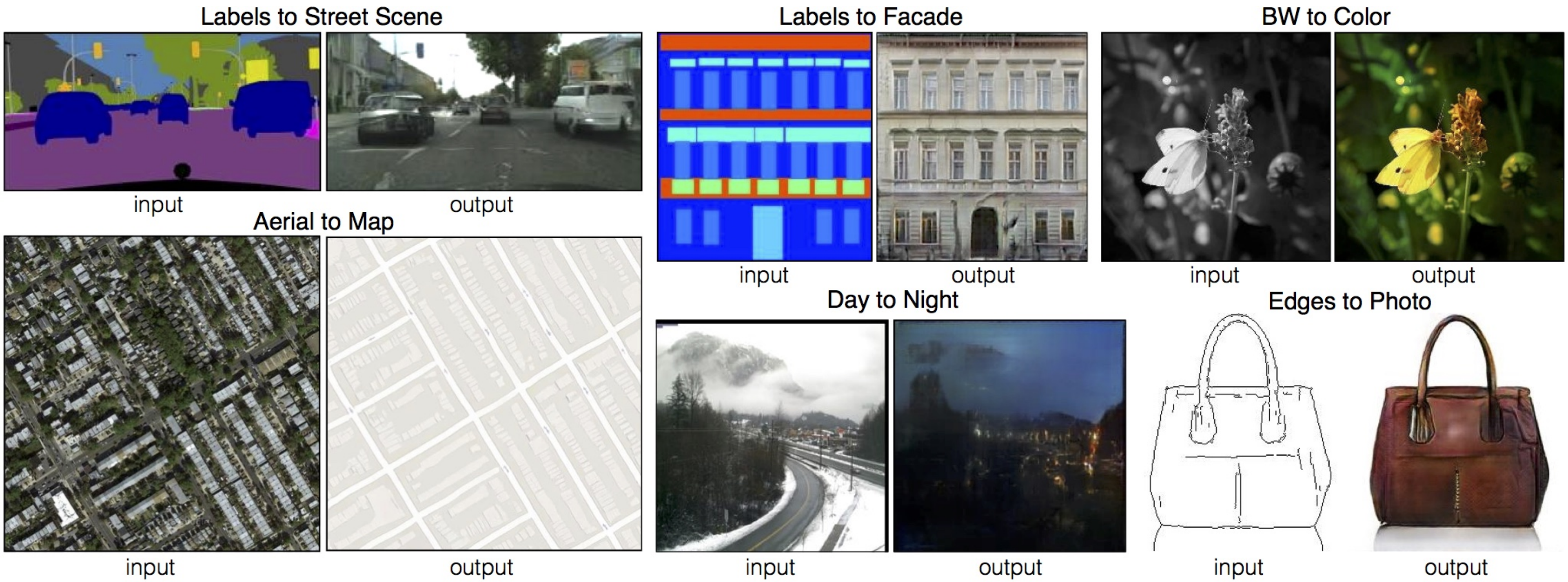
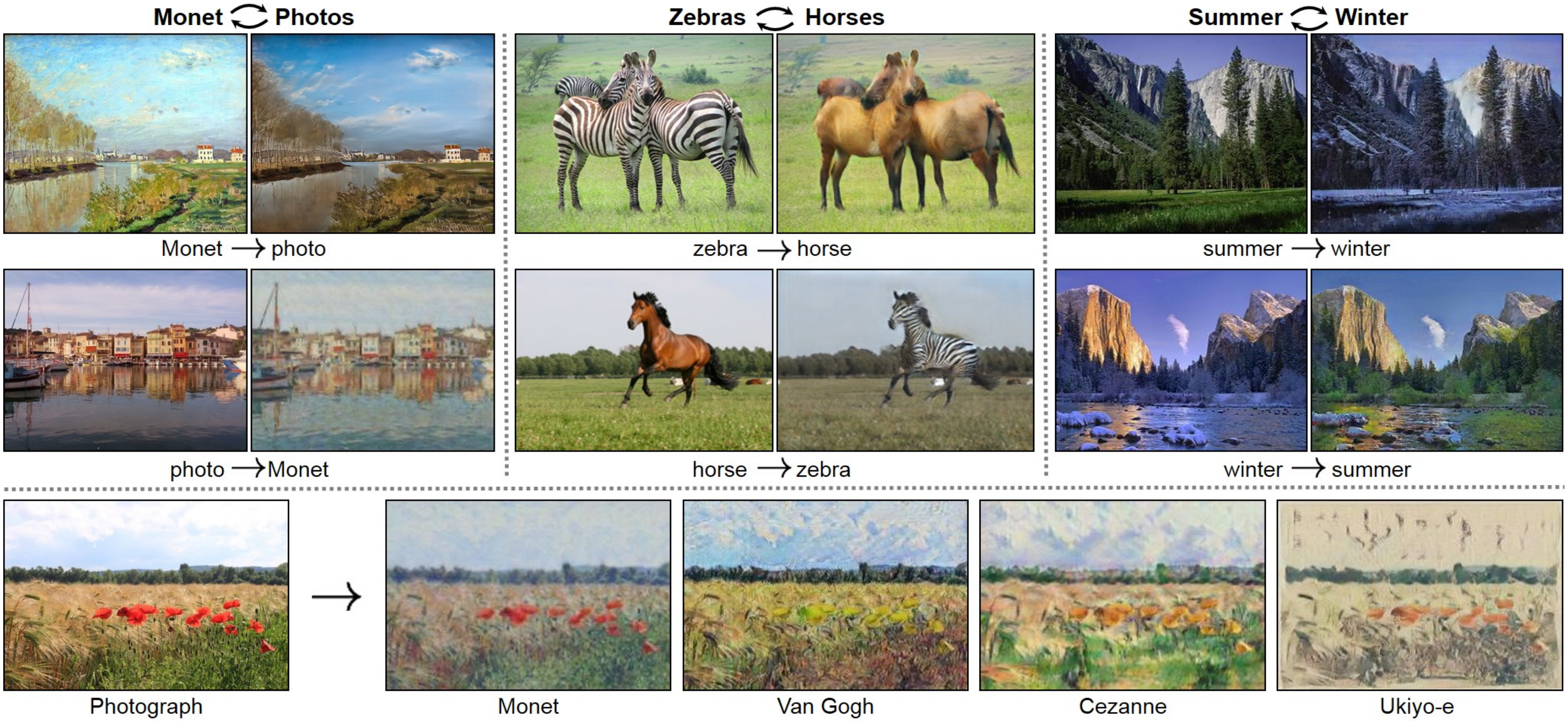


Figure 2: Training a conditional GAN to map edges→photo. The discriminator, D, learns to classify between fake (synthesized by the generator) and real {edge, photo} tuples. The generator, G, learns to fool the discriminator. Unlike an unconditional GAN, both the generator and discriminator observe the input edge map.

Image-to-image translation [Isola+ CVPR-17]



CycleGAN [Zhu+ ICCV-17]



Pose-to-image translation [Chan+ 2018]

▪

DeepFakes

More

- Optimal transport stuff:
  - Gabriel Peyré: *Optimal transport for machine learning* talk
  - Peyré and Cuturi, Computational Optimal Transport book
  - Kantorovich Initiative: `kantorovich.org`
  - Pacific Interdisciplinary Hub on Optimal Transport

- GANs / generative models…so much.