Generative Adversarial Nets and Distances Between Distributions

(from thispersondoesnotexist.com)

CPSC 340, Fall 2021

Generative Adversarial Nets and Distances Between Distributions

(from thispersondoesnotexist.com)

CPSC 340, Fall 2021

(we definitely won't get through all these slides, but let's see!)

Generative models

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

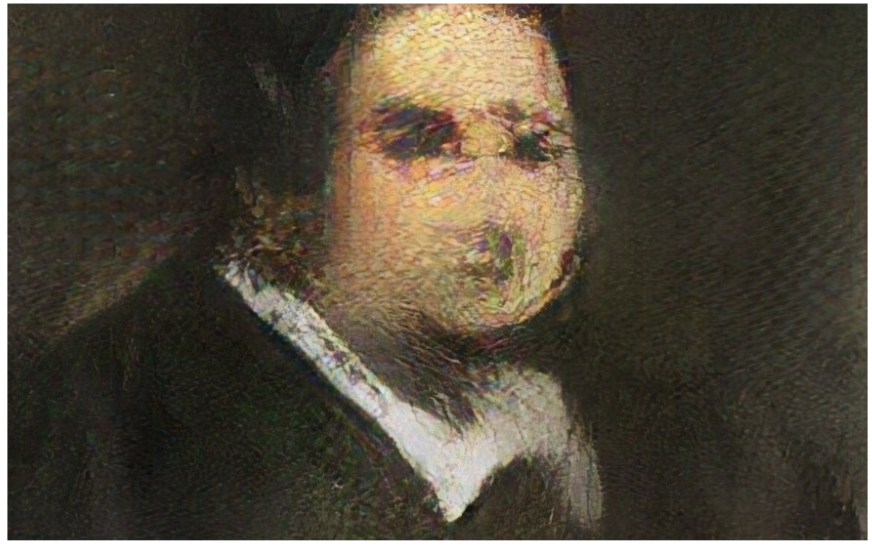
Generative models

- 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?

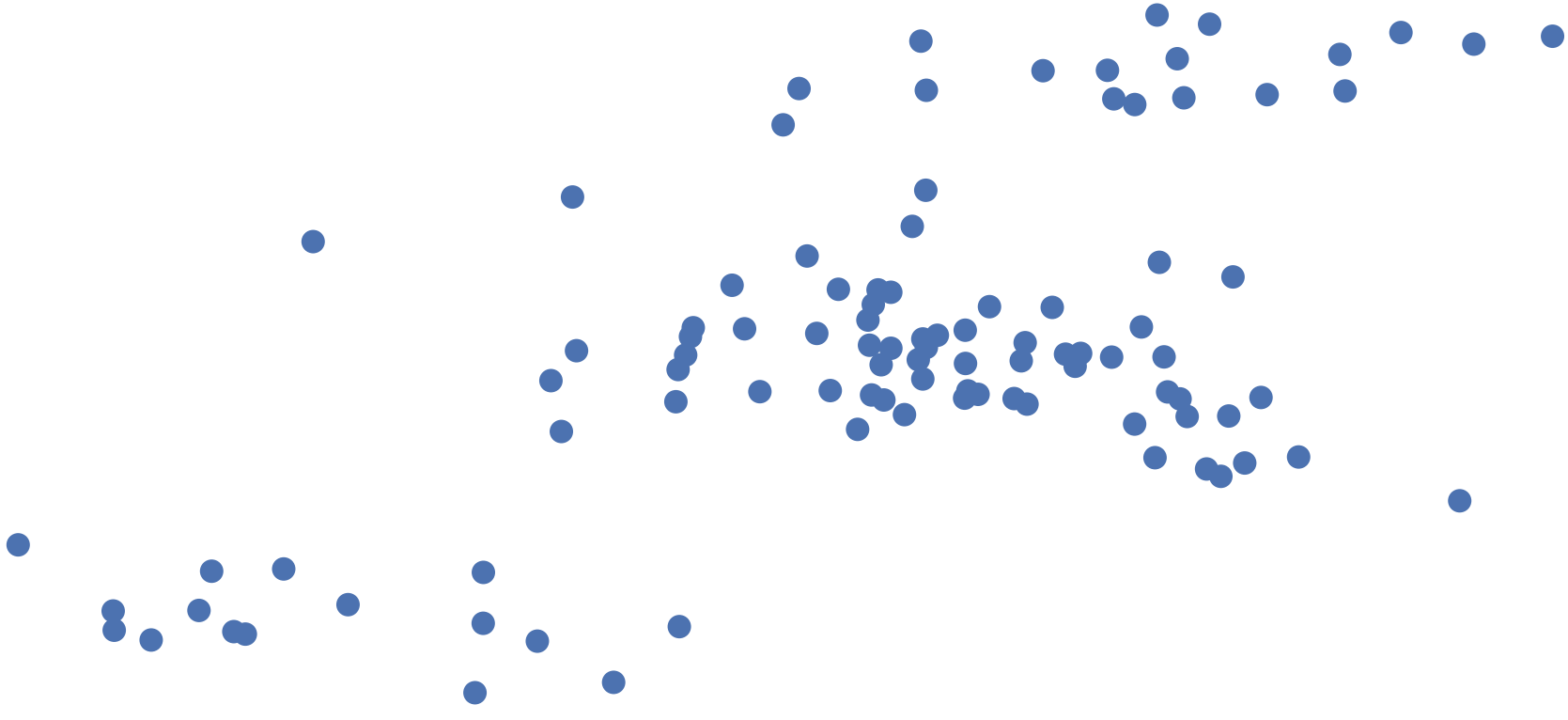
AI artwork sells for $432,500 — nearly 45 times its high estimate — as Christie's becomes the first auction house to offer a work of art created by an algorithm

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:

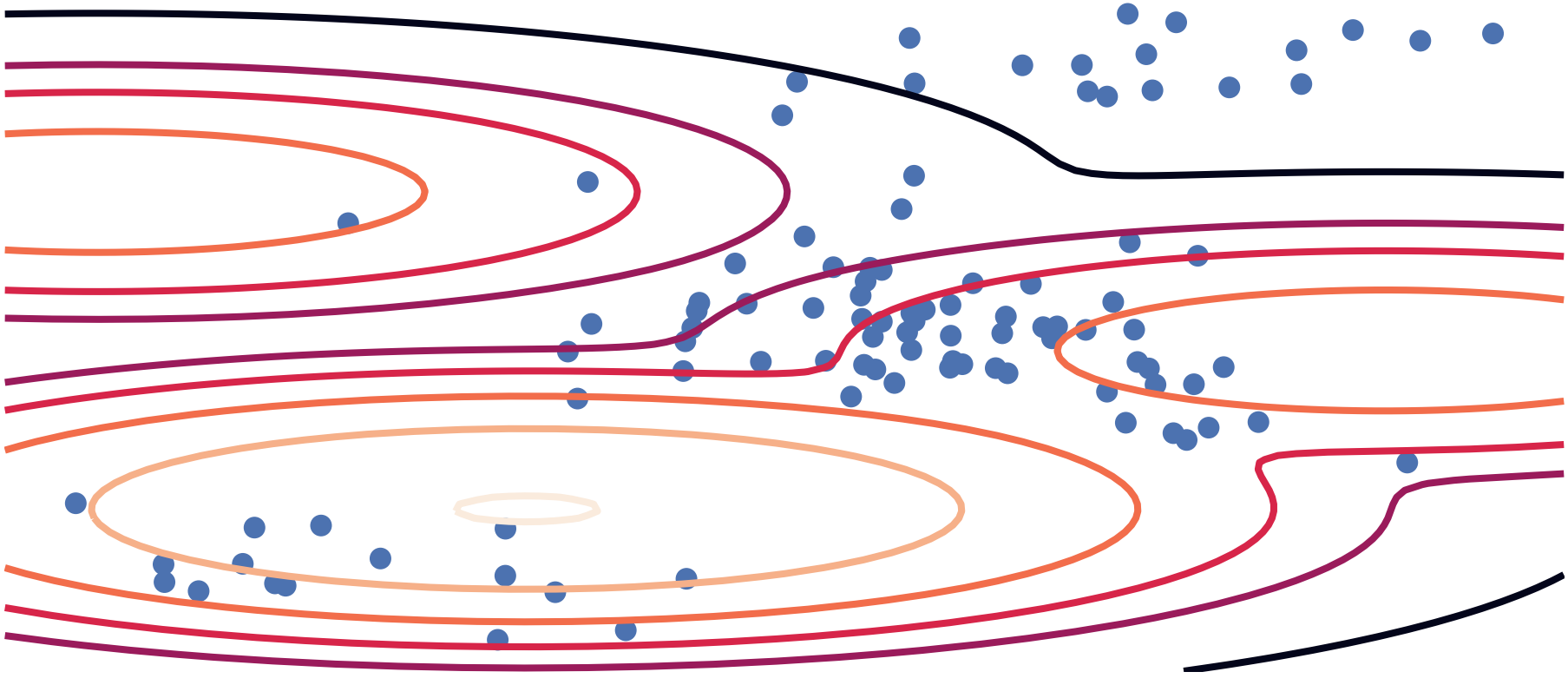
$$\min_G \max_D \mathbb{E}_x[\log(D(x))] + \mathbb{E}_z[\log(1 - D(G(z)))]$$

Image © Obvious

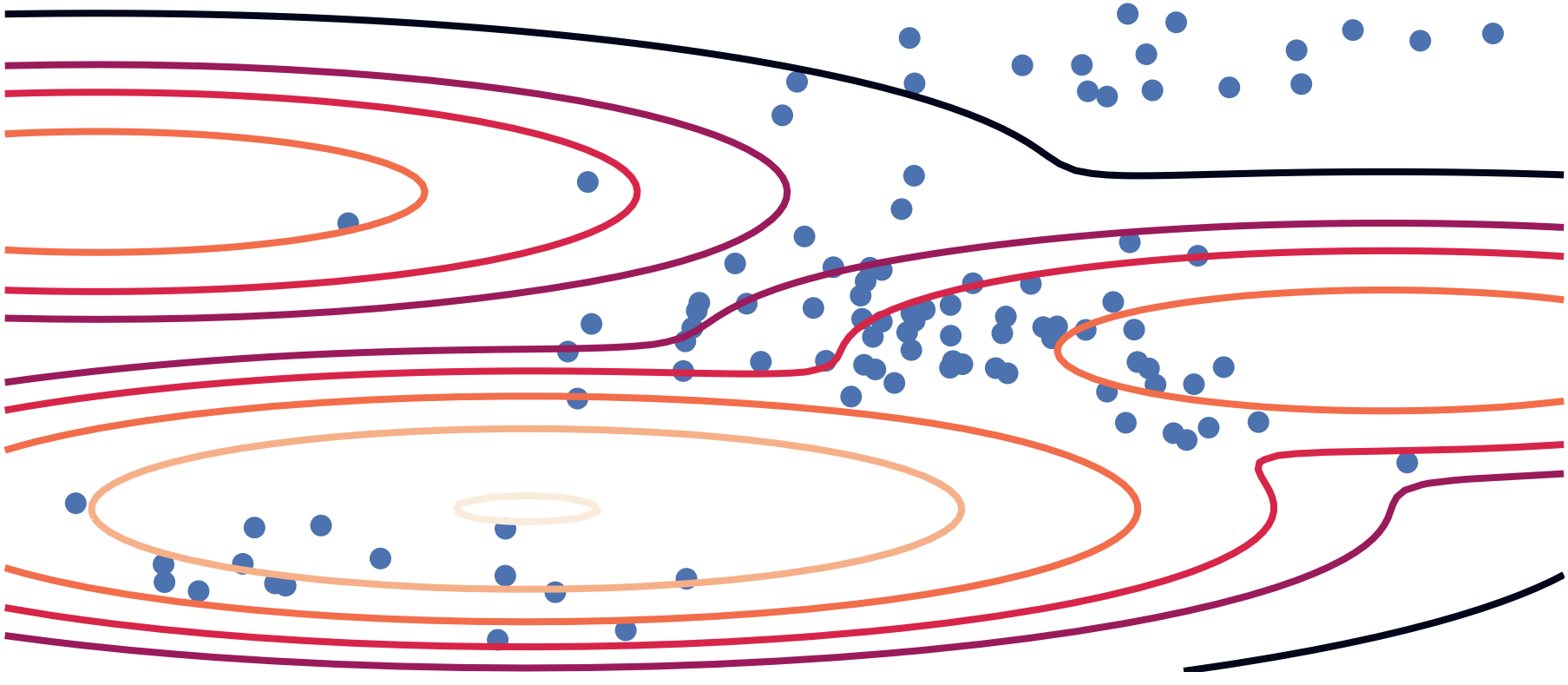
Generative models: a traditional way



Generative models: a traditional way

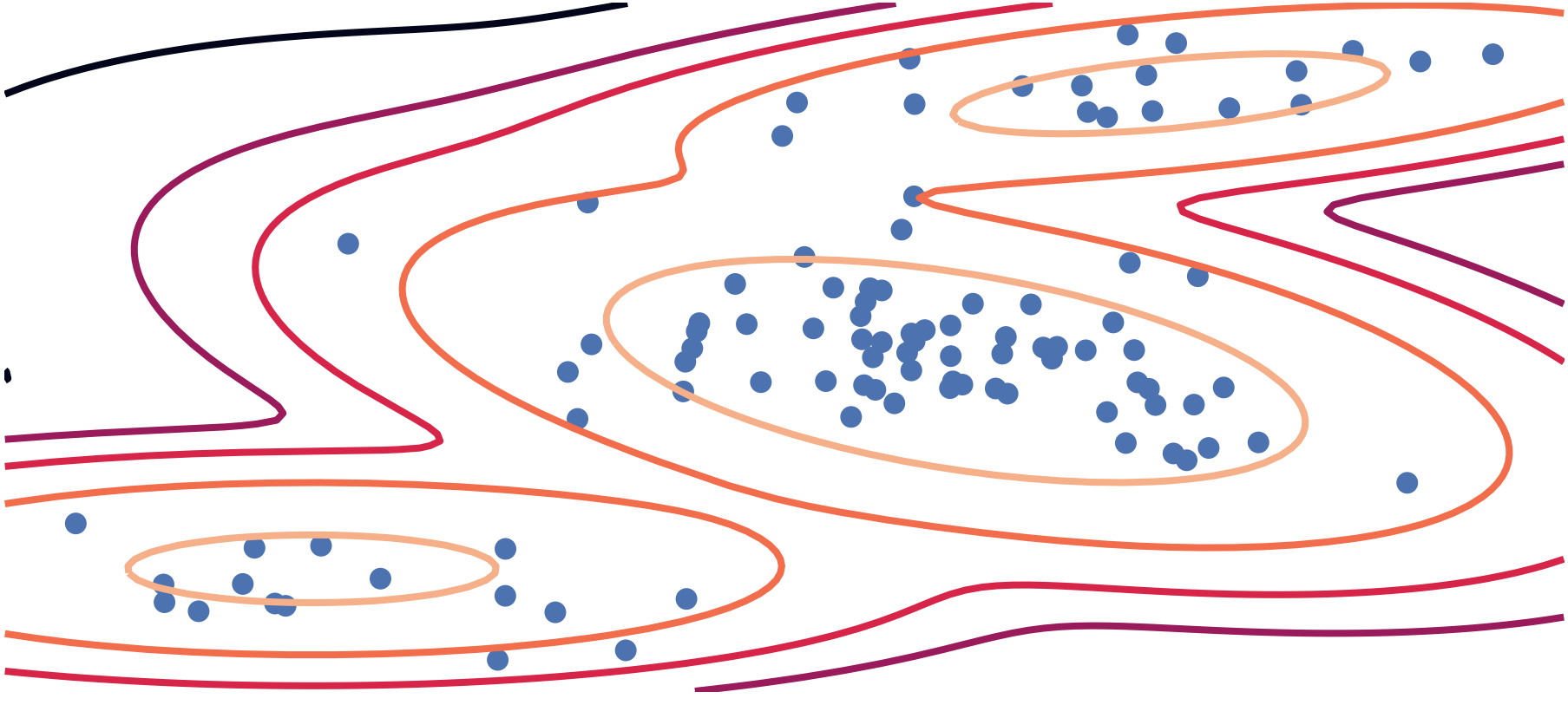


Generative models: a traditional way



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

Generative models: a traditional way



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

Kullback-Liebler (KL) divergence

- $\mathrm{KL}(\mathbb{P}\|\mathbb{Q}) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$
- A measure of how different two distributions are
- Related to entropy, etc: average number of bits to encode a sample from $\mathbb{P}$ with a code for $\mathbb{Q}$

Kullback-Liebler (KL) divergence

- $\mathrm{KL}(\mathbb{P}\|\mathbb{Q}) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$
- A measure of how different two distributions are
- Related to entropy, etc: average number of bits to encode a sample from $\mathbb{P}$ with a code for $\mathbb{Q}$
- Minimizing one direction gives you maximum likelihood:

$$\arg\min_\theta \mathrm{KL}(\mathbb{P}\|\mathbb{Q}_\theta) = \arg\min_\theta \int p(x) \log \frac{p(x)}{q_\theta(x)} \mathrm{d}x$$

Kullback-Liebler (KL) divergence

- $\mathrm{KL}(\mathbb{P}\|\mathbb{Q}) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$
- A measure of how different two distributions are
- Related to entropy, etc: average number of bits to encode a sample from $\mathbb{P}$ with a code for $\mathbb{Q}$
- Minimizing one direction gives you maximum likelihood:

$$\begin{aligned}\arg\min_\theta \mathrm{KL}(\mathbb{P}\|\mathbb{Q}_\theta) &= \arg\min_\theta \int p(x) \log \frac{p(x)}{q_\theta(x)} \mathrm{d}x \\ &= \arg\min_\theta -\mathrm{H}[\mathbb{P}] - \mathop{\mathbb{E}}_{X\sim\mathbb{P}}[\log q_\theta(X)]\end{aligned}$$

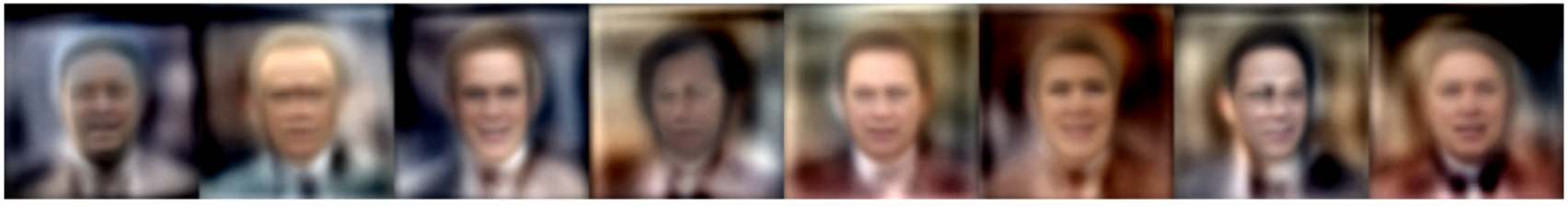
Kullback-Liebler (KL) divergence

- $\mathrm{KL}(\mathbb{P}\|\mathbb{Q}) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$
- A measure of how different two distributions are
- Related to entropy, etc: average number of bits to encode a sample from $\mathbb{P}$ with a code for $\mathbb{Q}$
- Minimizing one direction gives you maximum likelihood:

$$\begin{aligned}
\arg\min_\theta \mathrm{KL}(\mathbb{P}\|\mathbb{Q}_\theta) &= \arg\min_\theta \int p(x) \log \frac{p(x)}{q_\theta(x)} \mathrm{d}x \\
&= \arg\min_\theta - \mathrm{H}[\mathbb{P}] - \mathop{\mathbb{E}}_{X\sim\mathbb{P}}[\log q_\theta(X)] \\
&= \arg\max_\theta \mathop{\mathbb{E}}_{X\sim\mathbb{P}}[\log q_\theta(X)]
\end{aligned}$$

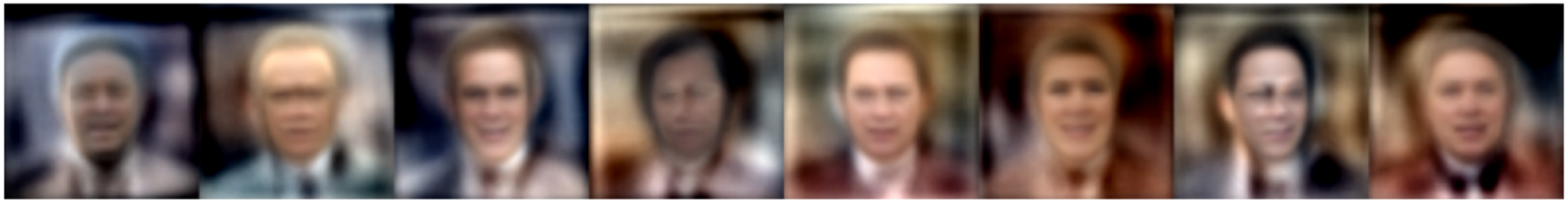
Traditional models for images

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



Traditional models for images

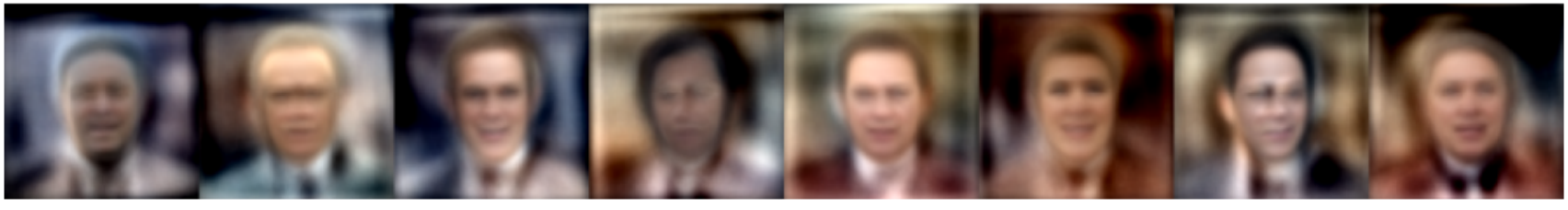
- 1987-style generative model of faces (Eigenface via Alex Egg)



- 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

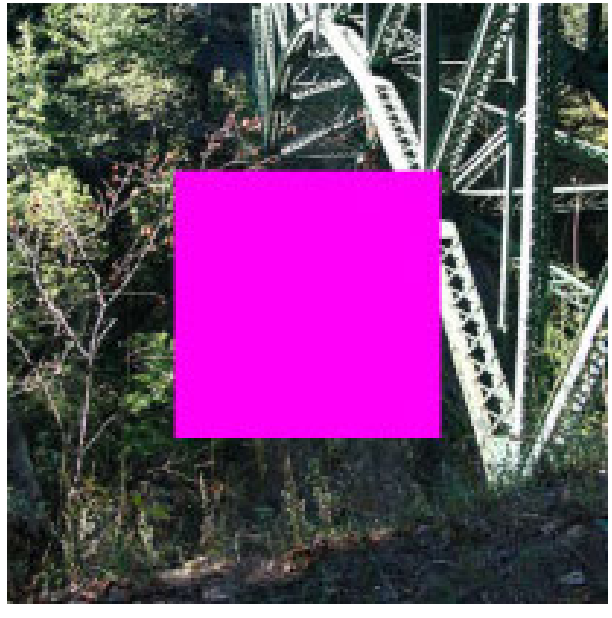
- One use case of generative models is inpainting [Harry Yang]:

A hard case for traditional approaches

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

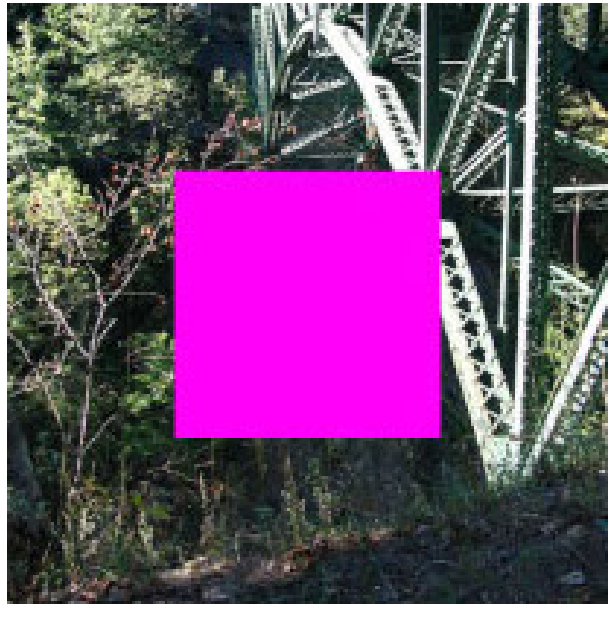
A hard case for traditional approaches

- 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]:



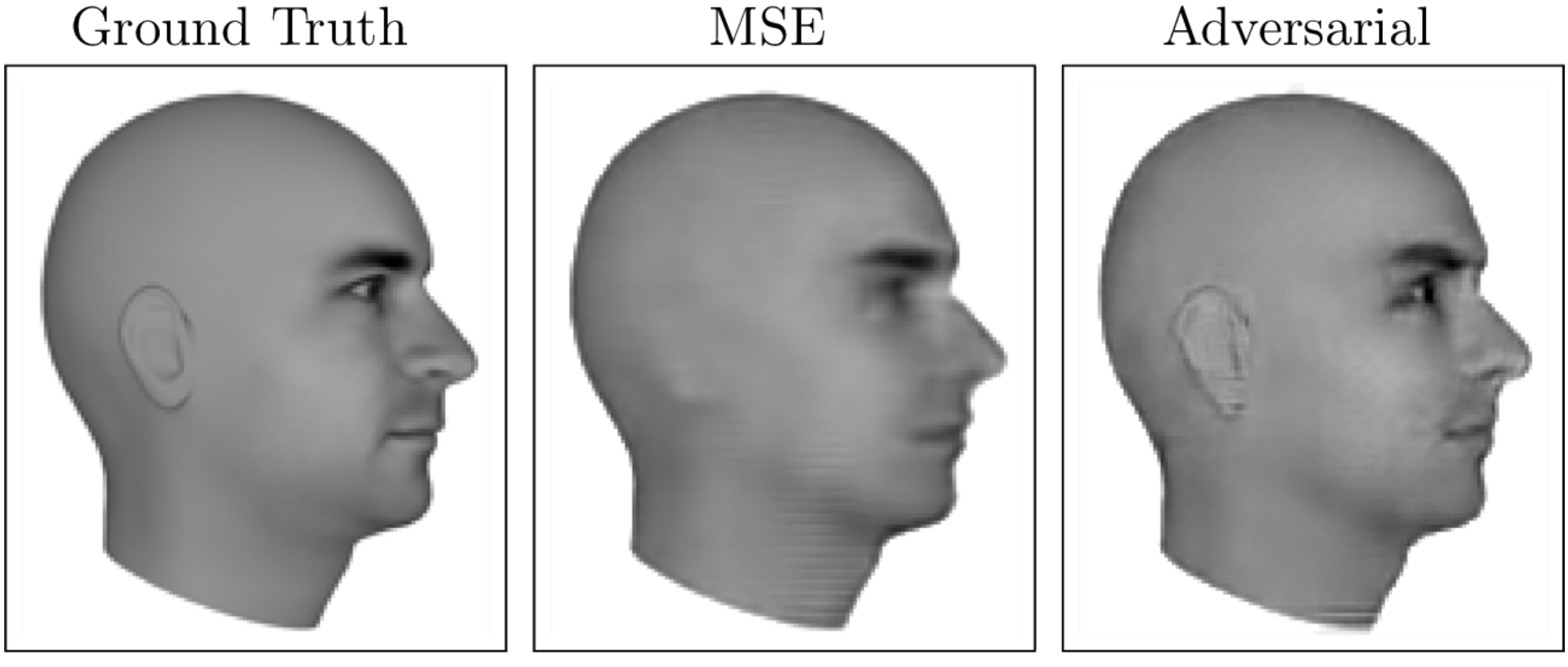
- L_2 loss / Gaussians will pick the *mean* of possibilities

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

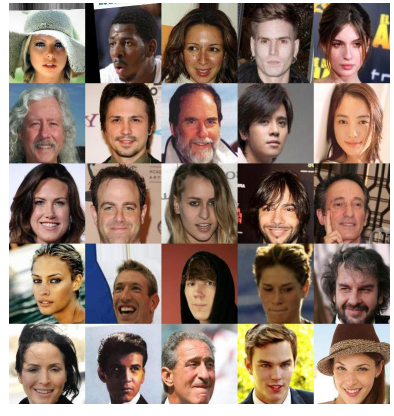
Is this real?

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

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

Discriminator



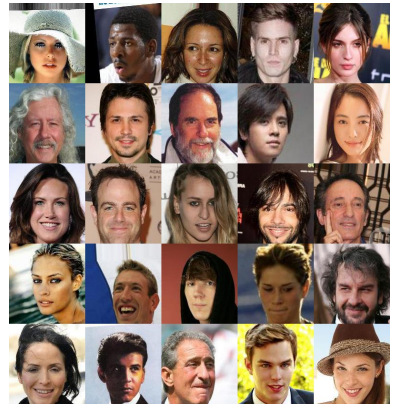
Is this real?

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

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

Discriminator



Is this real?

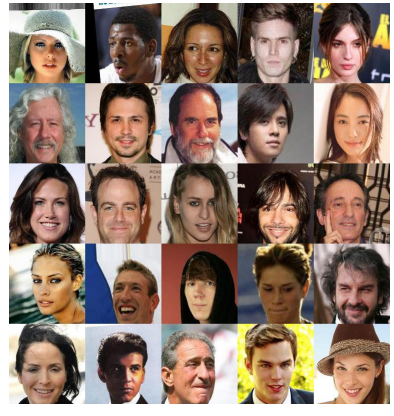
No way! $\mathbf{Pr(real) = 0.03}$

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

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

Discriminator



Is this real?

:(I'll try harder...

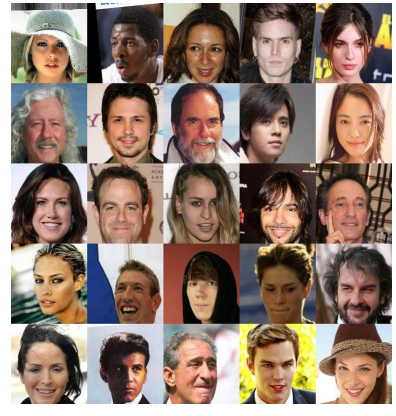
No way! $\Pr(\text{real}) = 0.03$

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

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

Discriminator



Is this real?

:(I'll try harder...

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

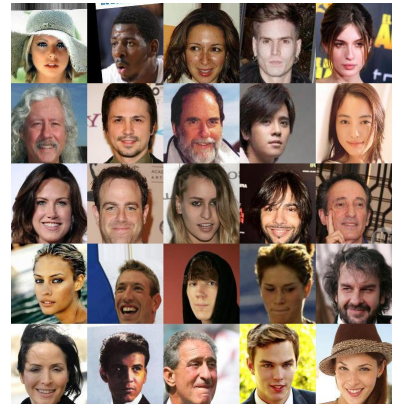
⋮

Trick a discriminator [Goodfellow+ NeurIPS-14]

Target ($\mathbb{P}$)

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

Discriminator



Is this real?

:(I'll try harder...

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

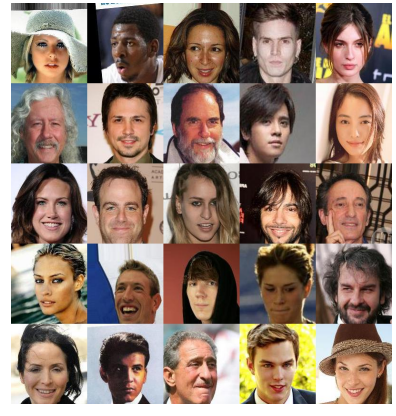
Is this real?

Trick a discriminator [Goodfellow+ NeurIPS-14]

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

Target ($\mathbb{P}$)

Discriminator



Is this real?

:(I'll try harder...

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

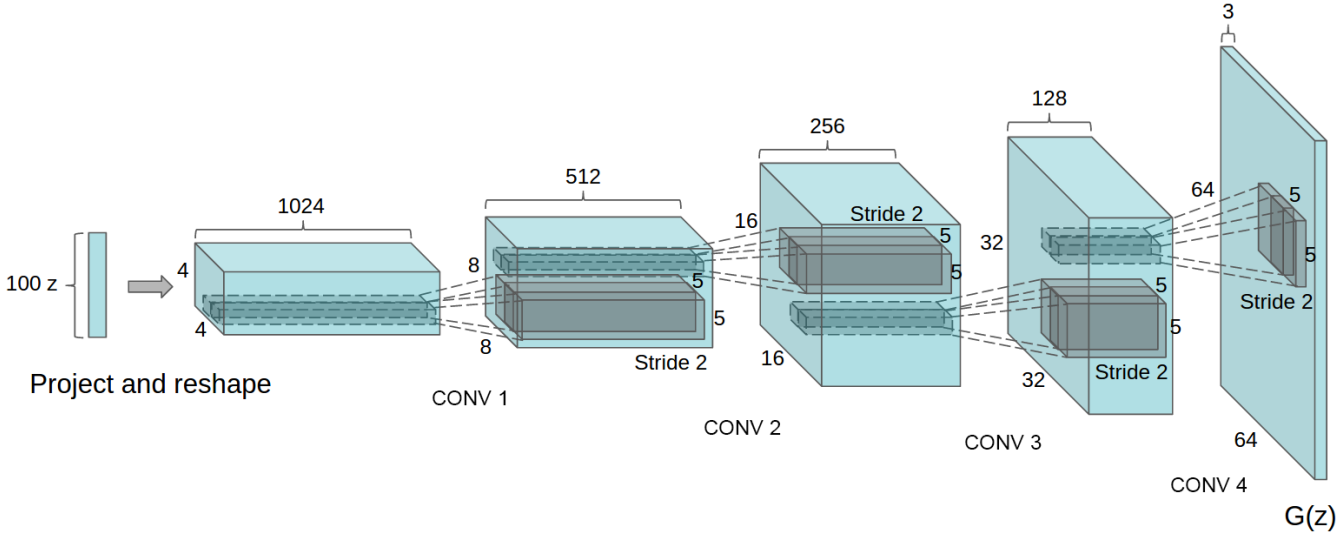
⋮

Is this real?

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

Generator networks

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



[Radford+ ICLR-16]

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

GANs in equations

- Tricking the discriminator:

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

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:

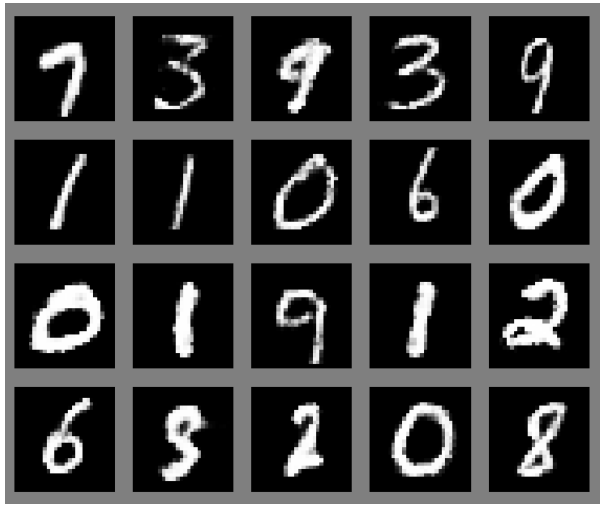
$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [\log D_\psi(X)] + \frac{1}{2} \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}_\theta} [\log(1 - D_\psi(Y))]$$

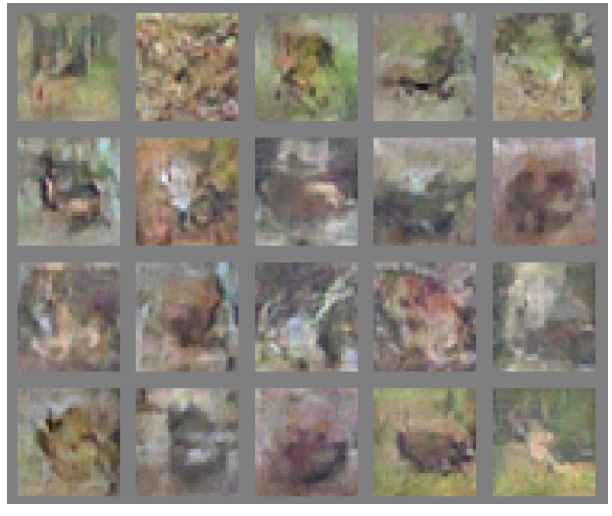
- Using the generator network for $\mathbb{Q}_\theta$:

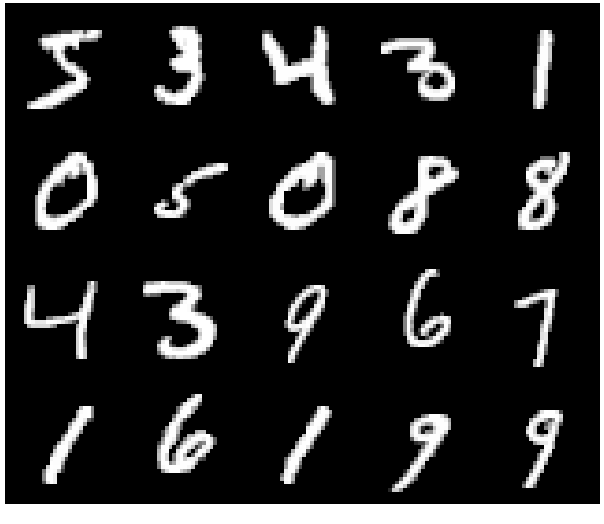
$$\min_{\theta} \max_{\psi} \frac{1}{2} \mathop{\mathbb{E}}_{X \sim \mathbb{P}} [\log D_\psi(X)] + \frac{1}{2} \mathop{\mathbb{E}}_{Z \sim \mathbb{Z}} [\log(1 - D_\psi(G_\theta(Z)))]$$

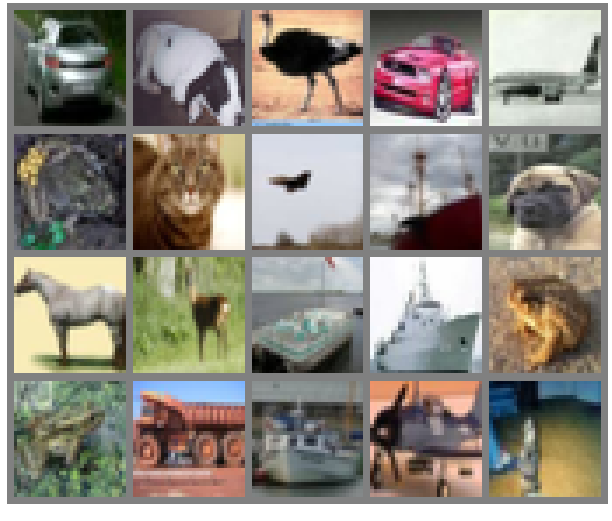
- Can do alternating gradient descent!

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





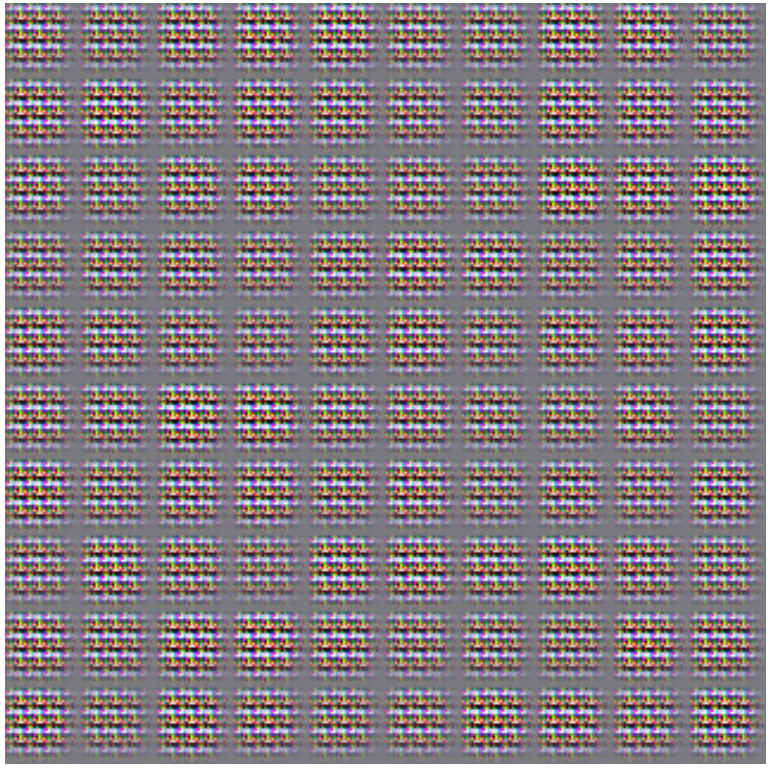




DCGAN results [Radford+ ICLR-16]

Training instability

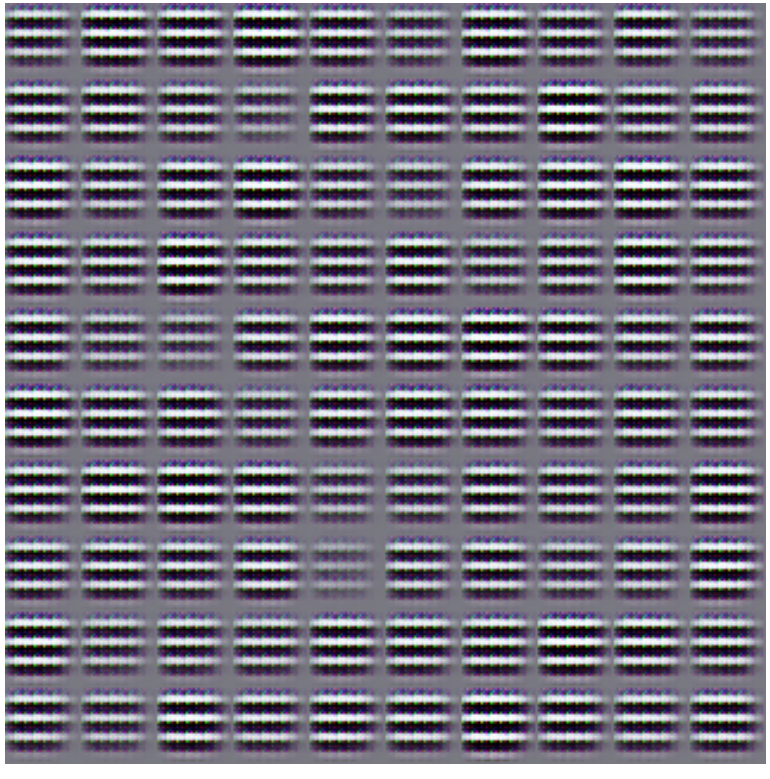
Running code from [Salimans+ NeurIPS-16]:



Run 1, epoch 1

Training instability

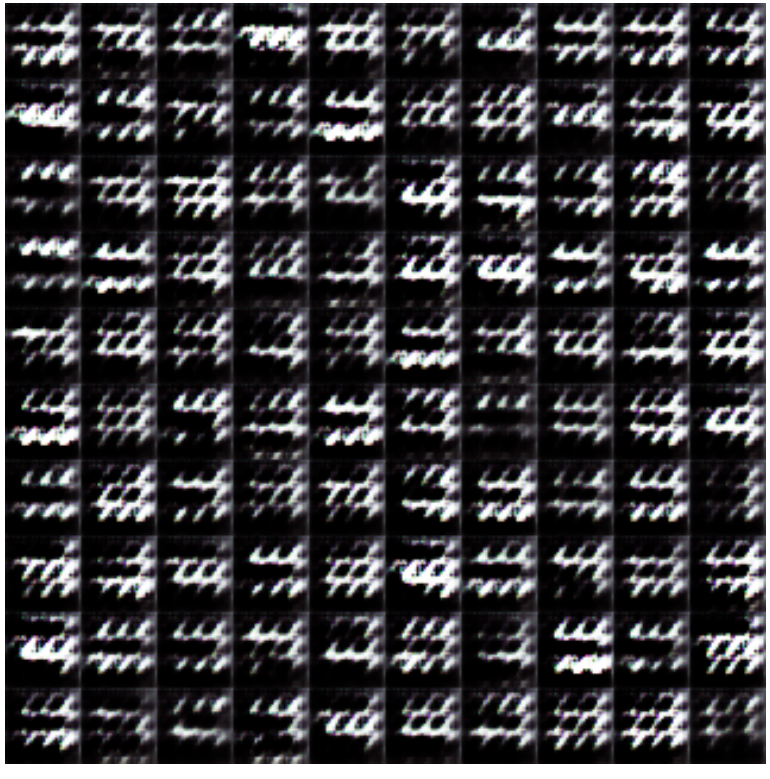
Running code from [Salimans+ NeurIPS-16]:



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

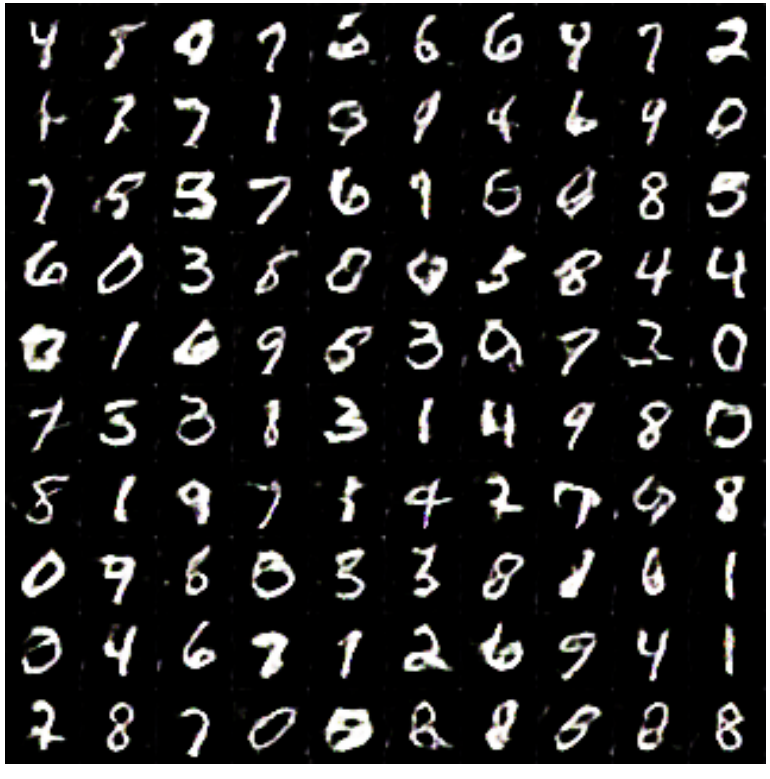
Training instability

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

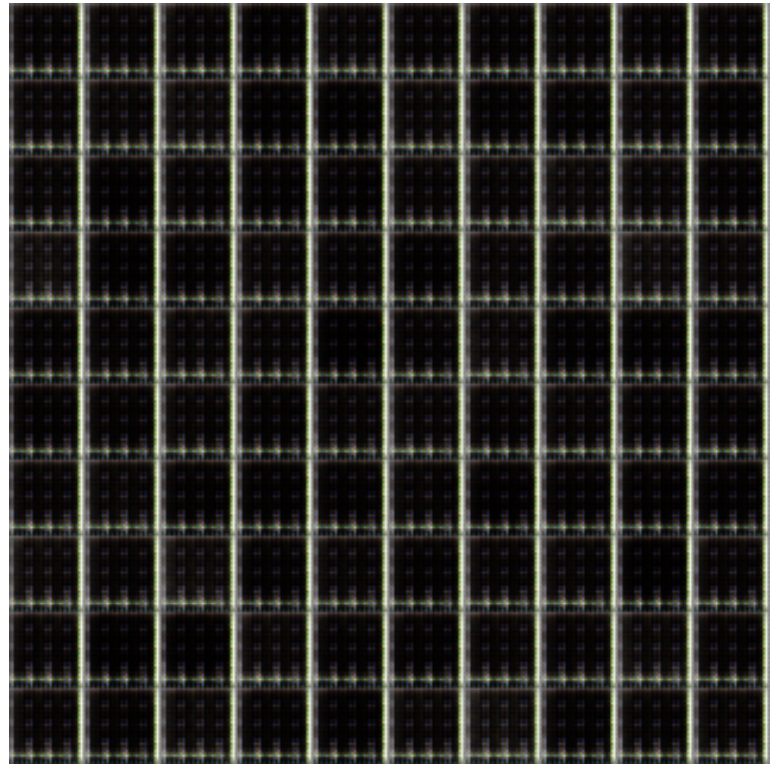
Running code from [Salimans+ NeurIPS-16]:

Run 1, epoch 900

Training instability

Running code from [Salimans+ NeurIPS-16]:

Run 1, epoch 900

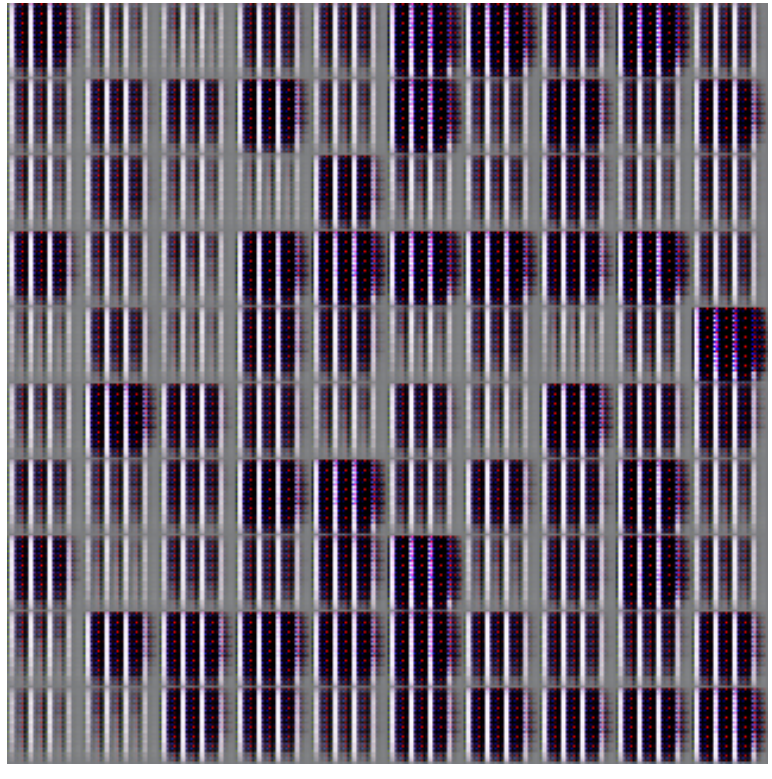


Run 2, epoch 1

Training instability

Running code from [Salimans+ NeurIPS-16]:

Run 1, epoch 900

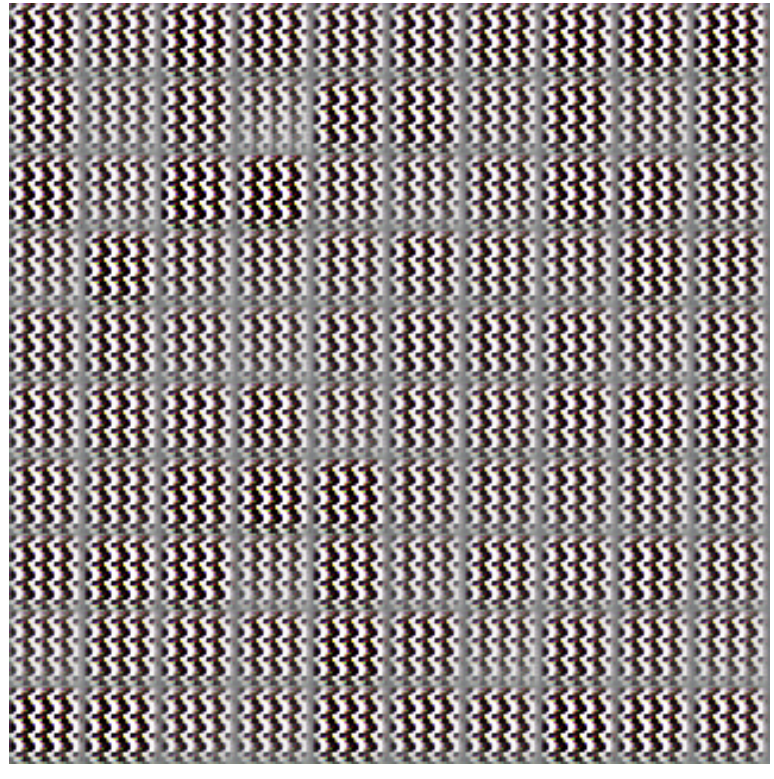


Run 2, epoch 2

Training instability

Running code from [Salimans+ NeurIPS-16]:

Run 1, epoch 900

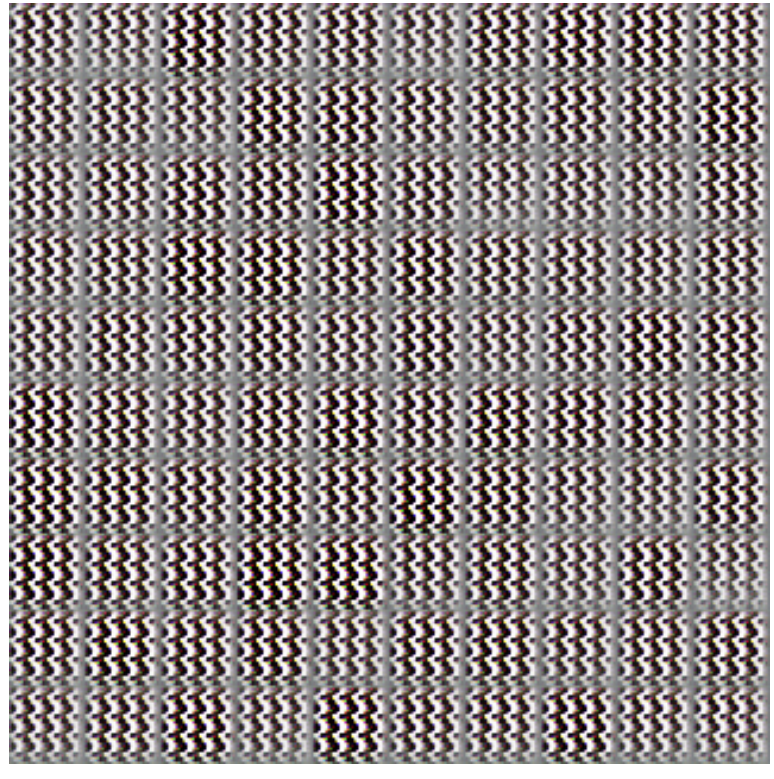


Run 2, epoch 3

Training instability

Running code from [Salimans+ NeurIPS-16]:

Run 1, epoch 900

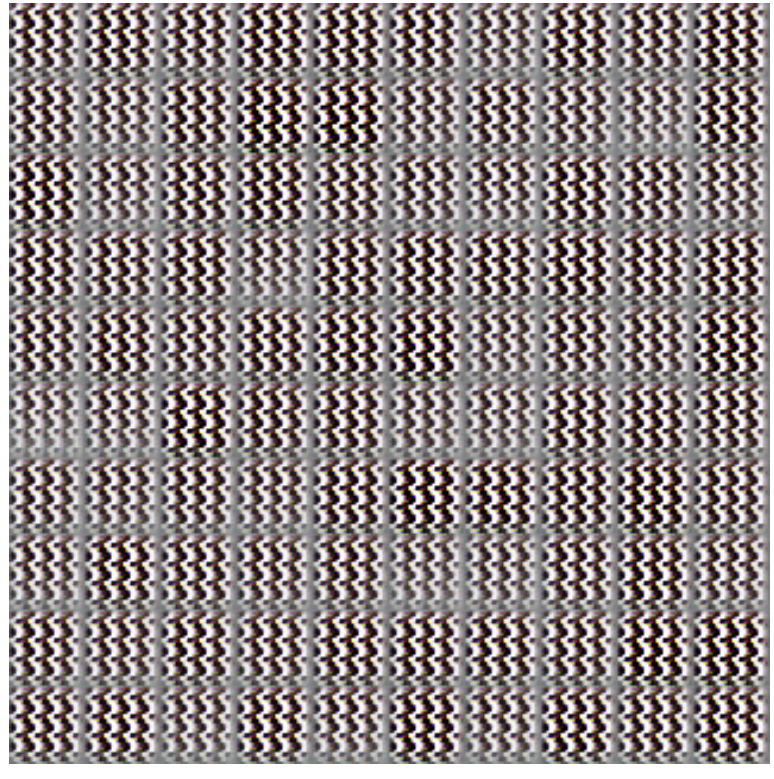


Run 2, epoch 4

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

Is this real?

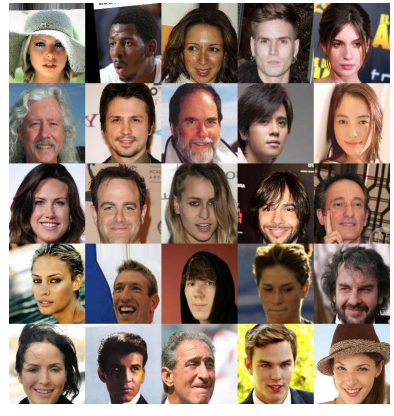
Discriminator

Discriminator point of view

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

Target ($\mathbb{P}$)

Discriminator



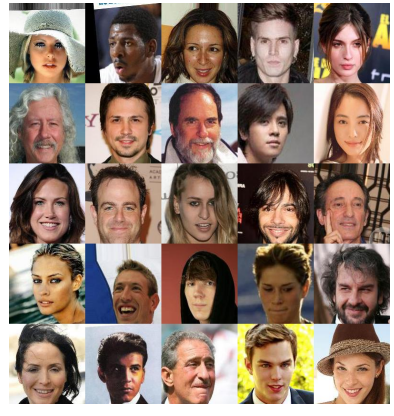
Is this real?

Discriminator point of view

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

Target ($\mathbb{P}$)

Discriminator



Is this real?

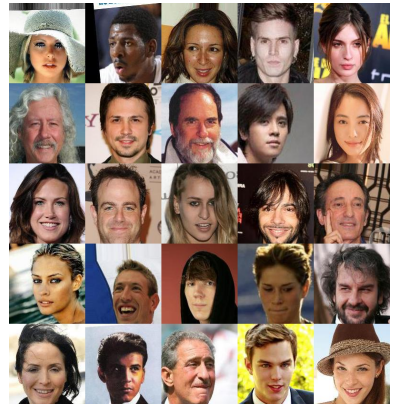
No way! $\mathbf{Pr(real) = 0.00}$

Discriminator point of view

Target ($\mathbb{P}$)

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

Discriminator



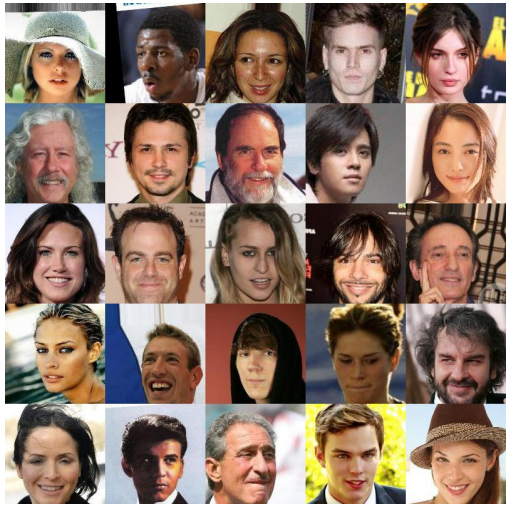
Is this real?

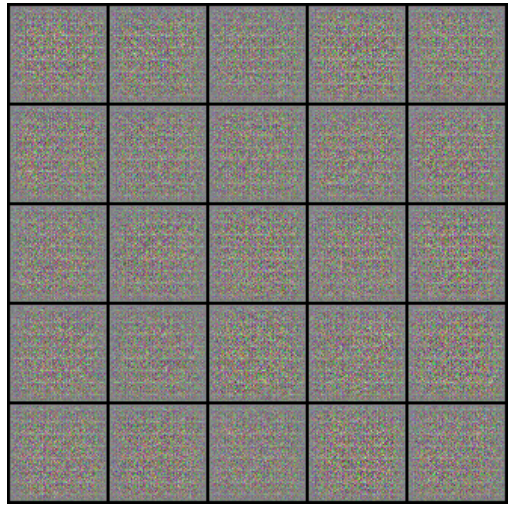
No way! $\mathbf{Pr(real) = 0.00}$

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

How likely is disjoint support?

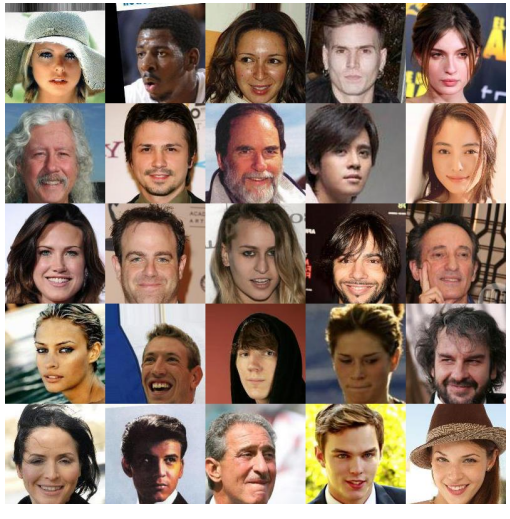
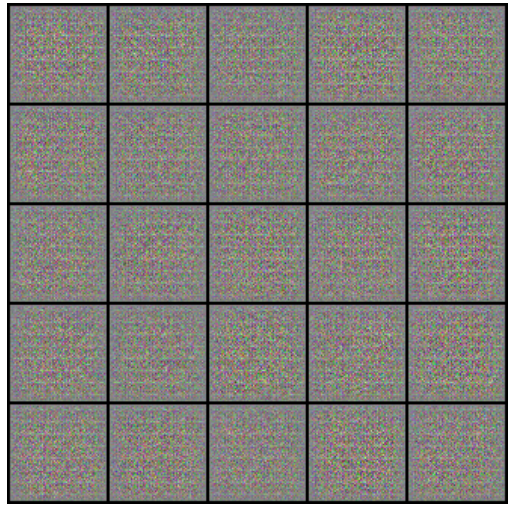
- At initialization, pretty reasonable:

$\mathbb{P}$: 

$\mathbb{Q}_\theta$: 

How likely is disjoint support?

- At initialization, pretty reasonable:

$\mathbb{P}$:  $\mathbb{Q}_\theta$: 

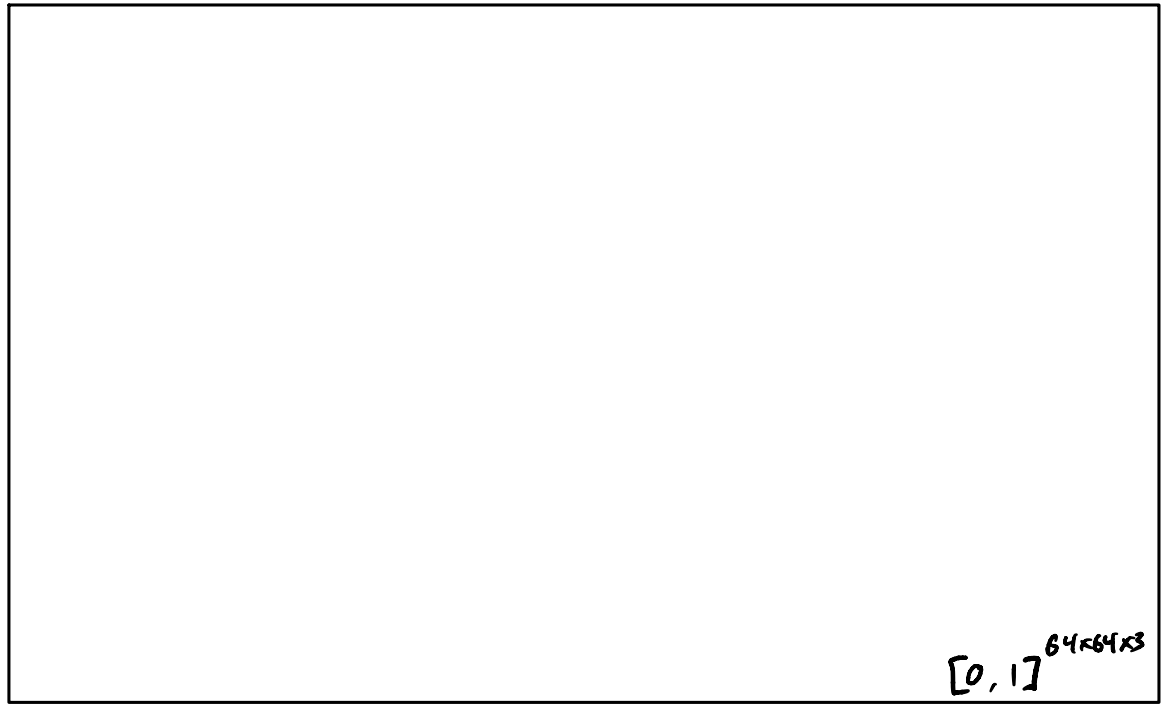
- Note that intersecting *almost nowhere* (with probability 0) doesn't change matters

Manifold structure

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

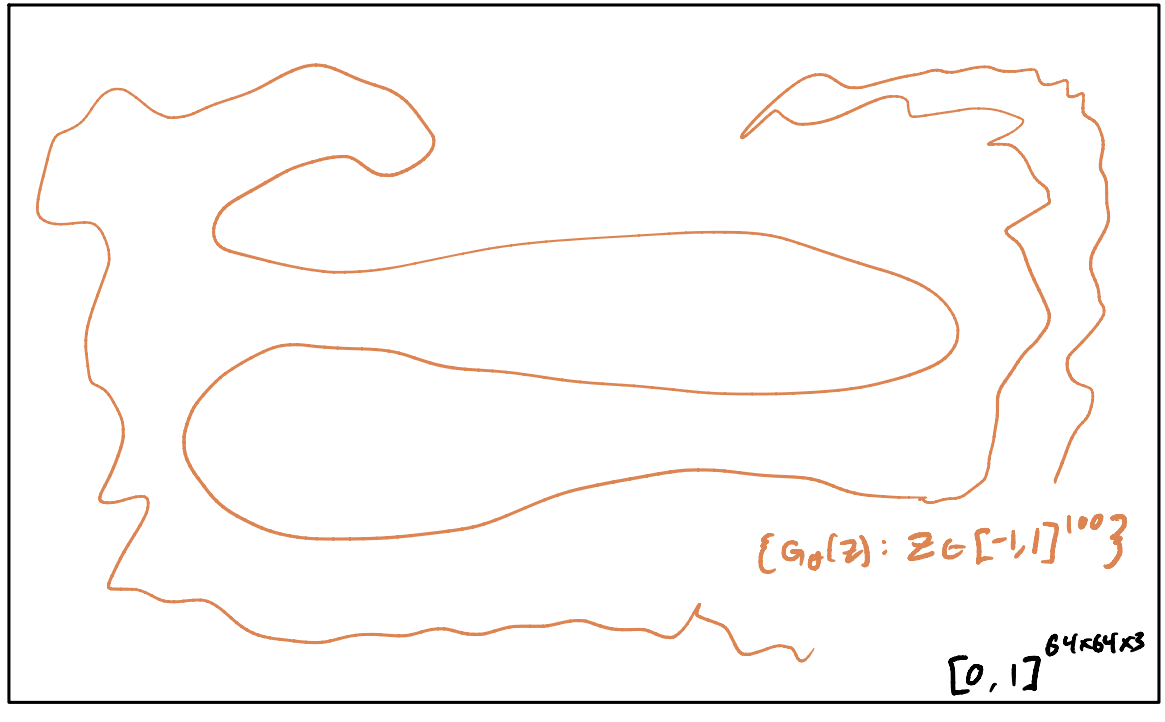
Manifold structure

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



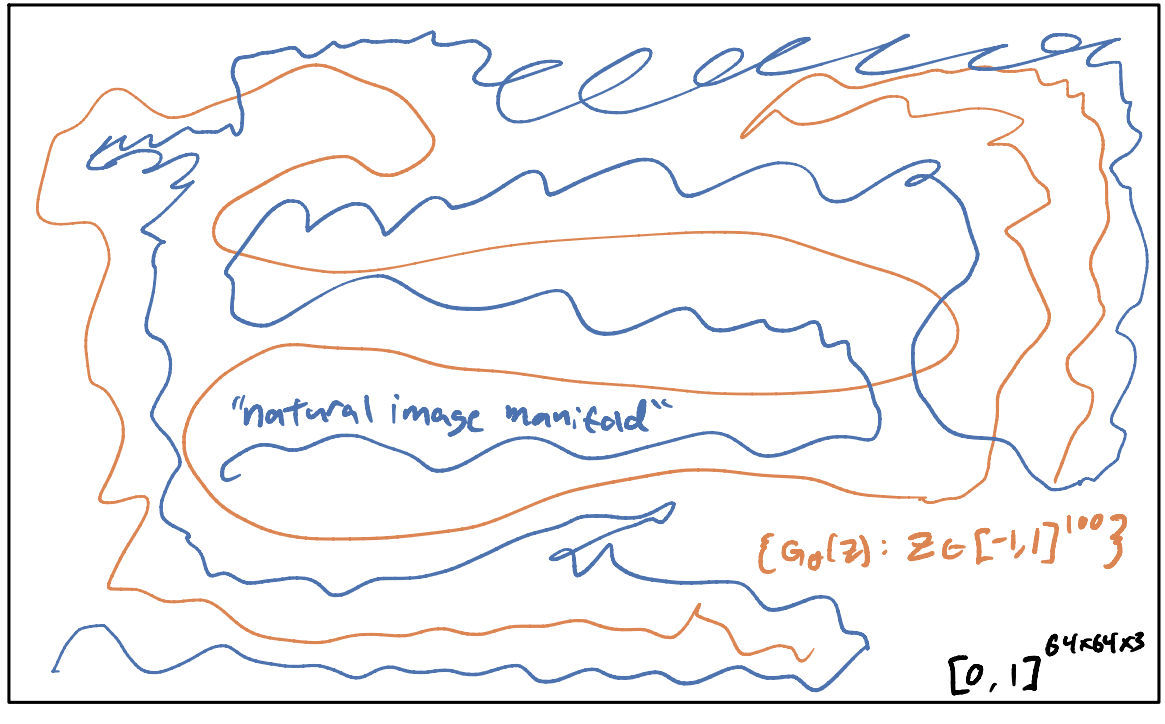
Manifold structure

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



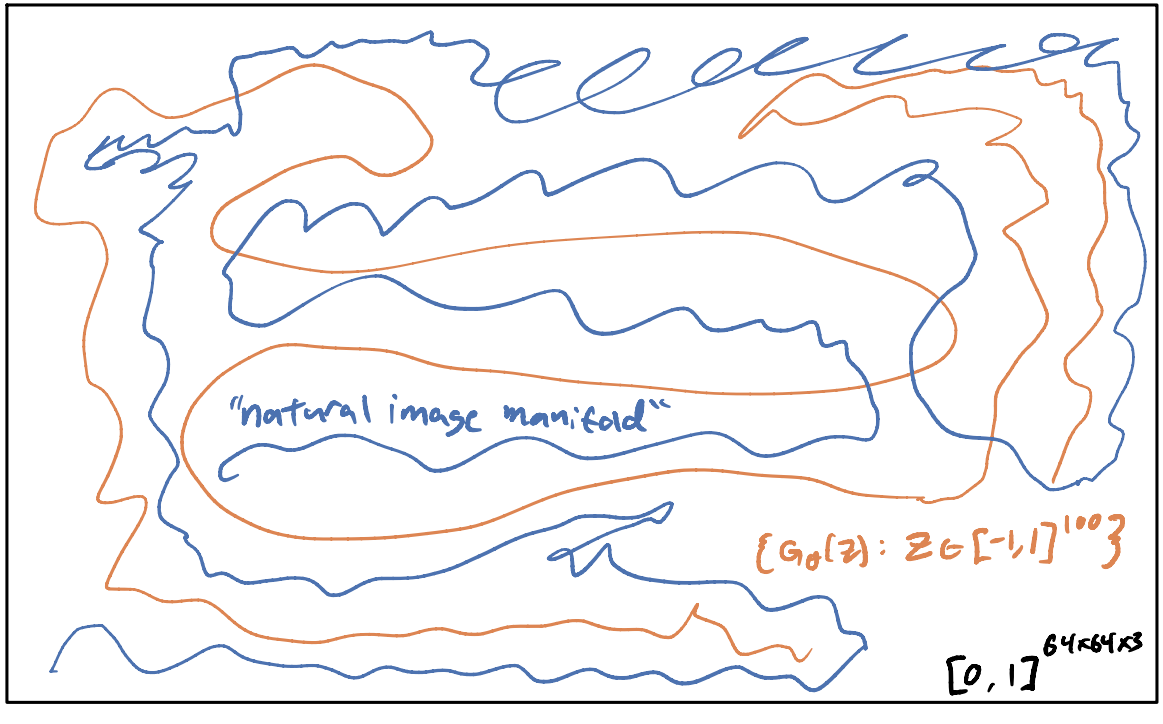
Manifold structure

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



Manifold structure

- If $G_\theta : \mathbb{R}^{100} \to \mathbb{R}^{64\times64\times3}$, for usual G_θ, $\mathbb{Q}_\theta$ is supported on a countable union of manifolds with dim ≤ 100
- "Natural image manifold" for $\mathbb{P}$ 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

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

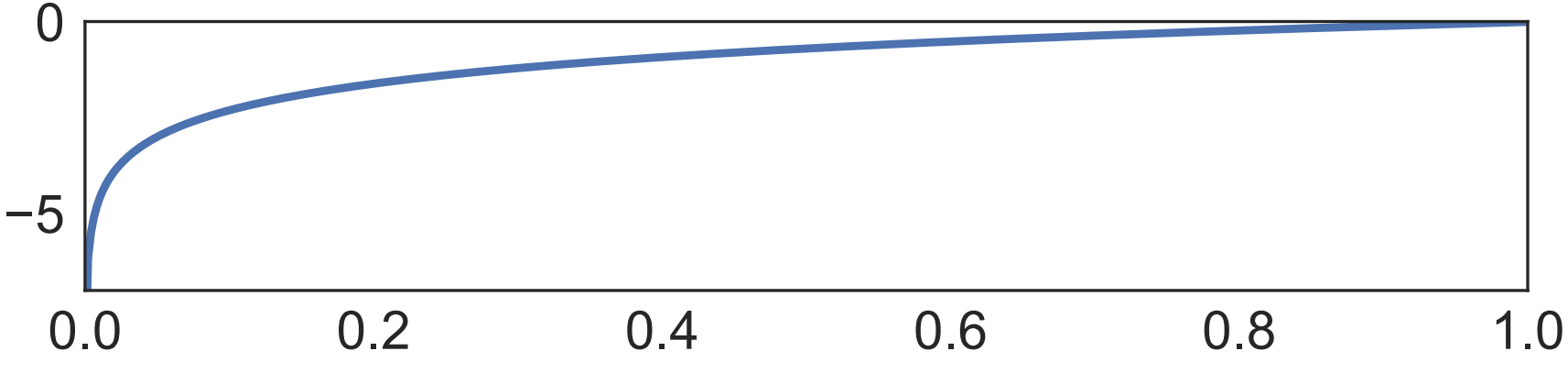
- $\max_{\theta} \log D_{\psi}(G_{\theta}(Z))$, not $\min_{\theta} \log(1 - D_{\psi}(G_{\theta}(Z)))$

A heuristic partial workaround

- Original GANs almost never use the minimax game

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

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

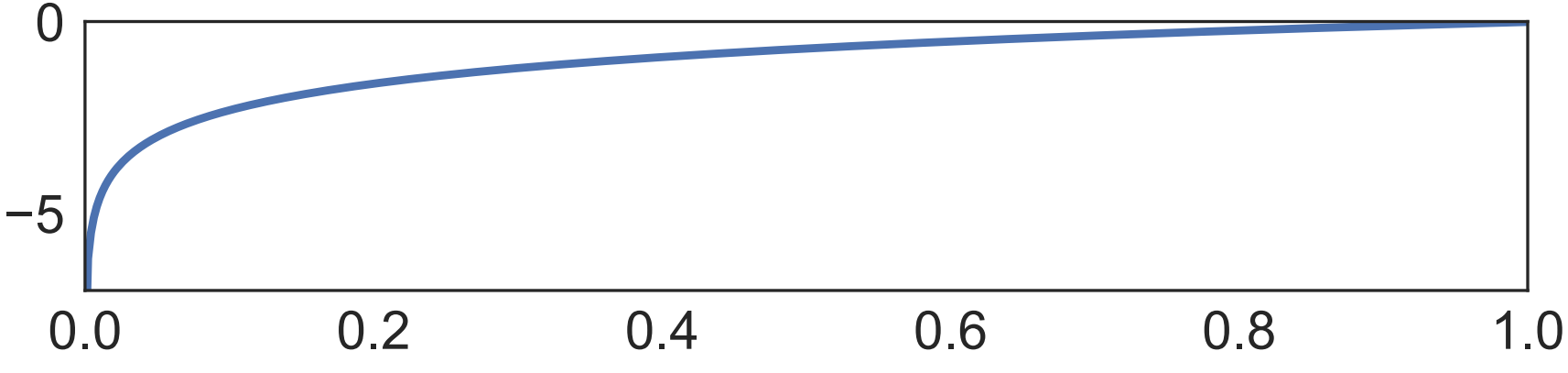


A heuristic partial workaround

- 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 $\log 1$



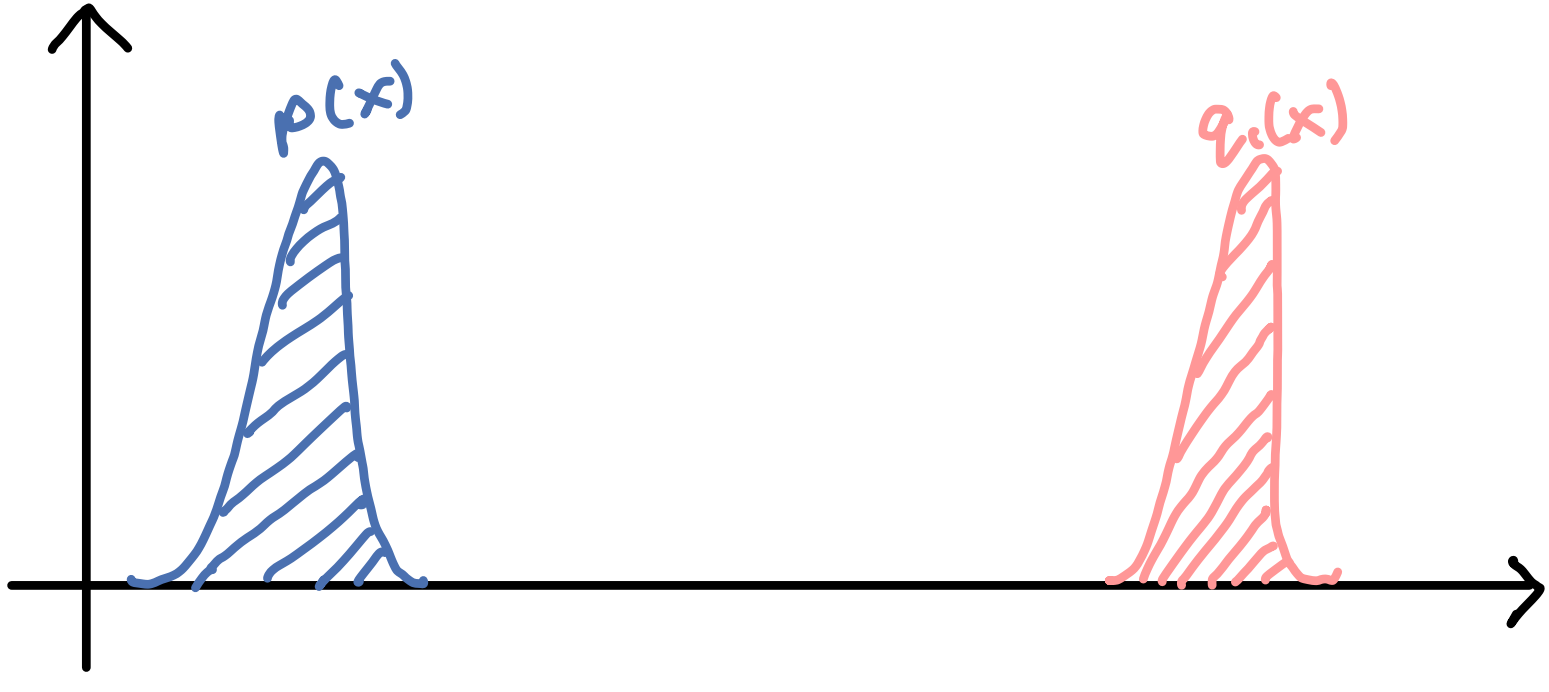
- When D_{ψ} is near-perfect, makes it unstable instead of stuck

Real problem: KL/JS don't know geometry!

- $\int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$ treats *each* x *totally separately*

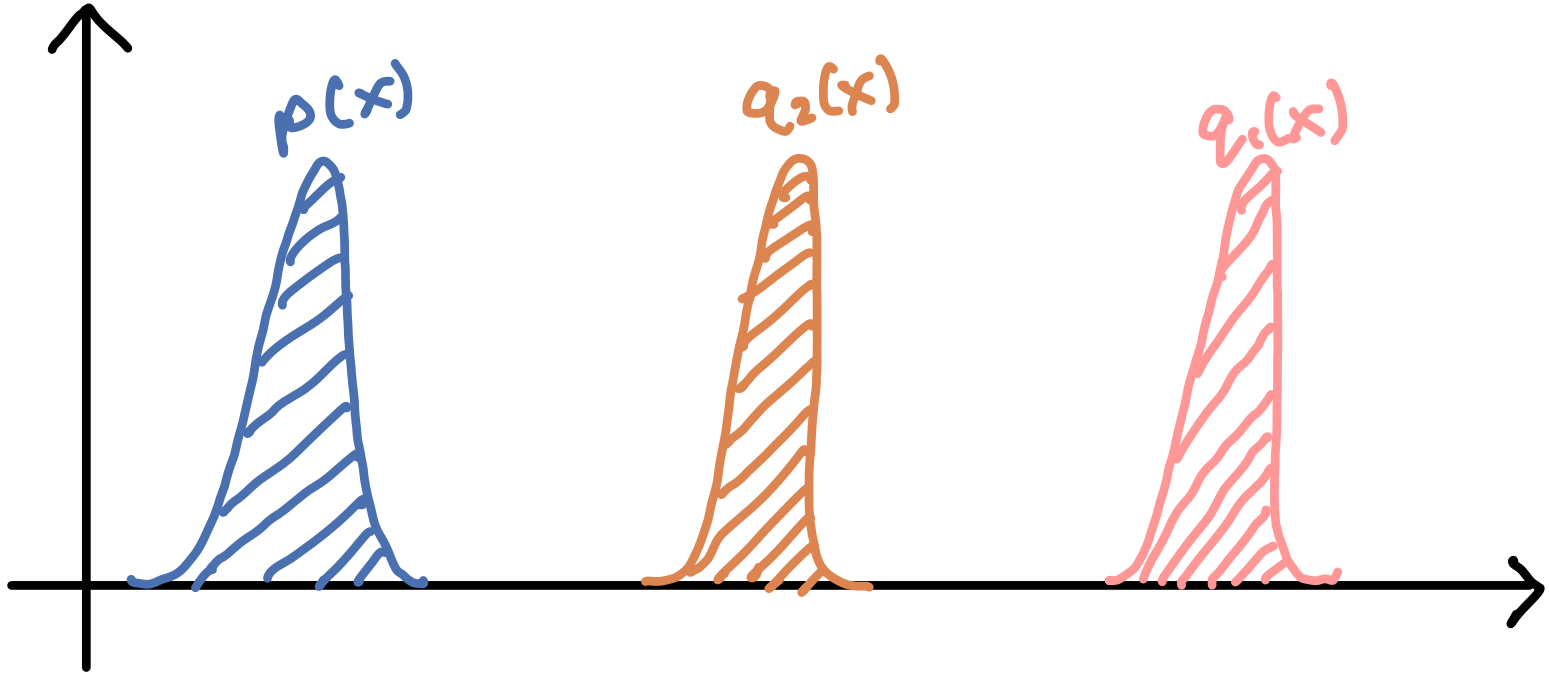
Real problem: KL/JS don't know geometry!

- $\int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$ treats *each x totally separately*



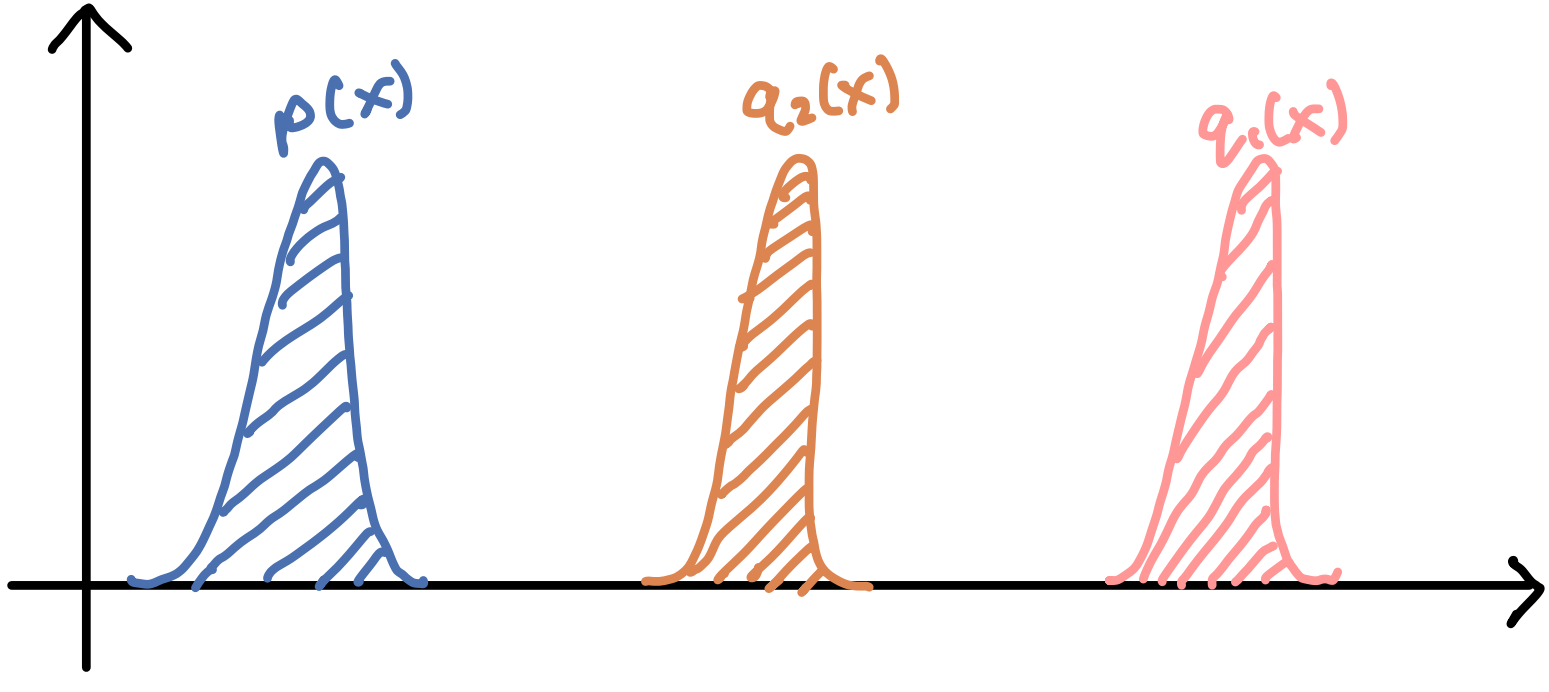
Real problem: KL/JS don't know geometry!

- $\int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$ treats *each x totally separately*



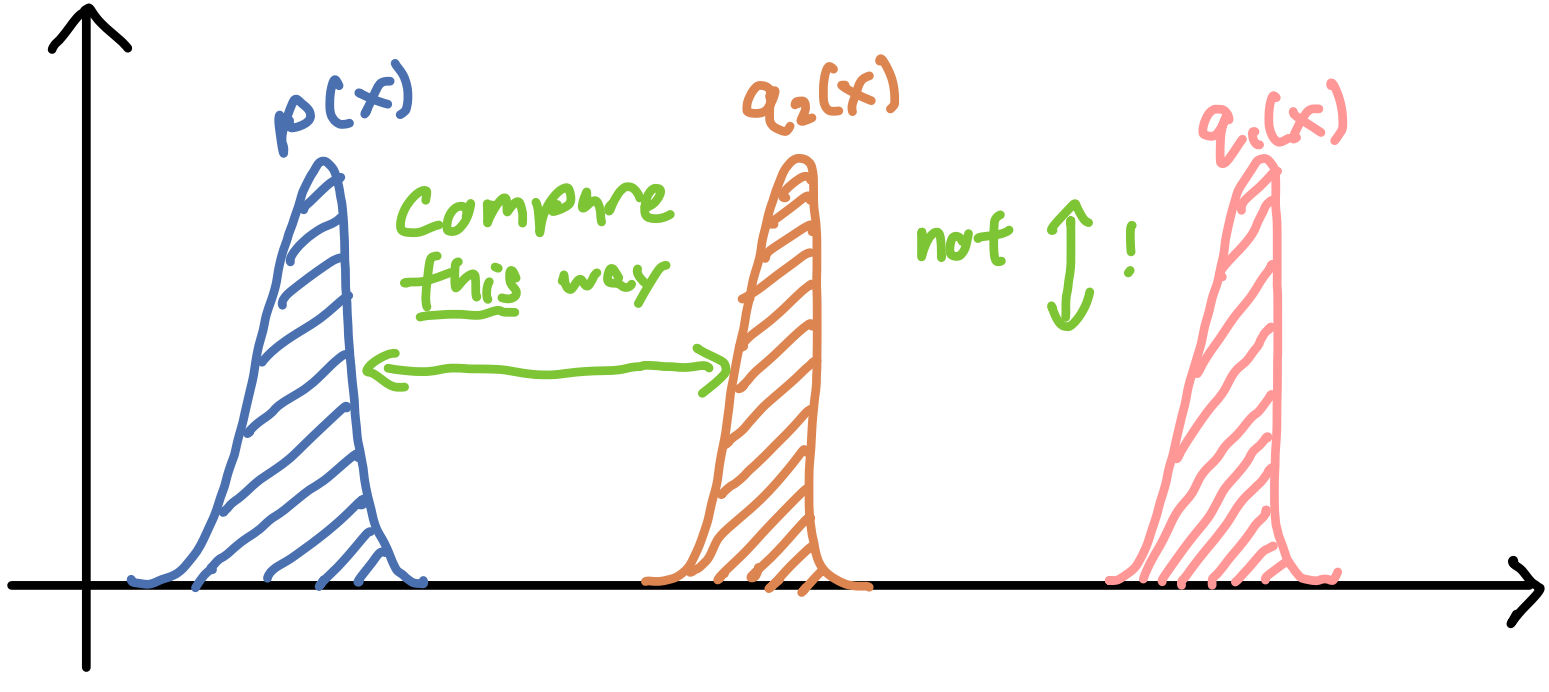
Real problem: KL/JS don't know geometry!

- $\int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$ treats *each x totally separately*
  - Can have $\mathbb{Q}_n \to \mathbb{Q}$ but $\mathrm{JS}(\mathbb{Q}_n, \mathbb{Q}) = \log 2 \not\to 0$



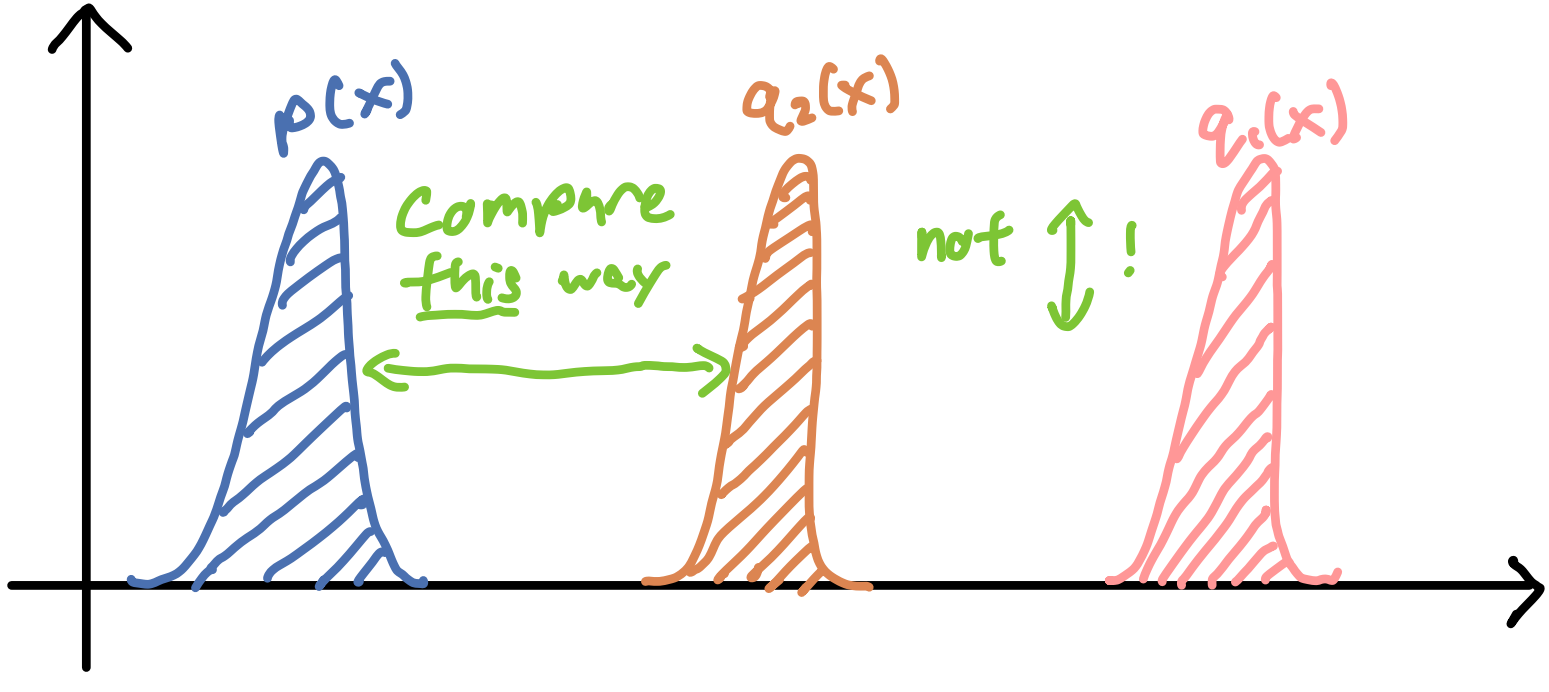
Real problem: KL/JS don't know geometry!

- $\int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$ treats *each x totally separately*
  - Can have $\mathbb{Q}_n \to \mathbb{Q}$ but $\mathrm{JS}(\mathbb{Q}_n, \mathbb{Q}) = \log 2 \not\to 0$
- Need some notion of geometry



Real problem: KL/JS don't know geometry!

- $\int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$ treats *each x totally separately*
  - Can have $\mathbb{Q}_n \to \mathbb{Q}$ but $\mathrm{JS}(\mathbb{Q}_n, \mathbb{Q}) = \log 2 \not\to 0$
- Need some notion of geometry
  - Should be *continuous in the weak topology*



Better Solution 1: Optimal Transport

Wasserstein, aka earth mover's distance:
How far do I have to move probability mass to turn $\mathbb{P}$ into $\mathbb{Q}$?

(from David Alvarez-Melis / Nicolo Fusi)

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

(C is a joint distribution on (X, Y) with marginals $\mathbb{P}$ and $\mathbb{Q}$)

Wasserstein via Kantorovich-Rubinstein duality

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

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

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

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

Wasserstein via Kantorovich-Rubinstein duality

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

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

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

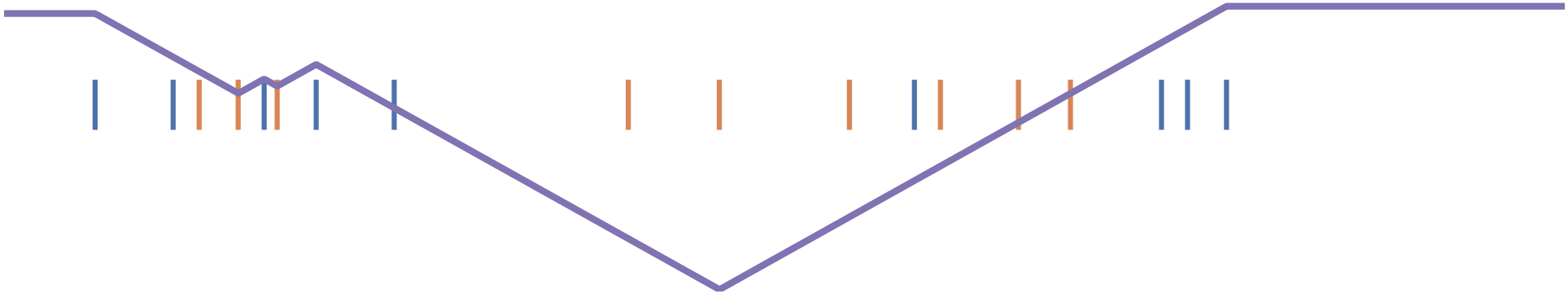
$\mathcal{W}$ is continuous (bounded domain): if $\mathbb{Q}_\theta \to \mathbb{P}$, then $\mathcal{W}(\mathbb{Q}_\theta, \mathbb{P}) \to 0$

Wasserstein via Kantorovich-Rubinstein duality

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

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

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



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

WGAN [Arjovsky/Chintala/Bottou ICML-17]

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

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Idea: replace discriminator D_ψ with a critic f_ψ
- Need to enforce $\|f_\psi\|_{\mathbf{Lip}} \leq 1$ ...or, really, any constant

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Idea: replace discriminator D_ψ with a critic f_ψ
- Need to enforce $\|f_\psi\|_{\text{Lip}} \leq 1$ ...or, really, any constant
- $$f_\psi(x) = h_L(W_L\, h_{L-1}(\cdots h_1(W_1 x)))$$

  so for usual deep nets,

$$\|f_\psi\|_{\text{Lip}} \leq \|h_L\|_{\text{Lip}} \|W_L\|_{\text{op}} \cdots \|h_1\|_{\text{Lip}} \|W_1\|_{\text{op}}$$

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Idea: replace discriminator D_ψ with a critic f_ψ
- Need to enforce $\|f_\psi\|_{\text{Lip}} \leq 1$ ...or, really, any constant
- $$f_\psi(x) = h_L(W_L\, h_{L-1}(\cdots h_1(W_1 x)))$$

  so for usual deep nets,

$$\|f_\psi\|_{\text{Lip}} \leq \|h_L\|_{\text{Lip}} \|W_L\|_{\text{op}} \cdots \|h_1\|_{\text{Lip}} \|W_1\|_{\text{op}}$$

- $\|W\|_{\text{op}} := \sup_{x \neq 0} \frac{\|Wx\|_2}{\|x\|} = \sigma_{\max}(W)$ is *spectral* norm

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Idea: replace discriminator D_ψ with a critic f_ψ
- Need to enforce $\|f_\psi\|_{\text{Lip}} \leq 1$ ...or, really, any constant
- $$f_\psi(x) = h_L(W_L\, h_{L-1}(\cdots h_1(W_1 x)))$$

  so for usual deep nets,

$$\|f_\psi\|_{\text{Lip}} \leq \|h_L\|_{\text{Lip}} \|W_L\|_{\text{op}} \cdots \|h_1\|_{\text{Lip}} \|W_1\|_{\text{op}}$$

- $\|W\|_{\text{op}} := \sup_{x \neq 0} \frac{\|Wx\|_2}{\|x\|} = \sigma_{\max}(W)$ is *spectral* norm
- h is fixed; often $\|h\|_{\text{Lip}} \leq 1$

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Need

$$\|f_\psi\|_{\text{Lip}} \leq \left(\prod_{j=1}^{L} \|h_L\|_{\text{Lip}}\right) \|W_L\|_{\text{op}} \cdots \|W_1\|_{\text{op}} \leq \text{const}$$

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Need

$$\|f_\psi\|_{\text{Lip}} \leq \left(\prod_{j=1}^{L} \|h_L\|_{\text{Lip}}\right) \|W_L\|_{\text{op}} \cdots \|W_1\|_{\text{op}} \leq \text{const}$$

- WGANs: Bound $\|W_i\|_\infty \leq C$
  - This implies $\|W_i\|_{\text{op}} \leq \sqrt{k_i k_{i-1}} C$

WGAN [Arjovsky/Chintala/Bottou ICML-17]

- Need

$$\|f_\psi\|_{\text{Lip}} \leq \left(\prod_{j=1}^{L} \|h_L\|_{\text{Lip}}\right) \|W_L\|_{\text{op}} \cdots \|W_1\|_{\text{op}} \leq \text{const}$$

- WGANs: Bound $\|W_i\|_\infty \leq C$
  - This implies $\|W_i\|_{\text{op}} \leq \sqrt{k_i k_{i-1}} C$
- ...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*

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

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

WGAN-GP [Gulrajani+ NeurIPS-17]

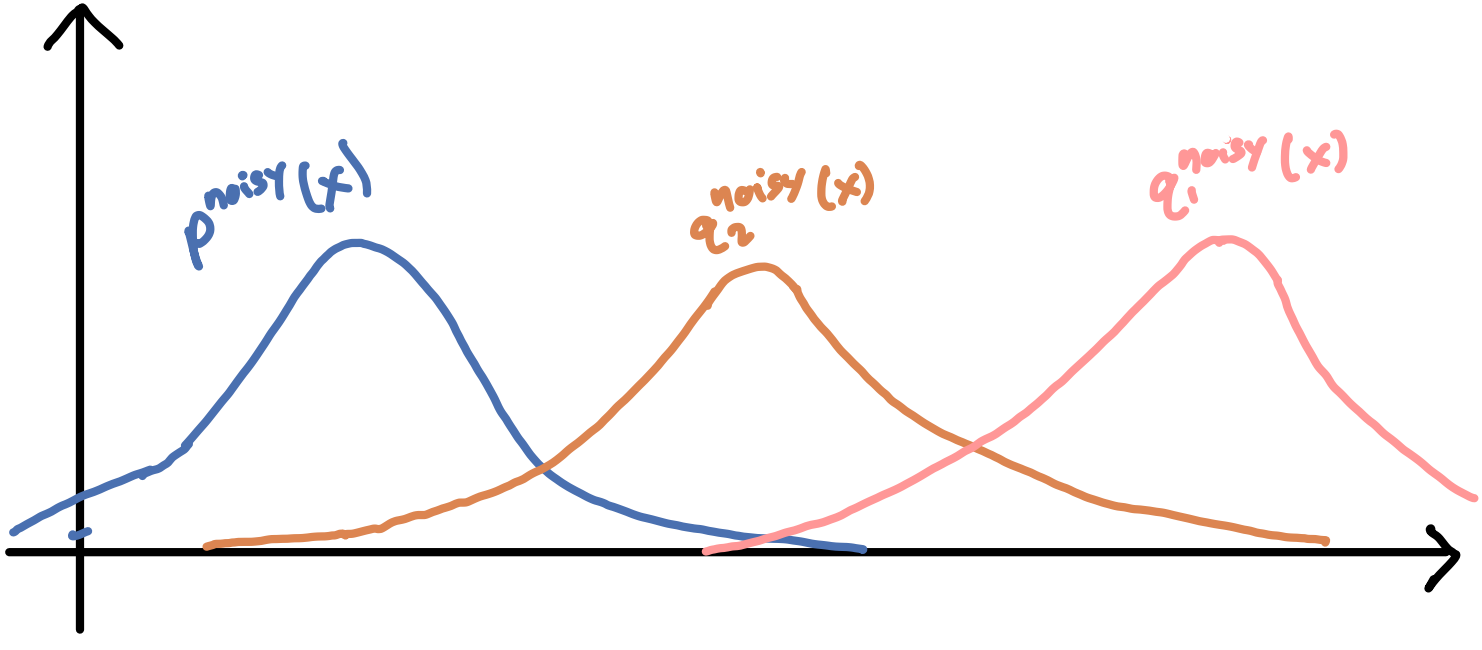
- Controlling $\|\nabla f(X)\|$ *everywhere* is hard
- Instead, control $\|\nabla f(\tilde{X})\|$ *on average, near the data*

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

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

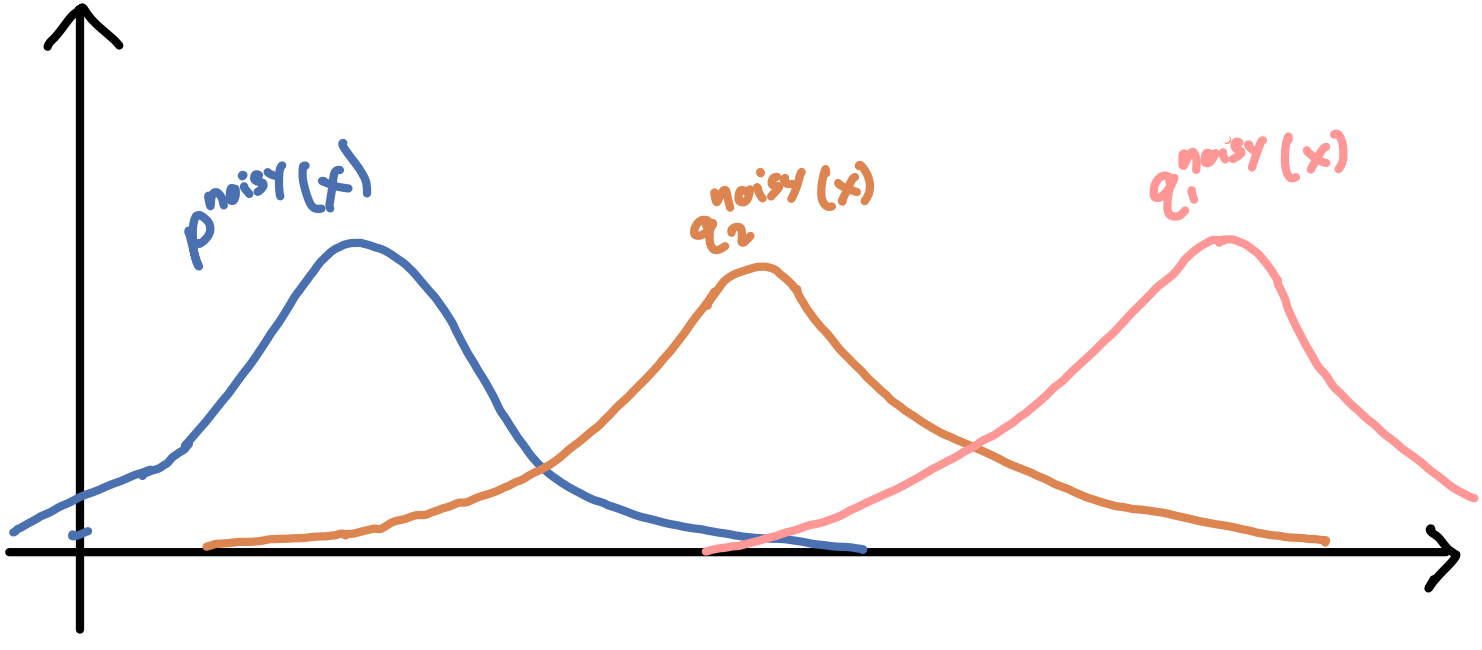
Solution 2: add noise

- Can keep JS if we “blur” the data
- Use $X + \varepsilon$, $Y + \varepsilon'$ for some independent, full-dim noise ε



Solution 2: add noise

- Can keep JS if we “blur” the data
- 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 in closed form

- Use $X + \varepsilon$, $Y + \varepsilon'$ for some independent, full-dim noise ε
- 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 in closed form

- Use $X + \varepsilon$, $Y + \varepsilon'$ for some independent, full-dim noise ε
- If $\varepsilon \sim \mathrm{N}(0, \gamma I)$ and $\gamma \to 0$, get [Mescheder+ NeurIPS-17]

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

- Similar kind of gradient penalty!

Solution 2': add noise in closed form

- Use $X + \varepsilon$, $Y + \varepsilon'$ for some independent, full-dim noise ε
- If $\varepsilon \sim \mathrm{N}(0, \gamma I)$ and $\gamma \to 0$, get [Mescheder+ NeurIPS-17]

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

- Similar 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: $f_\ell = h_\ell(W_\ell f_{\ell-1}(x))$
- Spectral normalization: $f_\ell = h\left(\frac{1}{\|W_\ell\|_{\text{op}}} W_\ell f_{\ell-1}(x)\right)$
- Guarantees $\|f\|_{\text{Lip}} \leq 1$, like WGAN
- With tricks, faster to evaluate than gradient penalties
- Not as well understood yet

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

- Regular deep nets: $f_\ell = h_\ell(W_\ell f_{\ell-1}(x))$
- Spectral normalization: $f_\ell = h\left(\frac{1}{\|W_\ell\|_{\text{op}}} W_\ell f_{\ell-1}(x)\right)$
- Guarantees* $\|f\|_{\text{Lip}} \leq 1$, like WGAN
- With tricks, 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: $\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}_\theta}\|^2 + \mathbf{Tr}\left(\Sigma_{\mathbb{P}} + \Sigma_{\mathbb{Q}_\theta} - 2\left(\Sigma_{\mathbb{P}}\Sigma_{\mathbb{Q}_\theta}\right)^{\frac{1}{2}}\right)$

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

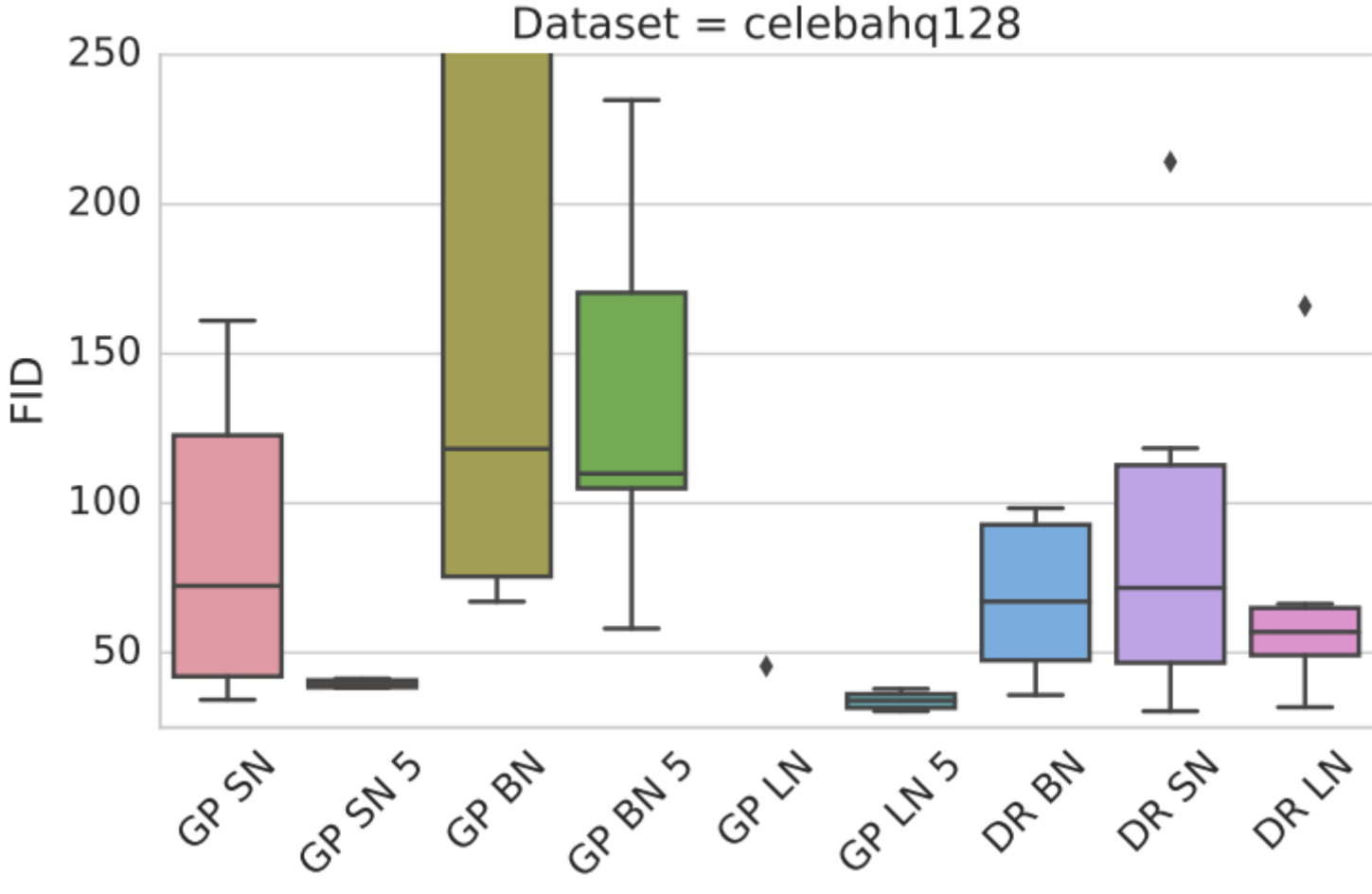
FID [Heusel+ NeurIPS-17] and KID [Bińkowski+ ICLR-18]

- 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

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

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

Maximum Mean Discrepancy

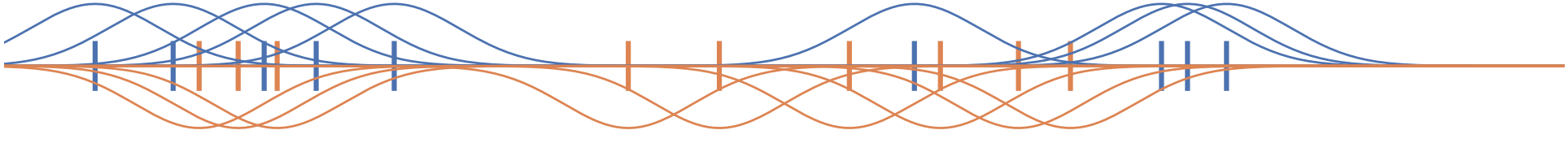
$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f:\|f\|_{\mathcal{H}_k} \leq 1} \mathop{\mathbb{E}}_{X \sim \mathbb{P}}[f(X)] - \mathop{\mathbb{E}}_{Y \sim \mathbb{Q}}[f(Y)]$$

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

Maximum Mean Discrepancy

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

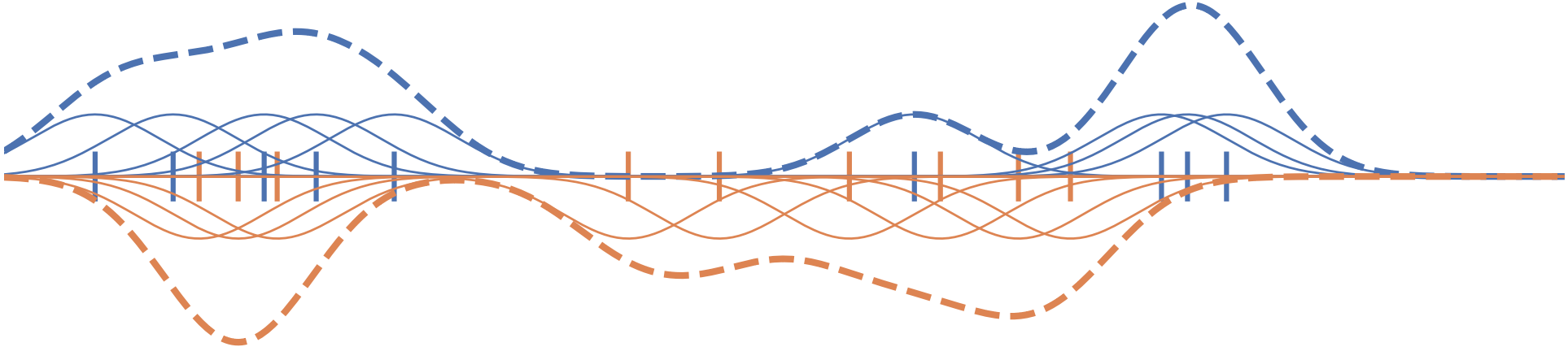
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

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

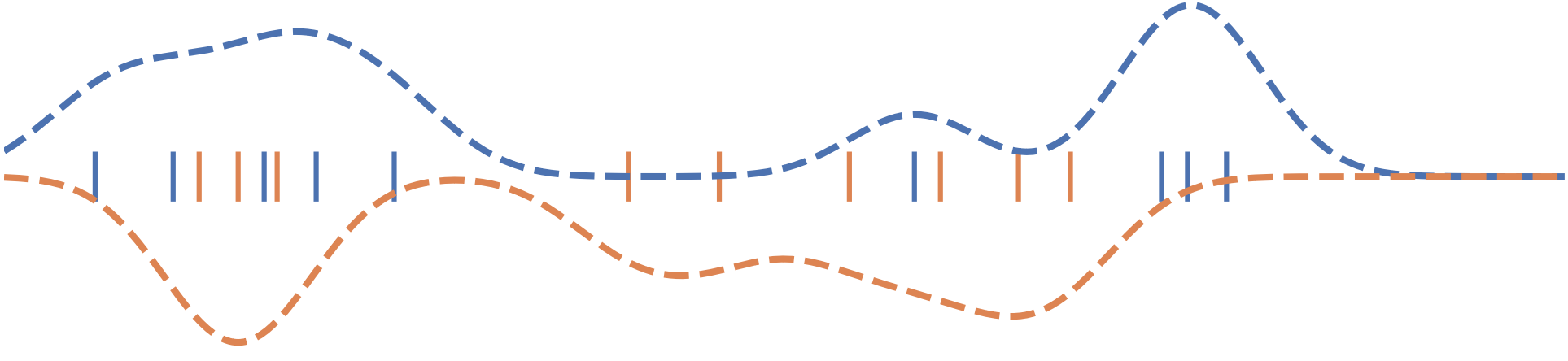
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

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

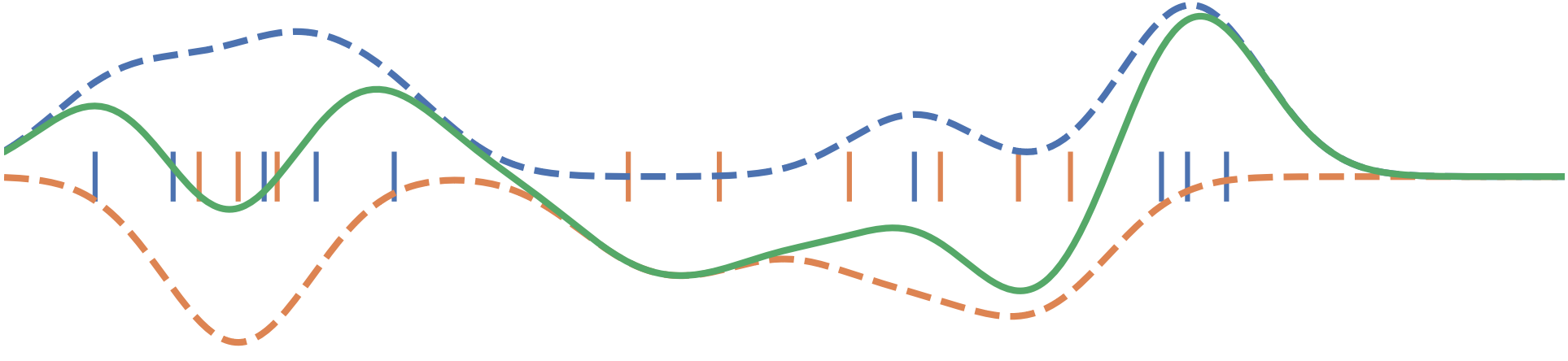
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

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

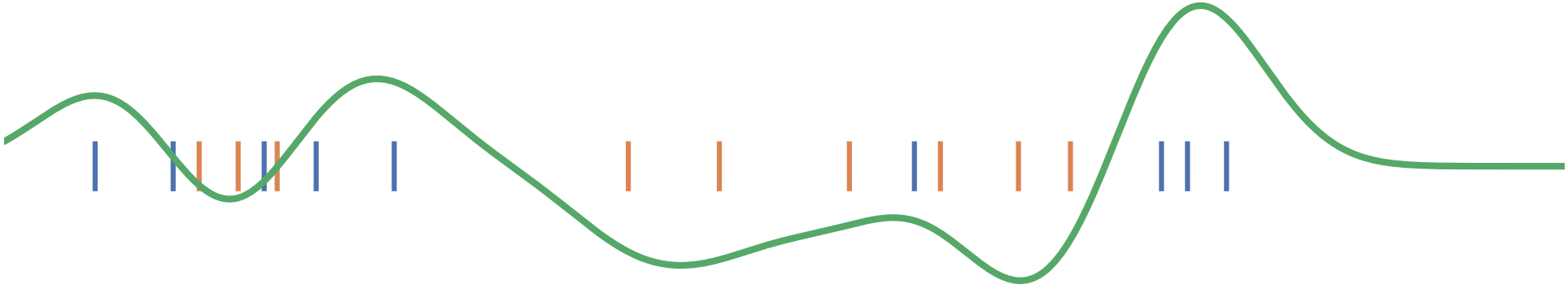
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

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

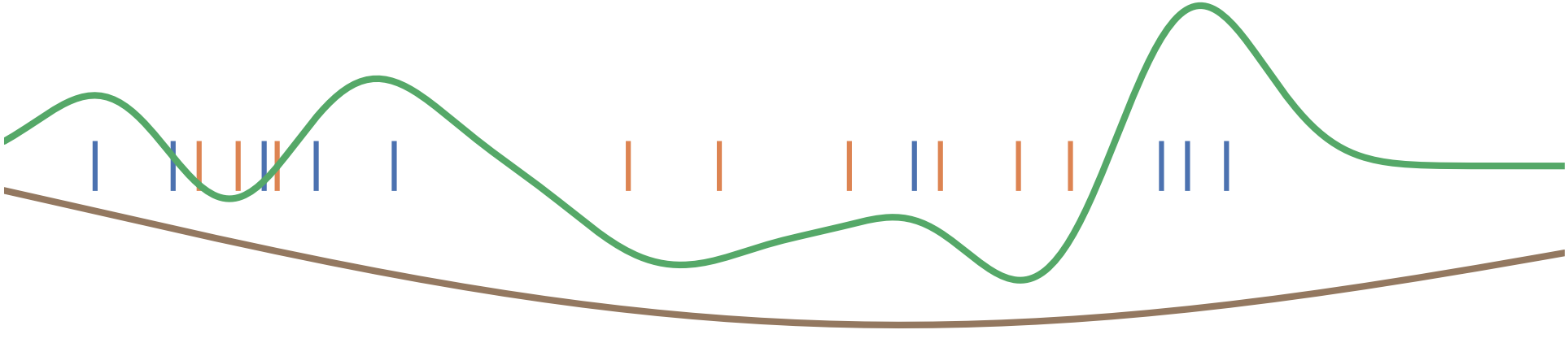
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Maximum Mean Discrepancy

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

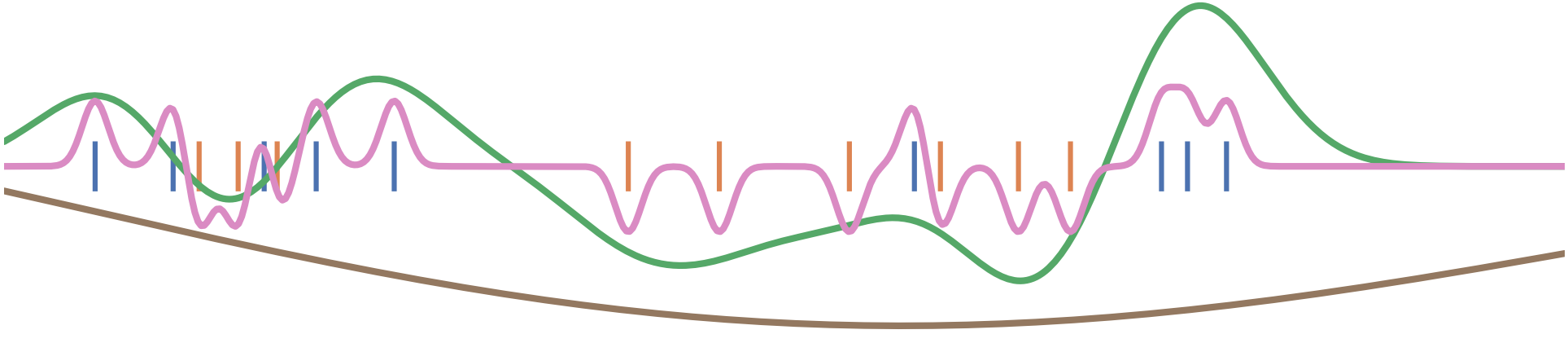
$\|f\|_{\mathcal{H}_k}$ is smoothness induced by kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



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 $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

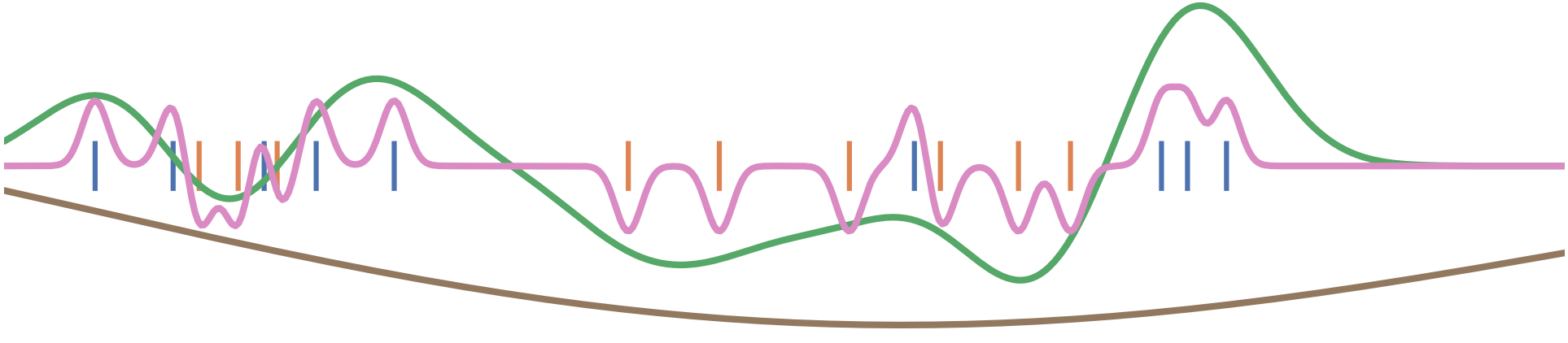


Maximum Mean Discrepancy

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

$\|f\|_{\mathcal{H}_k}$ 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}

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}

K_{YY}

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}

K_{YY}

K_{XY}

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

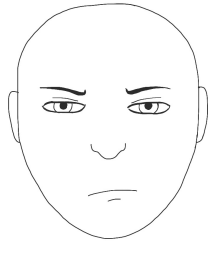
MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

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

Critic

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

How are these?



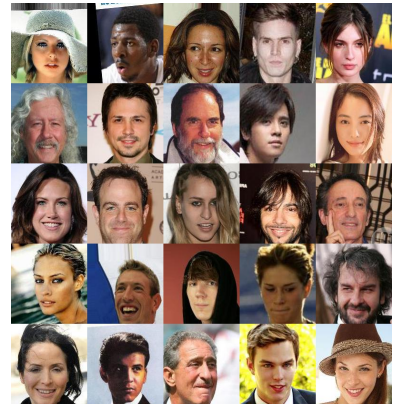
MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

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

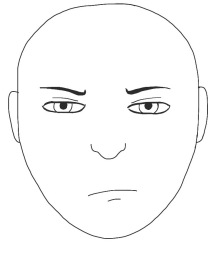
Target ($\mathbb{P}$)

Critic

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



How are these?



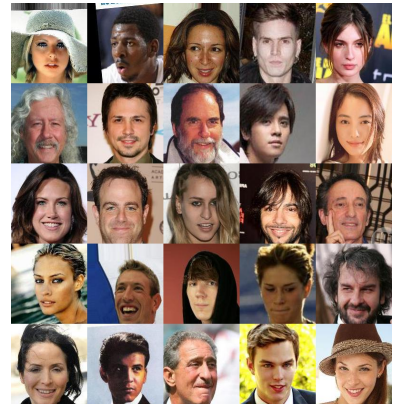
MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

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

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

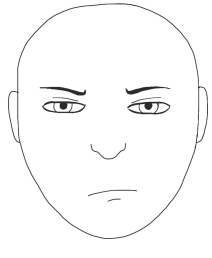
Critic

Target ($\mathbb{P}$)



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

How are these?



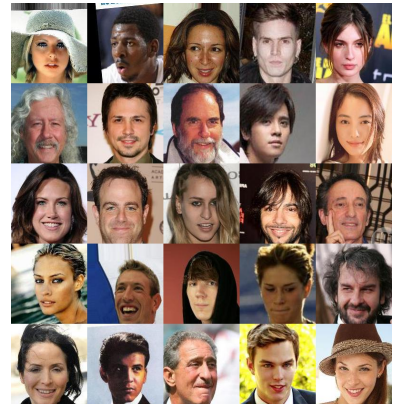
MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

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

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

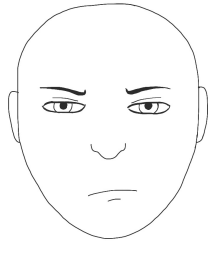
Critic

Target ($\mathbb{P}$)



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

How are these?



:(I'll try harder...

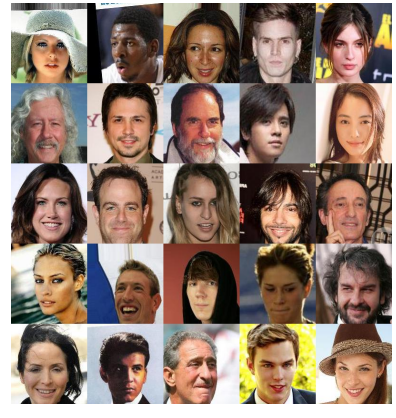
MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

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

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

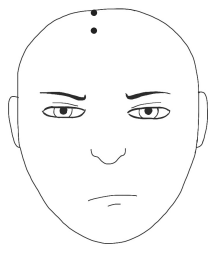
Critic

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

$\mathbb{P}$

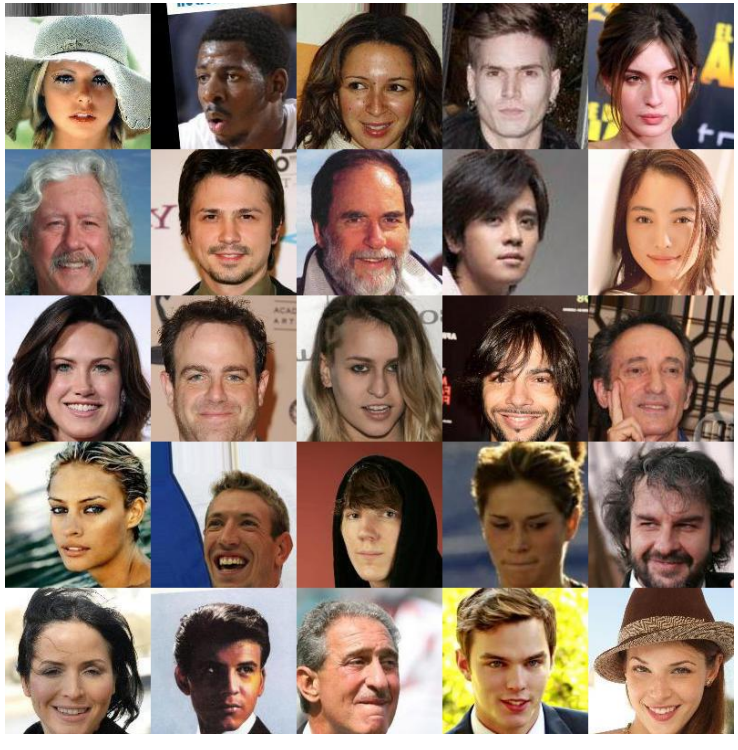
MMD models [Li+ ICML-15, Dziugaite+ UAI-15]

MNIST, mix of Gaussian kernels

$\mathbb{P}$

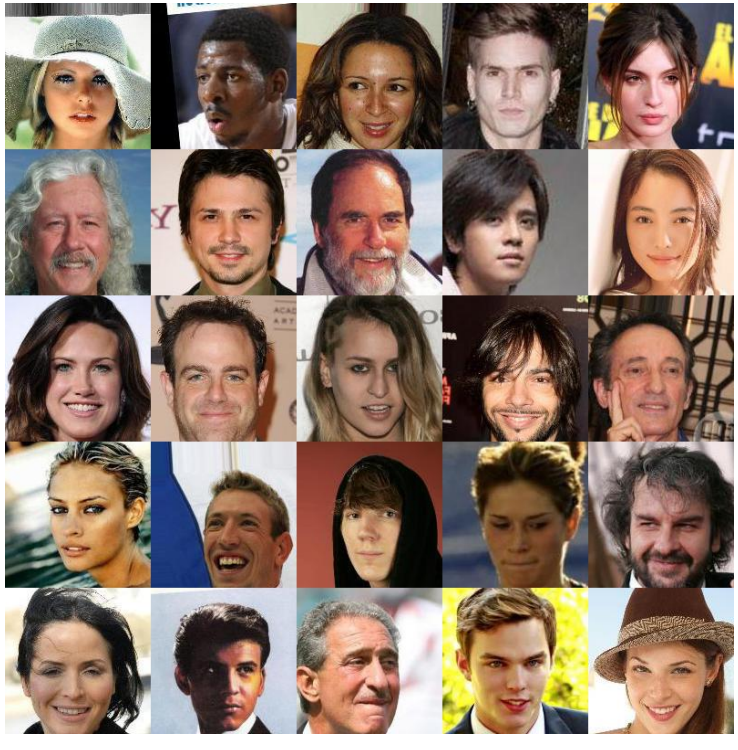
$\mathbb{Q}_\theta$

Celeb-A, mix of rational quadratic + linear kernels

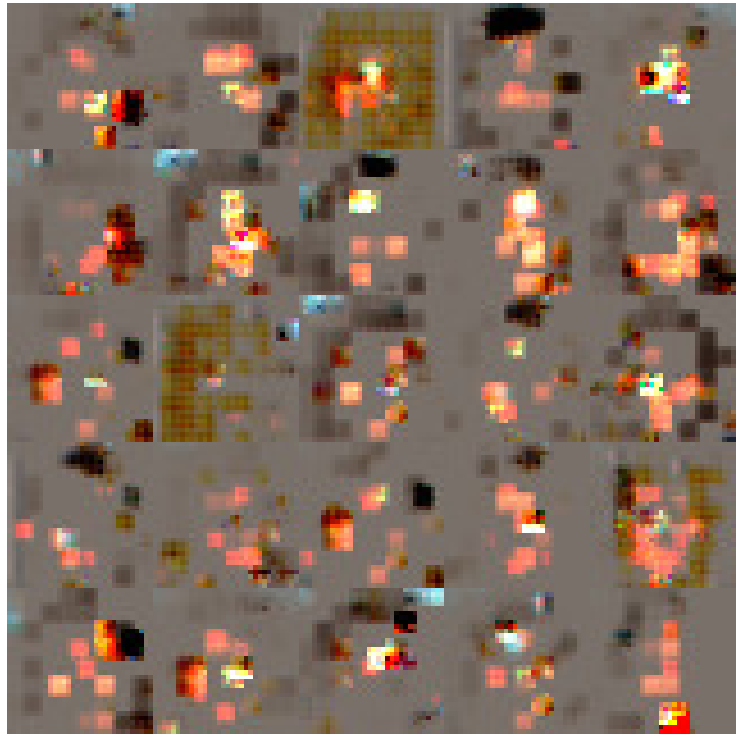


$\mathbb{P}$

Celeb-A, mix of rational quadratic + linear kernels



$\mathbb{P}$



$\mathbb{Q}_\theta$

MMD loss with a smarter kernel

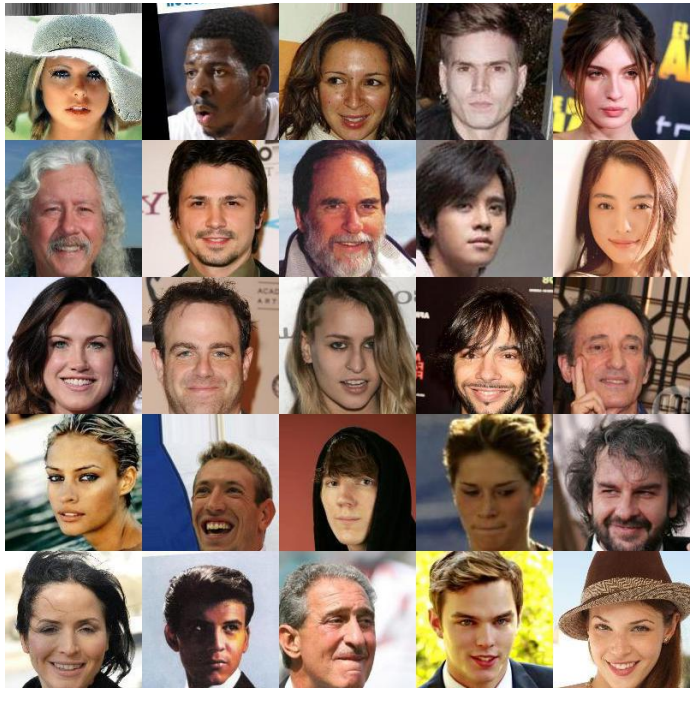
$$k(x, y) = k_{\text{top}}(\phi(x), \phi(y))$$

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

MMD loss with a smarter kernel

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

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

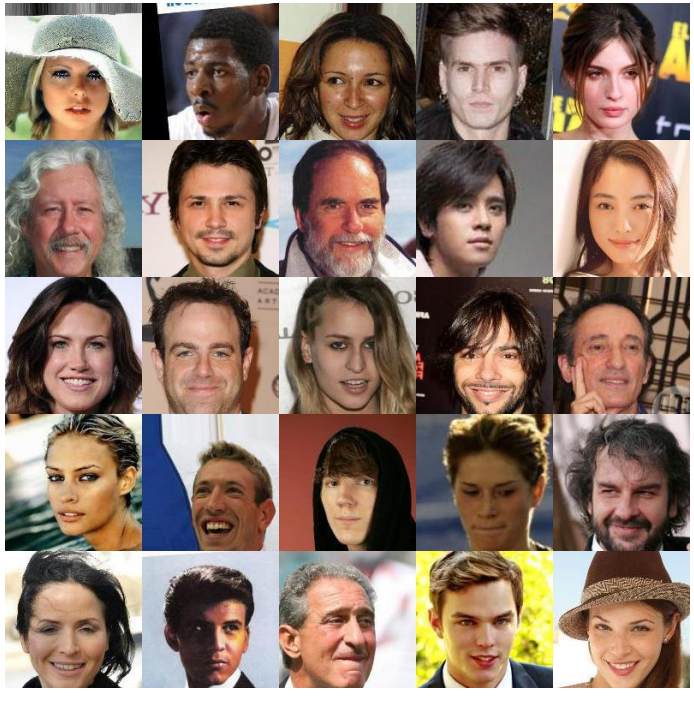


$\mathbb{P}$

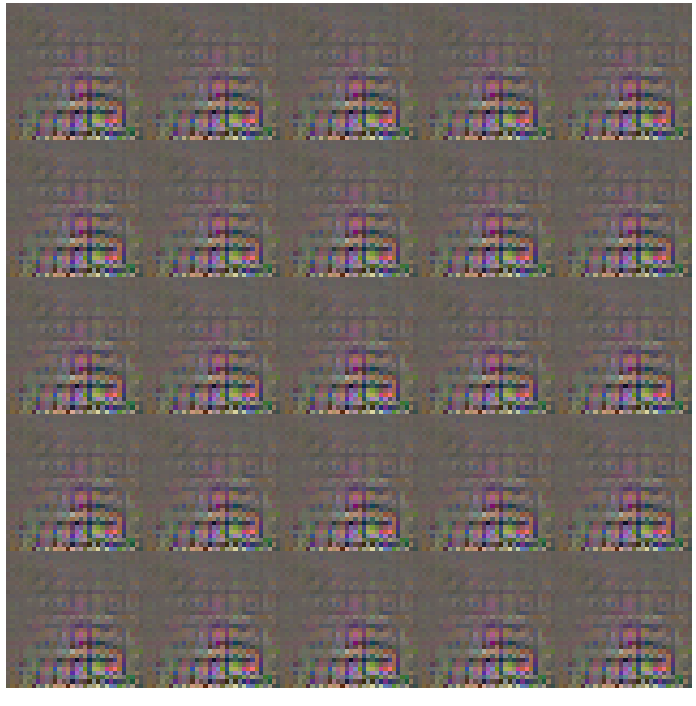
MMD loss with a smarter kernel

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

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



$\mathbb{P}$



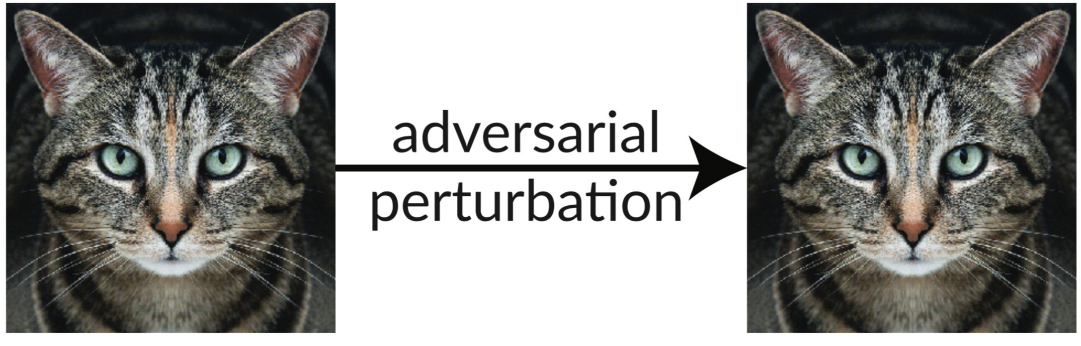
$\mathbb{Q}_\theta$

MMD loss with a smarter kernel

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

- $\phi : \mathcal{X} \to \mathbb{R}^d$
- k_{top} is a polynomial

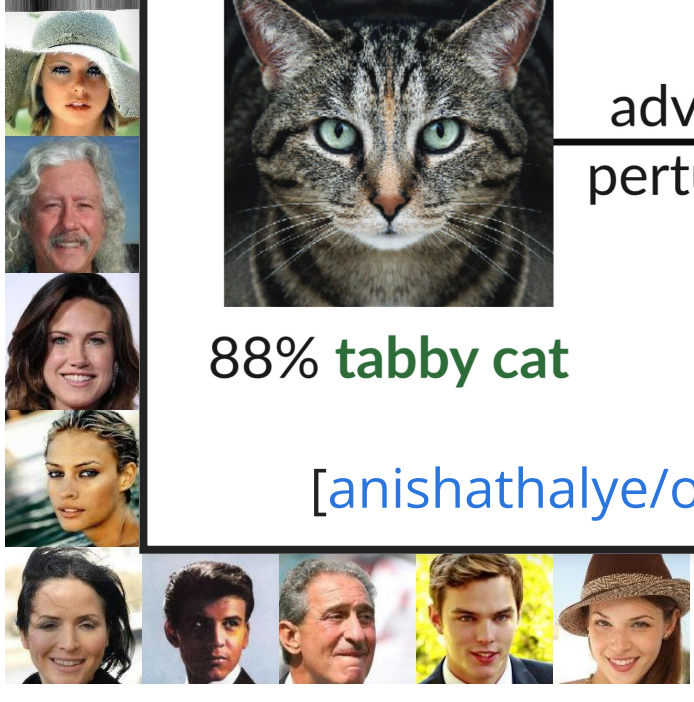
We just got adversarial examples!



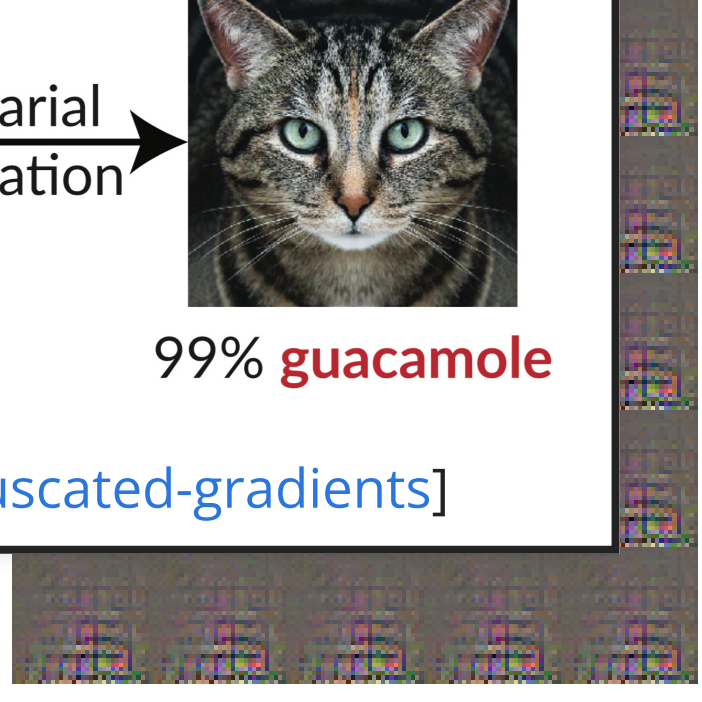
88% tabby cat

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:

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

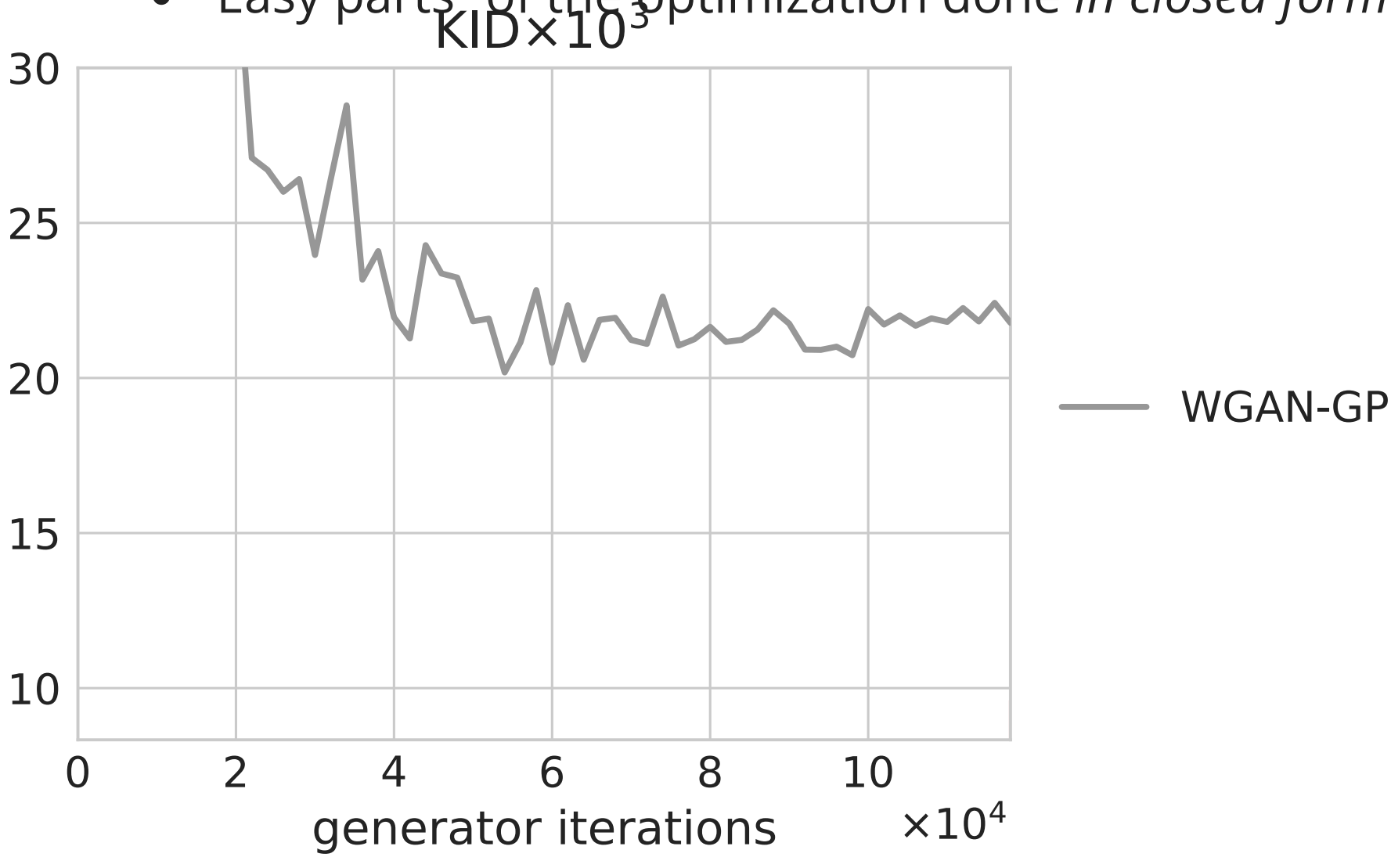
- New distance based on *all* these kernels:

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

- 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

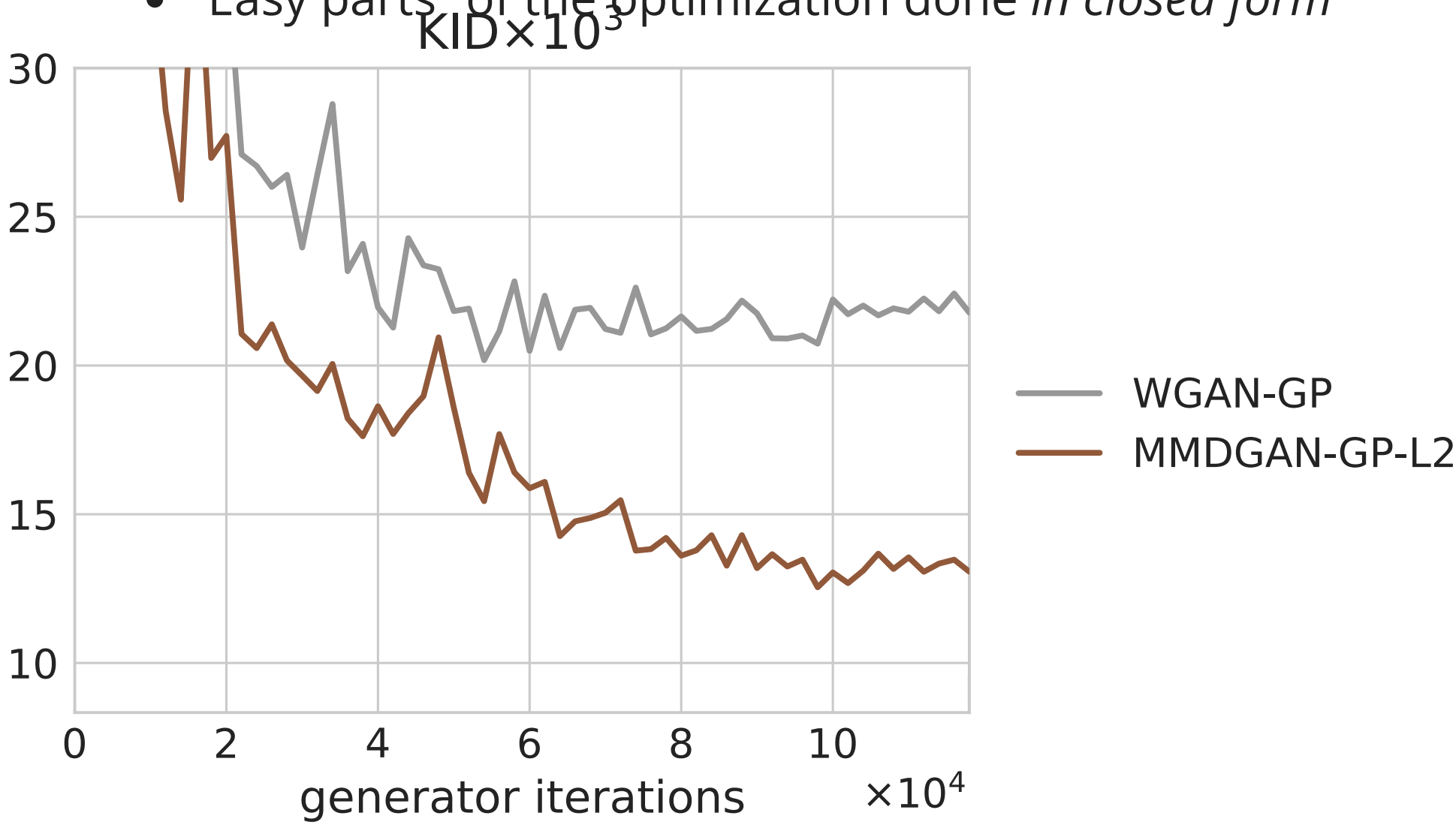
Why MMD GANs?

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



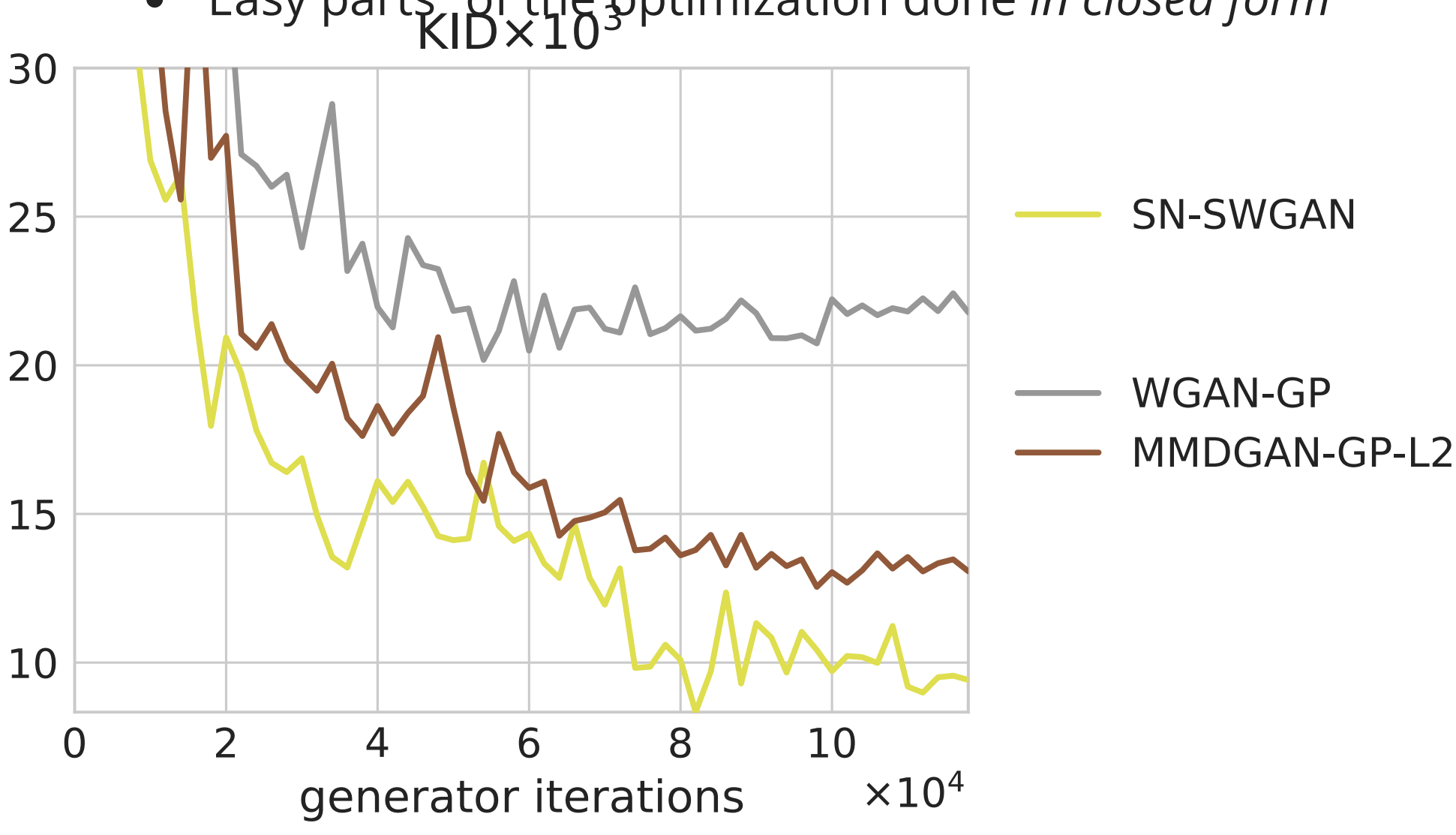
Why MMD GANs?

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



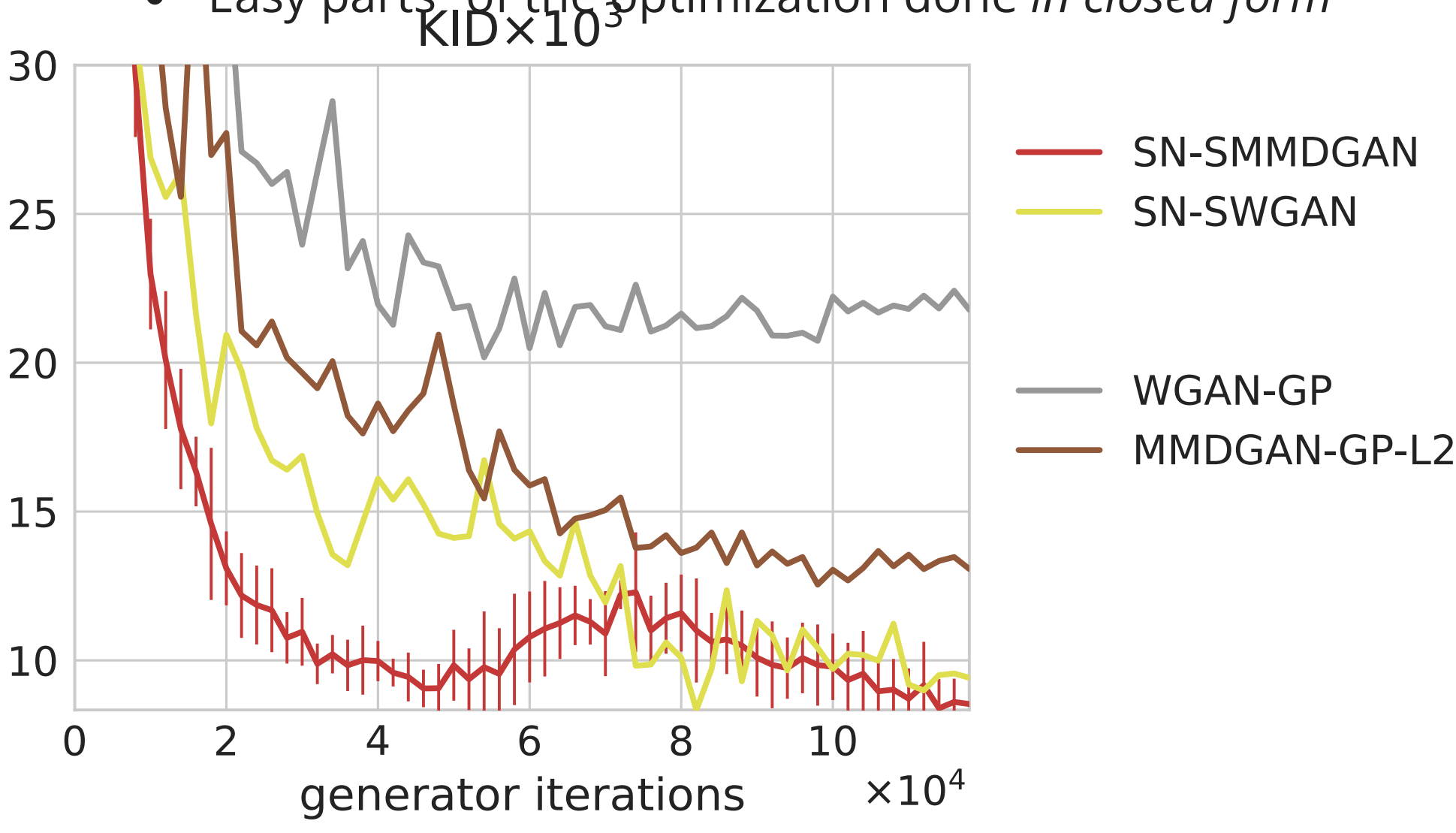
Why MMD GANs?

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



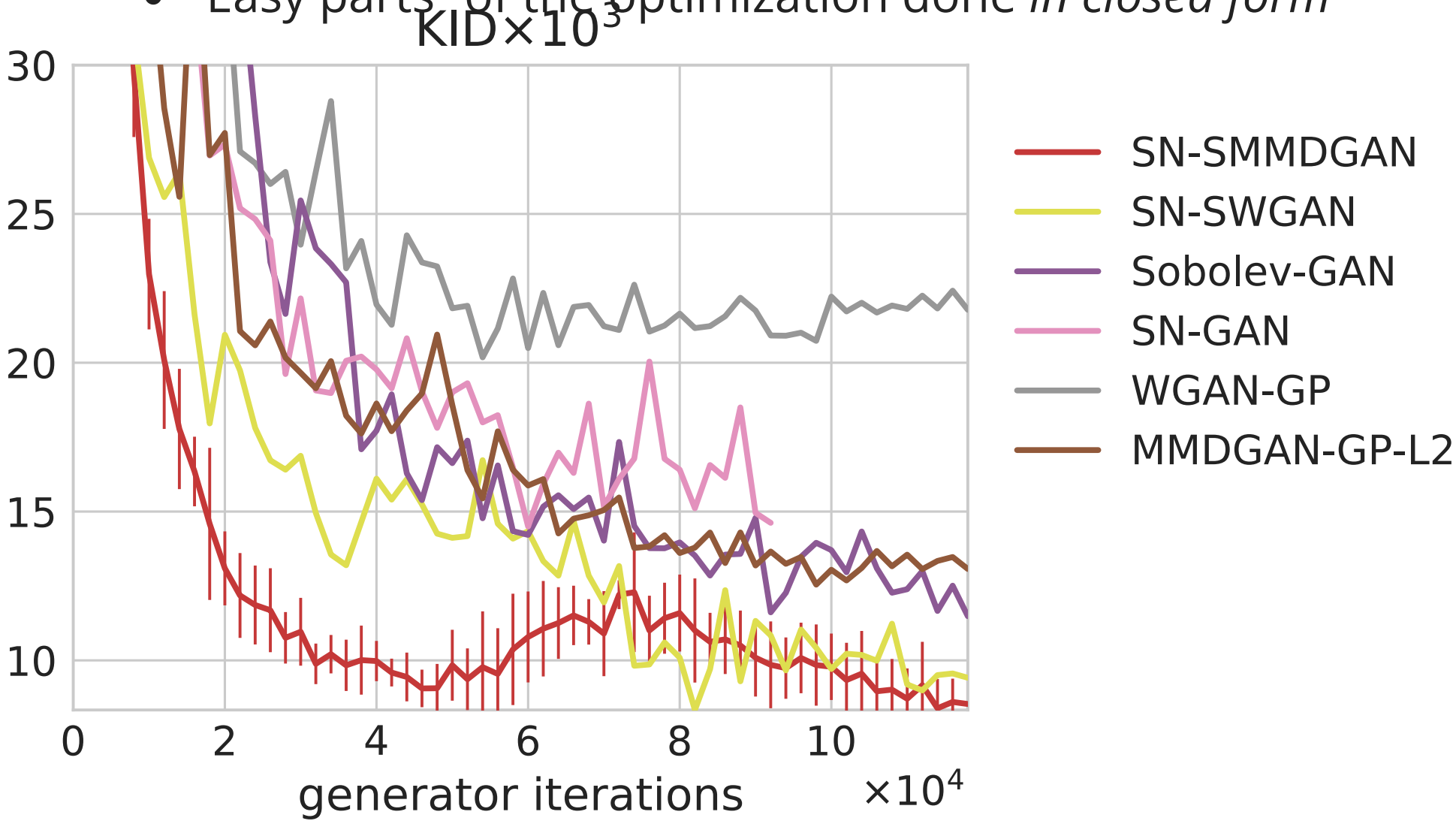
Why MMD GANs?

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

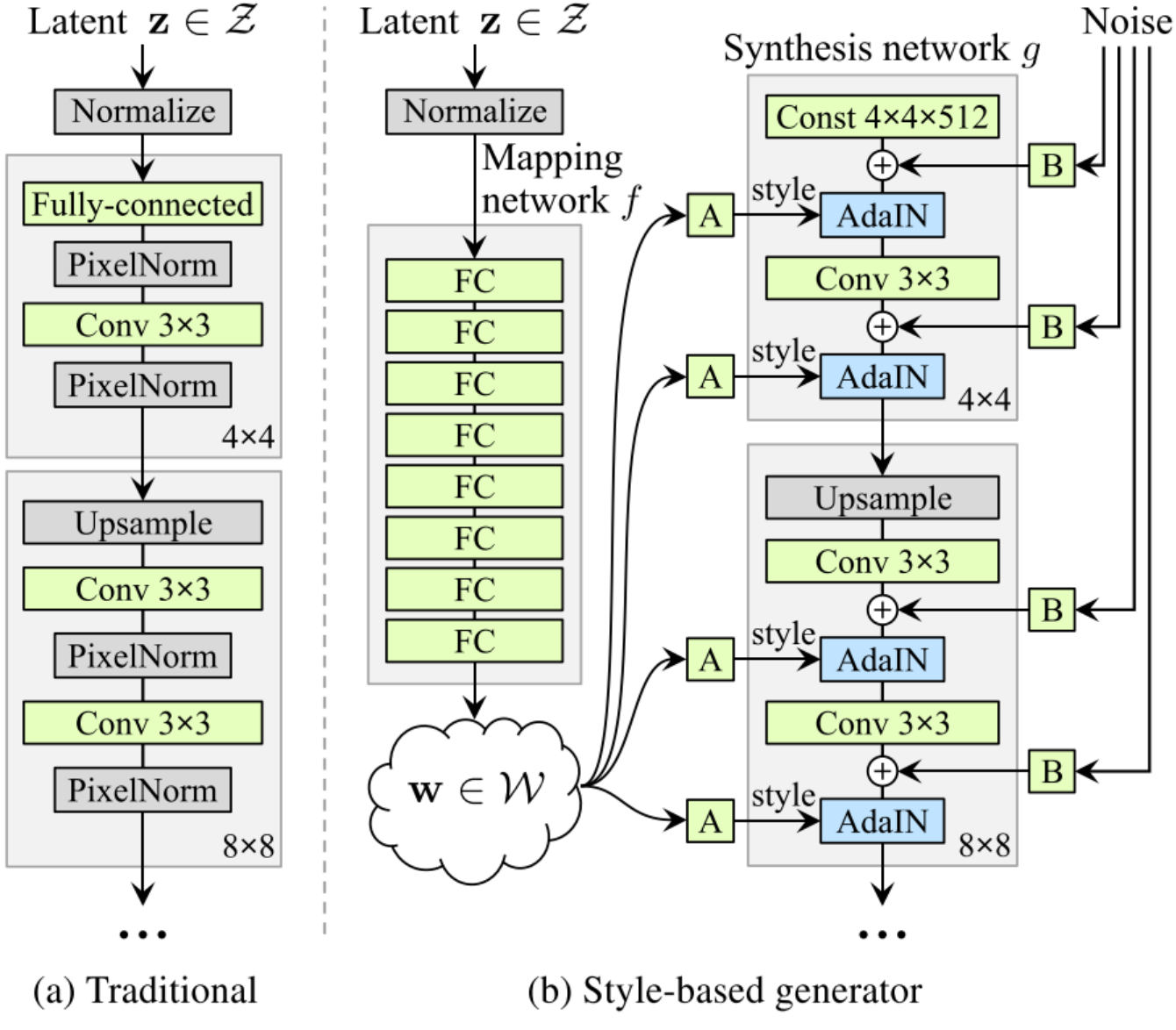


Why MMD GANs?

- "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]

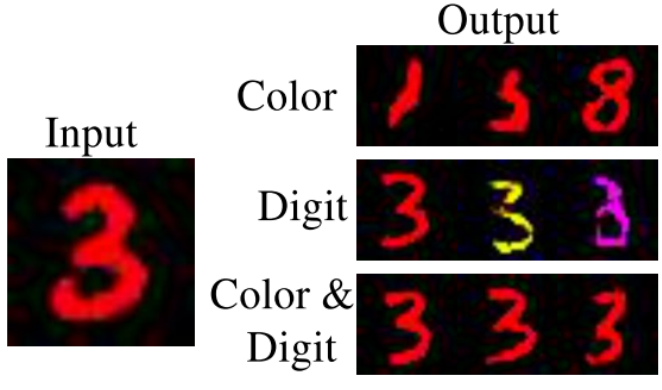
.

Finding samples you want [Jitkrittum+ ICML-19]

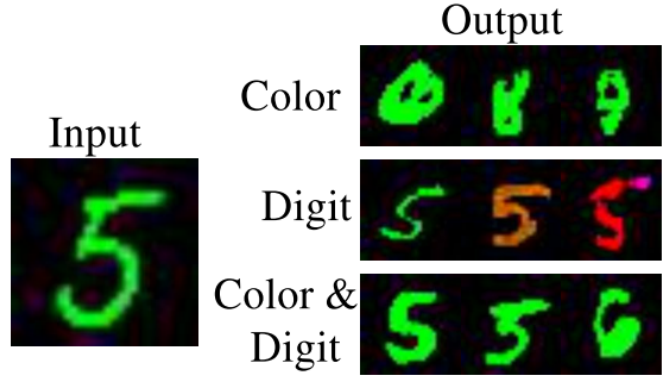
If we want to find "more samples like $\{X\}$":

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

(a) Samples from DCGAN

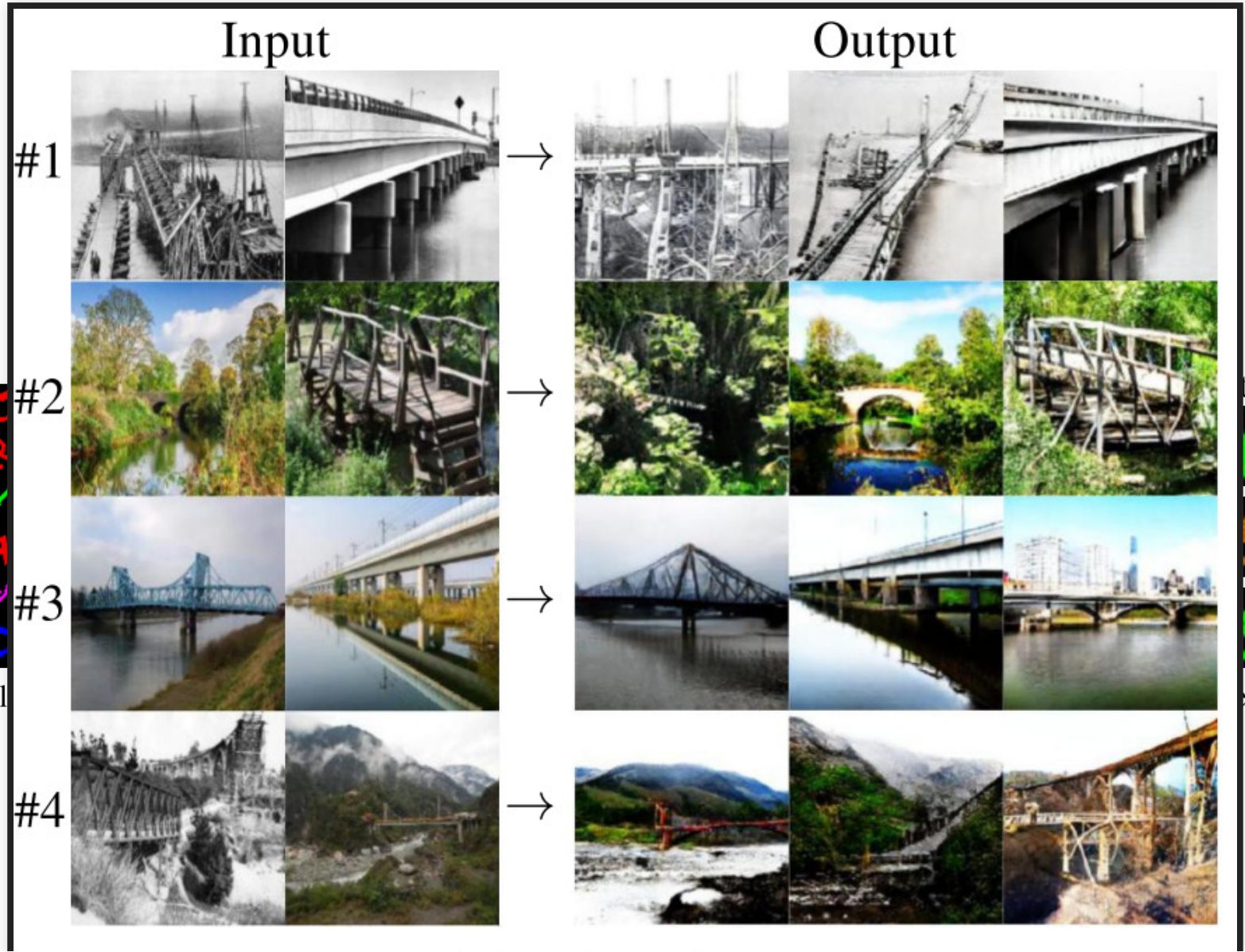


(b) Input: digit 3 in red



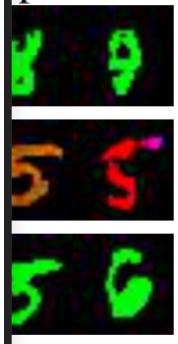
(c) Input: digit 5 in green

Finding samples you want [Jitkrittum+ ICML-19]



(a) Sampl

tput



een

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]

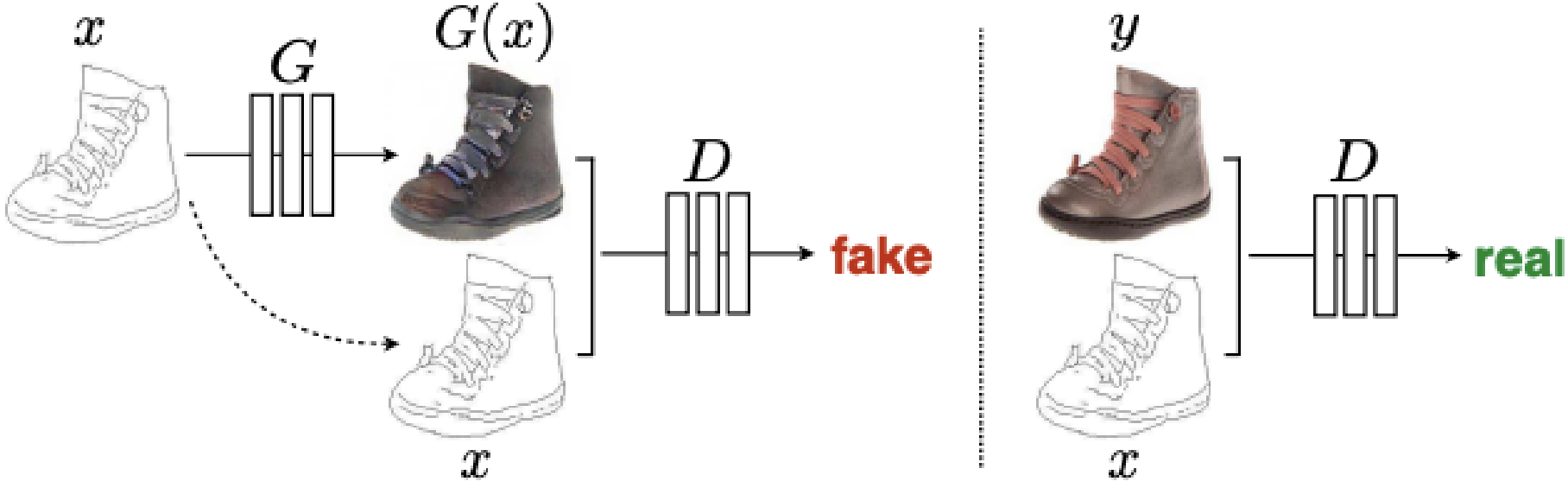
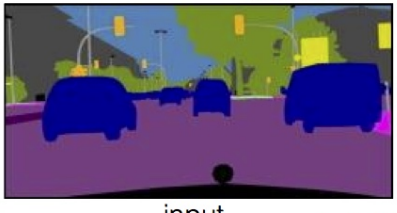


Figure 2: Training a conditional GAN to map edges→photo. The discriminator, D, learns to classify between fake (synthesized by the generator) and real {edge, photo} tuples. The generator, G, learns to fool the discriminator. Unlike an unconditional GAN, both the generator and discriminator observe the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

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

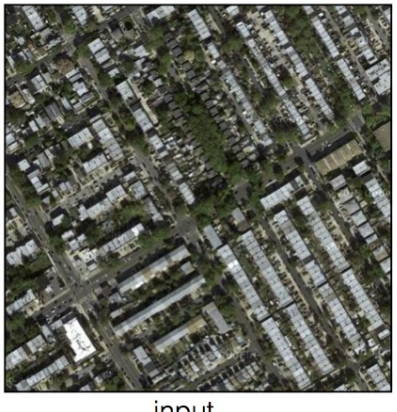
Labels to Street Scene



input

output

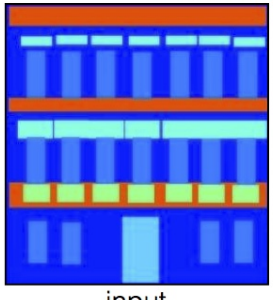
Aerial to Map



input

output

Labels to Facade



input

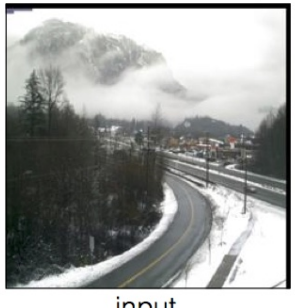
output

BW to Color

input

output

Day to Night



input

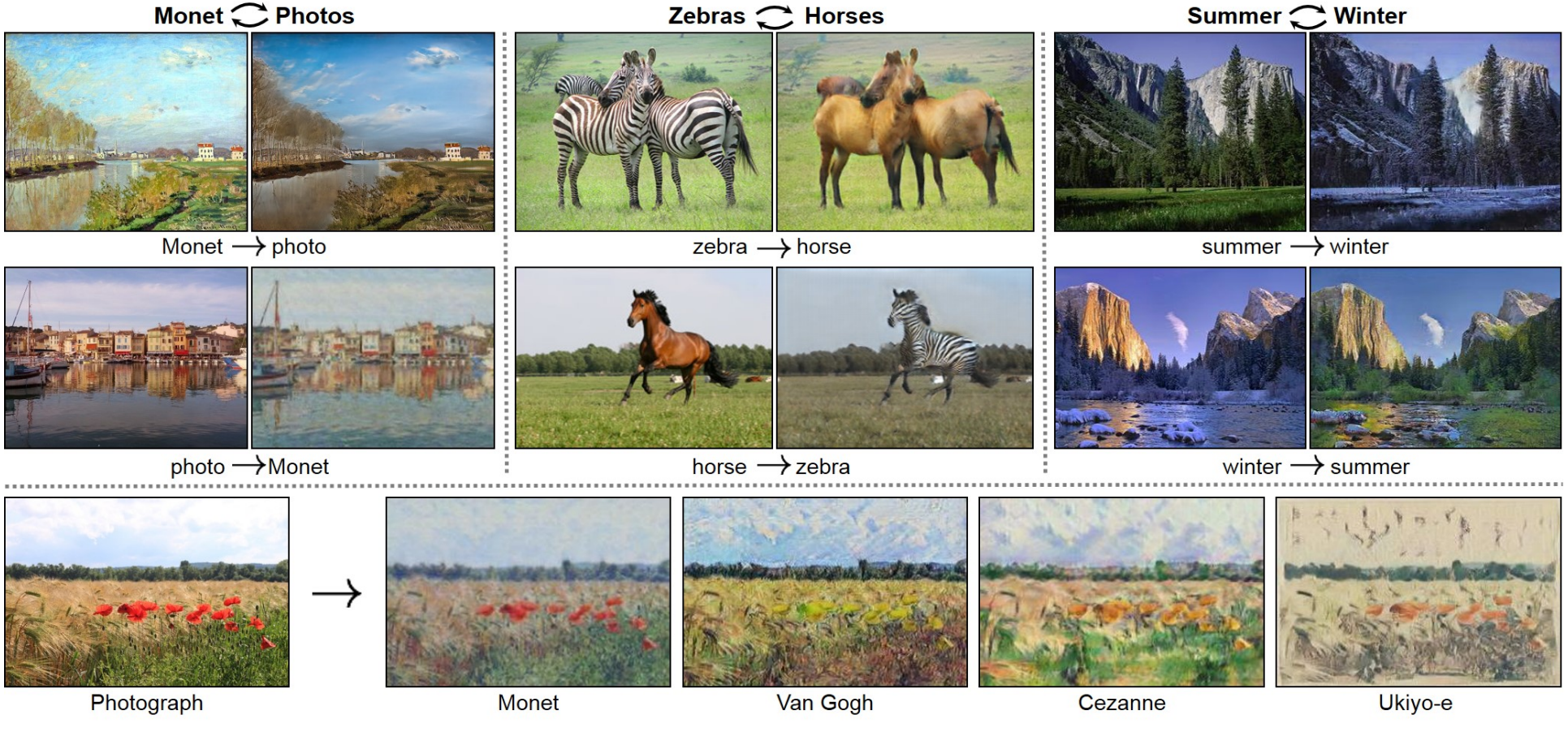
output

Edges to Photo

input

output

CycleGAN [Zhu+ ICCV-17]



Pose-to-image translation [Chan+ 2018]

.

YouTube: Everybody Dance Now

DeepFakes

YouTube: Mark Zuckerberg 'deepfake' will remain online

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.