Advanced Topics in Machine Learning
(deep generative modelling)

Lecture 5: Adversarial objectives for generative models

Nikolay Malkin

10 February 2026

Outline of Lecture 5

Adversarial objectives for generative models:

- Review and ingredients
  - Saddle-point optimisation
- Generative adversarial networks
  - Failure modes and solutions
- Summary of models seen so far

- Review and ingredients
  - Saddle-point optimisation
- Generative adversarial networks
  - Failure modes and solutions
- Summary of models seen so far

Jensen-Shannon divergence

$$\mathrm{KL}(p \,\|\, q) = \mathbb{E}_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right], \quad \mathrm{KL}(q \,\|\, p) = \mathbb{E}_{x\sim q}\left[\log \frac{q(x)}{p(x)}\right]$$

Jensen-Shannon divergence

$$\mathrm{KL}(p \,\|\, q) = \mathbb{E}_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right], \quad \mathrm{KL}(q \,\|\, p) = \mathbb{E}_{x\sim q}\left[\log \frac{q(x)}{p(x)}\right]$$

A compromise between forward and reverse KL: the **Jensen–Shannon (JS) divergence**

$$\mathrm{JS}(p, q) = \frac{1}{2}\mathrm{KL}\left(p \middle\| \frac{p+q}{2}\right) + \frac{1}{2}\mathrm{KL}\left(q \middle\| \frac{p+q}{2}\right)$$

- $\mathrm{JS}(p, q) \geq 0$, with equality only if $p = q$ as distributions
- $\mathrm{JS}(p, q) = \mathrm{JS}(q, p)$
- $0 \leq \mathrm{JS}(p, q) \leq \log 2$ (or ≤ 1, if using base-2 log)
- $\sqrt{\mathrm{JS}(p, q)}$ is a metric on the space of probability distributions

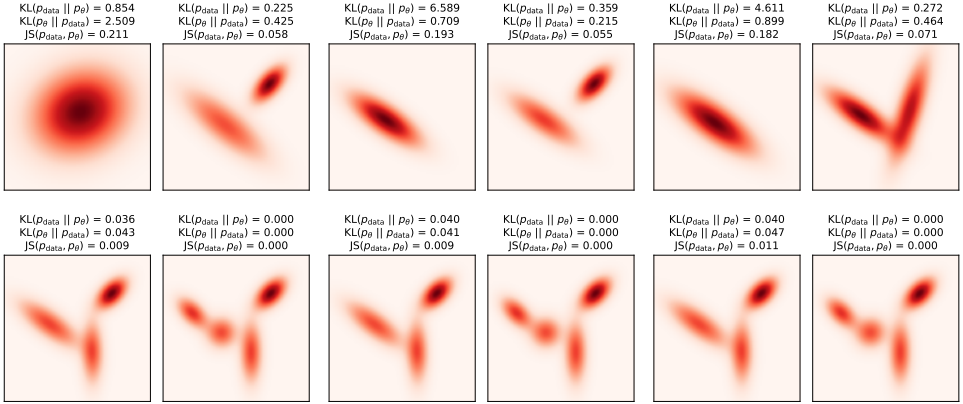
Summary of three divergences considered

Which divergence to use for generative modelling, if all are possible?

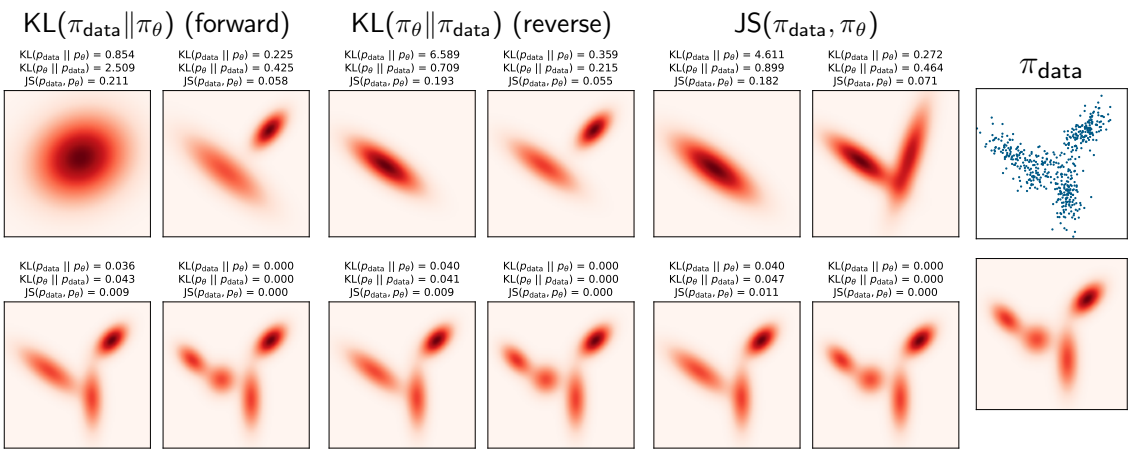
Summary of three divergences considered

Which divergence to use for generative modelling, if all are possible?

$\mathrm{KL}(\pi_{\text{data}} \| \pi_\theta)$ (forward) $\mathrm{KL}(\pi_\theta \| \pi_{\text{data}})$ (reverse) $\mathrm{JS}(\pi_{\text{data}}, \pi_\theta)$



Summary of three divergences considered



- Forward KL / MLE: **mode-covering** (high diversity, low fidelity)
- Reverse KL: **mode-seeking** (high fidelity, low diversity)

Adversarial optimisation

Typical ML optimisation problem (*e.g.*, normalising flow training):

$$\min_{h \in \mathcal{H}} f(h)$$

Adversarial optimisation

Typical ML optimisation problem (*e.g.*, normalising flow training):

$$\min_{h \in \mathcal{H}} f(h)$$

With auxiliary objects (*e.g.*, VAE training):

$$\min_{h \in \mathcal{H}} \min_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} f(h, h_{\text{aux}}) = \min_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} \min_{h \in \mathcal{H}} f(h, h_{\text{aux}})$$

Adversarial optimisation

Typical ML optimisation problem (*e.g.*, normalising flow training):

$$\min_{h \in \mathcal{H}} f(h)$$

With auxiliary objects (*e.g.*, VAE training):

$$\min_{h \in \mathcal{H}} \min_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} f(h, h_{\text{aux}}) = \min_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} \min_{h \in \mathcal{H}} f(h, h_{\text{aux}})$$

Adversarial optimisation:

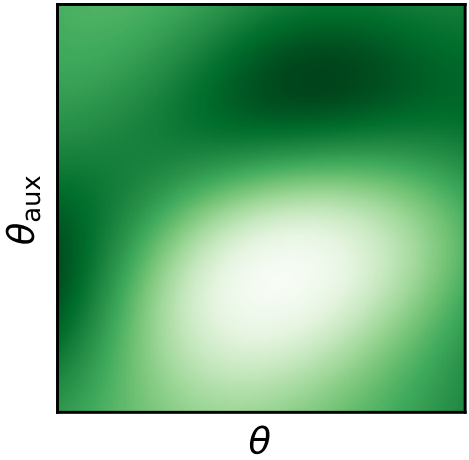
$$\min_{h \in \mathcal{H}} \max_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} f(h, h_{\text{aux}})$$

$$\max_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} \min_{h \in \mathcal{H}} f(h, h_{\text{aux}})$$

Adversarial optimisation

$$\min_{h \in \mathcal{H}} \max_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} f(h, h_{\text{aux}})$$

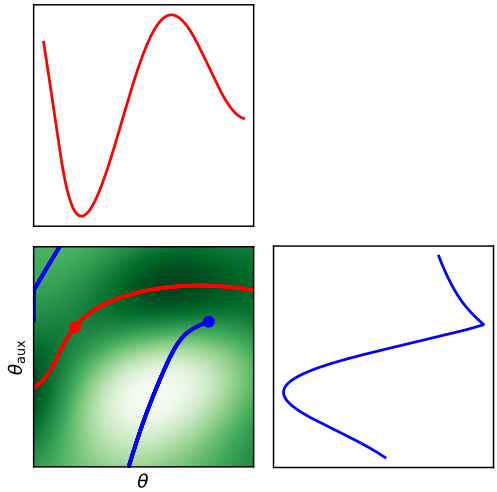
$$\max_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} \min_{h \in \mathcal{H}} f(h, h_{\text{aux}})$$



Adversarial optimisation

$$\min_{h \in \mathcal{H}} \max_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} f(h, h_{\text{aux}})$$

$$\max_{h_{\text{aux}} \in \mathcal{H}_{\text{aux}}} \min_{h \in \mathcal{H}} f(h, h_{\text{aux}})$$



The two problems (max-min and min-max) are not equivalent in general

Saddle-point optimisation with gradients

Solving $\min_\theta \max_\phi f(\theta, \phi)$ with gradient descent:

- Gradient steps on θ: $\theta \leftarrow \theta - \eta \nabla_\theta f(\theta, \phi)$
- Gradient steps on ϕ: $\phi \leftarrow \phi + \eta \nabla_\phi f(\theta, \phi)$

Saddle-point optimisation with gradients

Solving $\min_\theta \max_\phi f(\theta, \phi)$ with gradient descent:

- Gradient steps on θ: $\theta \leftarrow \theta - \eta \nabla_\theta f(\theta, \phi)$
- Gradient steps on ϕ: $\phi \leftarrow \phi + \eta \nabla_\phi f(\theta, \phi)$

At a stationary point (assuming some smoothness conditions):

- With ϕ fixed, θ is a local minimum of $f(\cdot, \phi)$
- With θ fixed, ϕ is a local maximum of $f(\theta, \cdot)$

Saddle-point optimisation with gradients

Solving $\min_\theta \max_\phi f(\theta, \phi)$ with gradient descent:

- Gradient steps on θ: $\theta \leftarrow \theta - \eta \nabla_\theta f(\theta, \phi)$
- Gradient steps on ϕ: $\phi \leftarrow \phi + \eta \nabla_\phi f(\theta, \phi)$

At a stationary point (assuming some smoothness conditions):

- With ϕ fixed, θ is a local minimum of $f(\cdot, \phi)$
- With θ fixed, ϕ is a local maximum of $f(\theta, \cdot)$
- $\nabla f(\theta, \phi) = 0$

Saddle-point optimisation with gradients

Solving $\min_\theta \max_\phi f(\theta, \phi)$ with gradient descent:

- Gradient steps on θ: $\theta \leftarrow \theta - \eta \nabla_\theta f(\theta, \phi)$
- Gradient steps on ϕ: $\phi \leftarrow \phi + \eta \nabla_\phi f(\theta, \phi)$

At a stationary point (assuming some smoothness conditions):

- With ϕ fixed, θ is a local minimum of $f(\cdot, \phi)$
- With θ fixed, ϕ is a local maximum of $f(\theta, \cdot)$
- $\nabla f(\theta, \phi) = 0$ and the Hessian $\nabla^2 f(\theta, \phi)$ has the structure

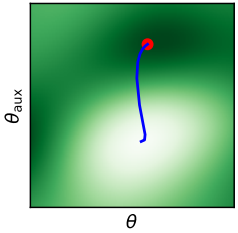
$$\begin{pmatrix} > 0 & * \\ * & < 0 \end{pmatrix}$$

and has both positive and negative eigenvalues (saddle point)

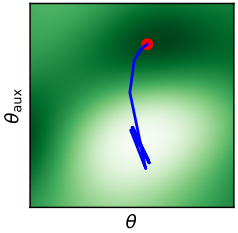
Some instabilities may arise; various techniques exist to mitigate them (*e.g.*, learning rates, extra[polated] gradient methods, regularisation, etc.)

Instability of saddle-point optimisation

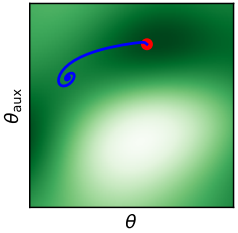
Cooperative
$\min_\theta \min_{\theta_{\text{aux}}}$
$\eta = 0.2$



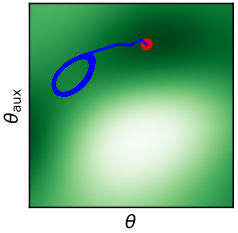
Cooperative
$\min_\theta \min_{\theta_{\text{aux}}}$
$\eta = 0.5$



Adversarial
$\min_\theta \max_{\theta_{\text{aux}}}$
$\eta = 0.2$



Adversarial
$\min_\theta \max_{\theta_{\text{aux}}}$
$\eta = 0.5$



- Review and ingredients
  - Saddle-point optimisation
- Generative adversarial networks
  - Failure modes and solutions
- Summary of models seen so far

Pushforward models without tractable density

We would like to build pushforward models $p_\theta = (f_\theta)_\# p_{\text{noise}}$ ($z \sim p_{\text{noise}}(z)$, $x = f_\theta(z)$) without the constraints of normalising flows:

- f_θ not invertible
- Base distribution p_{noise} not over the same space as the data distribution

Pushforward models without tractable density

We would like to build pushforward models $p_\theta = (f_\theta)_\# p_{\text{noise}}$ ($z \sim p_{\text{noise}}(z)$, $x = f_\theta(z)$) without the constraints of normalising flows:

- f_θ not invertible
- Base distribution p_{noise} not over the same space as the data distribution
  - If the latent is low-dimensional, the pushforward does not have full support and does not have a density
  - Can it still model the data distribution well?

Pushforward models without tractable density

We would like to build pushforward models $p_\theta = (f_\theta)_\# p_{\text{noise}}$ ($z \sim p_{\text{noise}}(z)$, $x = f_\theta(z)$) without the constraints of normalising flows:

- f_θ not invertible
- Base distribution p_{noise} not over the same space as the data distribution
  - If the latent is low-dimensional, the pushforward does not have full support and does not have a density
  - Can it still model the data distribution well? Yes, if the data lies on/near a low-dimensional manifold in the data space.

Pushforward models without tractable density

We would like to build pushforward models $p_\theta = (f_\theta)_\# p_{\text{noise}}$ ($z \sim p_{\text{noise}}(z)$, $x = f_\theta(z)$) without the constraints of normalising flows:

- f_θ not invertible
- Base distribution p_{noise} not over the same space as the data distribution
  - If the latent is low-dimensional, the pushforward does not have full support and does not have a density
  - Can it still model the data distribution well? Yes, if the data lies on/near a low-dimensional manifold in the data space.

The manifold hypothesis: (interesting) data lies on/near a low-dimensional manifold in the data space, so a low-dimensional z can be used

- A small number of latent factors of variation explain the data

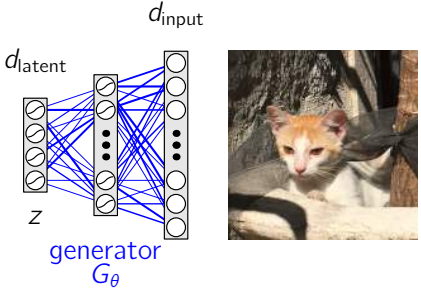
Generative adversarial networks: Overview

Two objects: generator G_θ and discriminator (binary classifier) D_ϕ:

Generative adversarial networks: Overview

Two objects: generator G_θ and discriminator (binary classifier) D_ϕ:

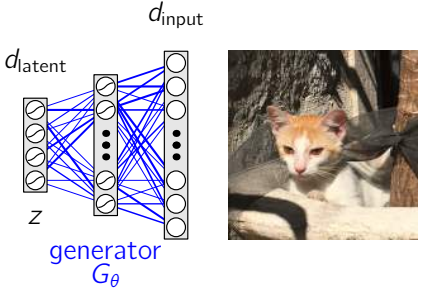
- $G_\theta : \mathbb{R}^{d_{\text{latent}}} \to \mathbb{R}^{d_{\text{data}}}$ generates samples $x = G_\theta(z)$ from noise $z \sim p_{\text{noise}}(z) = \mathcal{N}(x; 0, I_{d_{\text{latent}}})$

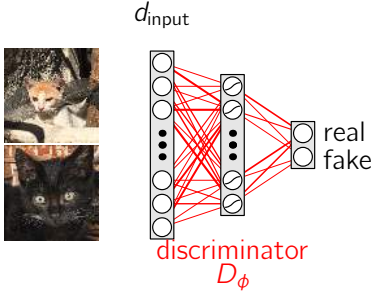


Generative adversarial networks: Overview

Two objects: generator G_θ and discriminator (binary classifier) D_ϕ:

- $G_\theta : \mathbb{R}^{d_{\text{latent}}} \to \mathbb{R}^{d_{\text{data}}}$ generates samples $x = G_\theta(z)$ from noise $z \sim p_{\text{noise}}(z) = \mathcal{N}(x; 0, I_{d_{\text{latent}}})$
- D_ϕ tries to distinguish between real samples $x \sim \pi_{\text{data}}$ and generated samples $x \sim p_\theta = (G_\theta)_{\#} p_{\text{noise}}$

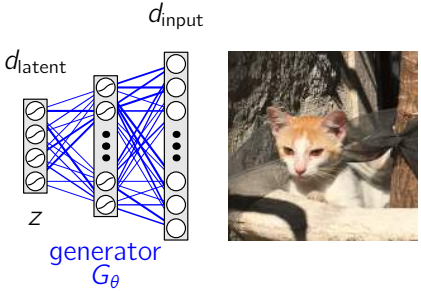


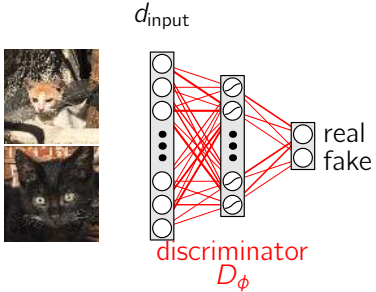


Generative adversarial networks: Overview

Two objects: generator G_θ and discriminator (binary classifier) D_ϕ:

- $G_\theta : \mathbb{R}^{d_{\text{latent}}} \to \mathbb{R}^{d_{\text{data}}}$ generates samples $x = G_\theta(z)$ from noise $z \sim p_{\text{noise}}(z) = \mathcal{N}(x; 0, I_{d_{\text{latent}}})$
- D_ϕ tries to distinguish between real samples $x \sim \pi_{\text{data}}$ and generated samples $x \sim p_\theta = (G_\theta)_{\#} p_{\text{noise}}$
- G_θ trained to fool D_ϕ (make it predict generated samples as real)





GAN learning objective

- Basic form of the GAN objective:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \underbrace{\mathbb{E}_{z \sim p_{\text{noise}}} [\log D_\phi(\text{fake} \mid G_\theta(z))]}_{= \mathbb{E}_{x \sim (G_\theta)_{\# p_{\text{noise}}}} \log D_\phi(\text{fake} \mid x)}$$

GAN learning objective

- Basic form of the GAN objective:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \underbrace{\mathbb{E}_{z \sim p_{\text{noise}}} [\log D_\phi(\text{fake} \mid G_\theta(z))]}_{= \mathbb{E}_{x \sim (G_\theta)_{\# p_{\text{noise}}}} \log D_\phi(\text{fake} \mid x)}$$

- Which parts of the objective depend on θ and ϕ?

GAN learning objective

- Basic form of the GAN objective:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \underbrace{\mathbb{E}_{z \sim p_{\text{noise}}} [\log D_\phi(\text{fake} \mid G_\theta(z))]}_{= \mathbb{E}_{x \sim (G_\theta)_{\# p_{\text{noise}}}} \log D_\phi(\text{fake} \mid x)}$$

- Which parts of the objective depend on θ and ϕ? Both depend on ϕ, only second term on θ.

GAN learning objective

- Basic form of the GAN objective:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \underbrace{\mathbb{E}_{z \sim p_{\text{noise}}} [\log D_\phi(\text{fake} \mid G_\theta(z))]}_{= \mathbb{E}_{x \sim (G_\theta)_{\# p_{\text{noise}}}} \log D_\phi(\text{fake} \mid x)}$$

- Which parts of the objective depend on θ and ϕ? Both depend on ϕ, only second term on θ.
- Qualitative analysis of dynamics (naïve):
  - What happens if G_θ produces samples that are very different from real data?

GAN learning objective

- Basic form of the GAN objective:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \underbrace{\mathbb{E}_{z \sim p_{\text{noise}}} [\log D_\phi(\text{fake} \mid G_\theta(z))]}_{= \mathbb{E}_{x \sim (G_\theta)_{\# p_{\text{noise}}}} \log D_\phi(\text{fake} \mid x)}$$

- Which parts of the objective depend on θ and ϕ? Both depend on ϕ, only second term on θ.
- Qualitative analysis of dynamics (naïve):
  - What happens if G_θ produces samples that are very different from real data? D_ϕ can easily distinguish them and pushes G_θ away from them.

GAN learning objective

- Basic form of the GAN objective:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \underbrace{\mathbb{E}_{z \sim p_{\text{noise}}} [\log D_\phi(\text{fake} \mid G_\theta(z))]}_{= \mathbb{E}_{x \sim (G_\theta)_{\# p_{\text{noise}}}} \log D_\phi(\text{fake}|x)}$$

- Which parts of the objective depend on θ and ϕ? Both depend on ϕ, only second term on θ.
- Qualitative analysis of dynamics (naïve):
  - What happens if G_θ produces samples that are very different from real data? D_ϕ can easily distinguish them and pushes G_θ away from them.
  - What happens if G_θ misses modes of the data distribution?

GAN learning objective

- Basic form of the GAN objective:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \underbrace{\mathbb{E}_{z \sim p_{\text{noise}}} [\log D_\phi(\text{fake} \mid G_\theta(z))]}_{= \mathbb{E}_{x \sim (G_\theta)_{\# p_{\text{noise}}}} \log D_\phi(\text{fake}|x)}$$

- Which parts of the objective depend on θ and ϕ? Both depend on ϕ, only second term on θ.
- Qualitative analysis of dynamics (naïve):
  - What happens if G_θ produces samples that are very different from real data? D_ϕ can easily distinguish them and pushes G_θ away from them.
  - What happens if G_θ misses modes of the data distribution? D_ϕ learns to distinguish them and pushes G_θ away from them.

GAN learning objective

What do GANs really optimise?

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_{\phi}(\text{real} \mid x)] + \mathbb{E}_{x \sim p_{\theta}} [\log D_{\phi}(\text{fake} \mid x)]$$

What do GANs really optimise?

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \mathbb{E}_{x \sim p_\theta} [\log D_\phi(\text{fake} \mid x)]$$

- Suppose π_{data} and p_θ have densities; then the optimal D for a fixed G_θ is

$$D(\text{real} \mid x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\theta(x)}, \quad D(\text{fake} \mid x) = \frac{p_\theta(x)}{p_{\text{data}}(x) + p_\theta(x)}$$

What do GANs really optimise?

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \mathbb{E}_{x \sim p_\theta} [\log D_\phi(\text{fake} \mid x)]$$

- Suppose π_{data} and p_θ have densities; then the optimal D for a fixed G_θ is

$$D(\text{real} \mid x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\theta(x)}, \quad D(\text{fake} \mid x) = \frac{p_\theta(x)}{p_{\text{data}}(x) + p_\theta(x)}$$

- If the discriminator always remains optimal, the objective becomes

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

What do GANs really optimise?

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim \pi_{\text{data}}} [\log D_\phi(\text{real} \mid x)] + \mathbb{E}_{x \sim p_\theta} [\log D_\phi(\text{fake} \mid x)]$$

- Suppose π_{data} and p_θ have densities; then the optimal D for a fixed G_θ is

$$D(\text{real} \mid x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\theta(x)}, \quad D(\text{fake} \mid x) = \frac{p_\theta(x)}{p_{\text{data}}(x) + p_\theta(x)}$$

- If the discriminator always remains optimal, the objective becomes

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

$$= \mathbb{E}_{x \sim p_{\text{data}}} \left[\log \frac{p_{\text{data}}(x)}{\frac{1}{2}(p_{\text{data}}(x) + p_\theta(x))} \right] + \mathbb{E}_{x \sim p_\theta} \left[\log \frac{p_\theta(x)}{\frac{1}{2}(p_{\text{data}}(x) + p_\theta(x))} \right] - \log 4$$

What do GANs really optimise?

- Suppose π_{data} and p_θ have densities; then the optimal D for a fixed G_θ is

$$D(\text{real} \mid x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\theta(x)}, \quad D(\text{fake} \mid x) = \frac{p_\theta(x)}{p_{\text{data}}(x) + p_\theta(x)}$$

- If the discriminator always remains optimal, the objective becomes

$$\begin{aligned}
&\mathbb{E}_{x \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\theta(x)}\right] + \mathbb{E}_{x \sim p_\theta}\left[\log \frac{p_\theta(x)}{p_{\text{data}}(x) + p_\theta(x)}\right] \\
&= \mathbb{E}_{x \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{\frac{1}{2}(p_{\text{data}}(x) + p_\theta(x))}\right] + \mathbb{E}_{x \sim p_\theta}\left[\log \frac{p_\theta(x)}{\frac{1}{2}(p_{\text{data}}(x) + p_\theta(x))}\right] - \log 4 \\
&= 2\text{JS}(p_{\text{data}}, p_\theta) - \log 4.
\end{aligned}$$

What do GANs really optimise?

- Suppose π_{data} and p_θ have densities; then the optimal D for a fixed G_θ is

$$D(\text{real} \mid x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\theta(x)}, \quad D(\text{fake} \mid x) = \frac{p_\theta(x)}{p_{\text{data}}(x) + p_\theta(x)}$$

- If the discriminator always remains optimal, the objective becomes

$$\begin{aligned}
&\mathbb{E}_{x \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\theta(x)}\right] + \mathbb{E}_{x \sim p_\theta}\left[\log \frac{p_\theta(x)}{p_{\text{data}}(x) + p_\theta(x)}\right] \\
&= \mathbb{E}_{x \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{\frac{1}{2}(p_{\text{data}}(x) + p_\theta(x))}\right] + \mathbb{E}_{x \sim p_\theta}\left[\log \frac{p_\theta(x)}{\frac{1}{2}(p_{\text{data}}(x) + p_\theta(x))}\right] - \log 4 \\
&= 2\text{JS}(p_{\text{data}}, p_\theta) - \log 4.
\end{aligned}$$

- Under 'perfect' optimisation of D, GANs minimise the Jensen-Shannon divergence between π_{data} and p_θ

Some successes of GANs

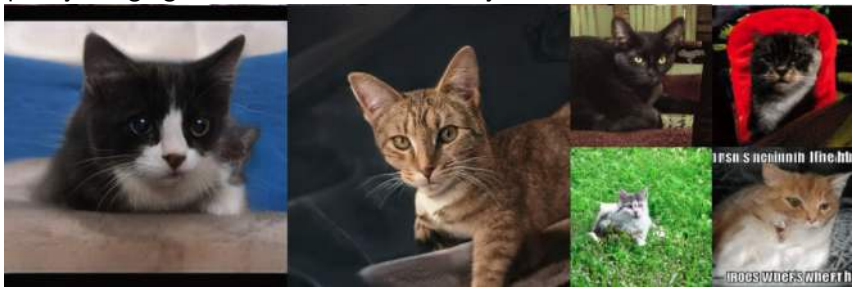
- High quality of generated samples (especially for images)
  - Vision inductive biases in D_ϕ $\rightsquigarrow$ good perceptual quality in human evaluation
- Dominant approach for image generation around 2015–2021, still used

Some successes of GANs

- High quality of generated samples (especially for images)
  - Vision inductive biases in D_ϕ $\rightsquigarrow$ good perceptual quality in human evaluation
- Dominant approach for image generation around 2015–2021, still used

Some successes of GANs

- Dominant approach for image generation around 2015–2021, still used
  - StyleGAN (2018), architecture separating 'content' and 'style' in the latent space $\rightsquigarrow$ high-quality image generation and controllability



[Karras et al., CVPR'19]

Some successes of GANs

- Dominant approach for image generation around 2015–2021, still used
  - StyleGAN (2018), architecture separating 'content' and 'style' in the latent space $\rightsquigarrow$ high-quality image generation and controllability
  - BigGAN (2018), class-conditional model, stability improvements

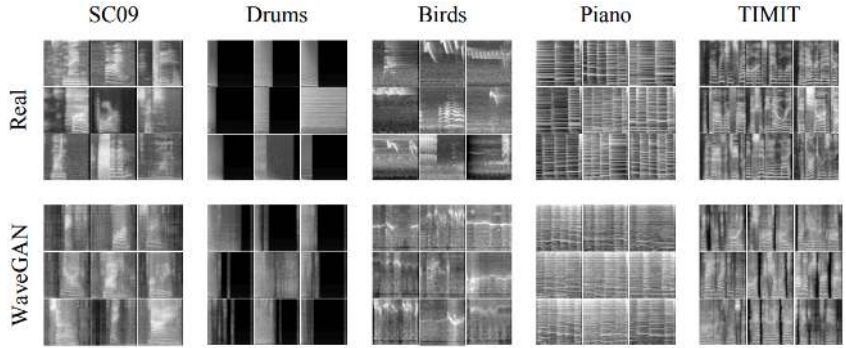
[Brock et al., ICLR'19]

Some successes of GANs

- Dominant approach for image generation around 2015–2021, still used
  - StyleGAN (2018), architecture separating 'content' and 'style' in the latent space $\rightsquigarrow$ high-quality image generation and controllability
  - BigGAN (2018), class-conditional model, stability improvements
  - Non-noise source distributions: CycleGAN (2017) for unpaired translation

Some successes of GANs

- Dominant approach for image generation around 2015–2021, still used
  - StyleGAN (2018), architecture separating 'content' and 'style' in the latent space $\rightsquigarrow$ high-quality image generation and controllability
  - BigGAN (2018), class-conditional model, stability improvements
  - Non-noise source distributions: CycleGAN (2017) for unpaired translation
  - Also for non-image data, e.g., WaveGAN for audio



Some successes of GANs

- High quality of generated samples (especially for images)
  - Vision inductive biases in D_ϕ $\rightsquigarrow$ good perceptual quality in human evaluation
- Dominant approach for image generation around 2015–2021, still used
  - StyleGAN (2018), architecture separating 'content' and 'style' in the latent space $\rightsquigarrow$ high-quality image generation and controllability
  - BigGAN (2018), class-conditional model, stability improvements
  - Non-noise source distributions: CycleGAN (2017) for unpaired translation
  - Also for non-image data, e.g., WaveGAN for audio

However, GANs were never successful for some modalities (e.g., text) and have been superseded by diffusion models for many applications.

Some failures of the basic objective

Qualitative analysis of dynamics (pessimistic):

- What happens if G_θ produces samples that are very different from real data? D_ϕ can easily distinguish them and pushes G_θ away from them.
  - But if too different, the optimal discriminator can perfectly distinguish them $\rightsquigarrow$ the generator receives no gradient signal to improve (saturation)

Some failures of the basic objective

Qualitative analysis of dynamics (pessimistic):

- What happens if G_θ produces samples that are very different from real data? D_ϕ can easily distinguish them and pushes G_θ away from them.
  - But if too different, the optimal discriminator can perfectly distinguish them $\rightsquigarrow$ the generator receives no gradient signal to improve (saturation)
- What happens if G_θ misses modes of the data distribution? D_ϕ learns to distinguish them and pushes G_θ away from them.
  - If G_θ misses modes of the data distribution, the optimal discriminator does not need to learn to distinguish them well $\rightsquigarrow$ the generator receives no gradient signal to improve (mode collapse)

Some failures of the basic objective

Qualitative analysis of dynamics (pessimistic):

- What happens if G_θ produces samples that are very different from real data? D_ϕ can easily distinguish them and pushes G_θ away from them.
  - But if too different, the optimal discriminator can perfectly distinguish them $\rightsquigarrow$ the generator receives no gradient signal to improve (saturation)
- What happens if G_θ misses modes of the data distribution? D_ϕ learns to distinguish them and pushes G_θ away from them.
  - If G_θ misses modes of the data distribution, the optimal discriminator does not need to learn to distinguish them well $\rightsquigarrow$ the generator receives no gradient signal to improve (mode collapse)
- Instability near the saddle point (oscillations, divergence)

Some failures of the basic objective

Ideal case:

$10\times$ slower learning rate for discriminator:

$10\times$ slower learning rate for generator:

Some failures of the basic objective

Ideal case:

$10\times$ slower learning rate for discriminator:

$10\times$ slower learning rate for generator:

Some failures of the basic objective

Ideal case:

$10\times$ slower learning rate for discriminator:

$10\times$ slower learning rate for generator:

Some failures of the basic objective

Latent dimension too small to capture data distribution (+ oscillations):

Solutions to optimisation failures

- Optimisation tricks: learning rate scheduling, extragradient methods, ...

Solutions to optimisation failures

- Optimisation tricks: learning rate scheduling, extragradient methods, ...
- To mitigate saturation, use a 'non-saturating' generator loss:

$$\min_\theta \mathbb{E}_{z\sim p_{\text{noise}}} [\log D_\phi(\text{fake} \mid G_\theta(z))]$$

$$\rightsquigarrow \min_\theta \mathbb{E}_{z\sim p_{\text{noise}}} [-\log D_\phi(\text{real} \mid G_\theta(z))]$$

or other variants (Wasserstein GAN, f-GAN, MMD-GAN, etc.)

Solutions to optimisation failures

- Optimisation tricks: learning rate scheduling, extragradient methods, ...
- To mitigate saturation, use a 'non-saturating' generator loss:

$$\min_\theta \mathbb{E}_{z\sim p_{\text{noise}}} [\log D_\phi(\text{fake} \mid G_\theta(z))]$$

$$\rightsquigarrow \min_\theta \mathbb{E}_{z\sim p_{\text{noise}}} [-\log D_\phi(\text{real} \mid G_\theta(z))]$$

  or other variants (Wasserstein GAN, f-GAN, MMD-GAN, etc.)
- Various regularisation terms proposed to mitigate instability

['The GAN is dead; long live the GAN!', NeurIPS'24]

- Review and ingredients
  - Saddle-point optimisation
- Generative adversarial networks
  - Failure modes and solutions
- Summary of models seen so far

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation			

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation	approximate	exact	no

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation	approximate	exact	no
Sample quality (fidelity)			

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation	approximate	exact	no
Sample quality (fidelity)	poor	good	best

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation	approximate	exact	no
Sample quality (fidelity)	poor	good	best
Mode coverage (diversity)			

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation	approximate	exact	no
Sample quality (fidelity)	poor	good	best
Mode coverage (diversity)	good	good	poor

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation	approximate	exact	no
Sample quality (fidelity)	poor	good	best
Mode coverage (diversity)	good	good	poor
Trainability at scale			

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation	approximate	exact	no
Sample quality (fidelity)	poor	good	best
Mode coverage (diversity)	good	good	poor
Trainability at scale	best	good	poor

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation	approximate	exact	no
Sample quality (fidelity)	poor	good	best
Mode coverage (diversity)	good	good	poor
Trainability at scale	best	good	poor
Ability to impose inductive biases			

Summary of models seen so far

We have seen three classes of deep generative models: VAEs, normalising flows, GANs

	VAE	NF	GAN
Likelihood estimation	approximate	exact	no
Sample quality (fidelity)	poor	good	best
Mode coverage (diversity)	good	good	poor
Trainability at scale	best	good	poor
Ability to impose inductive biases	good	poor	good

Conclusion and looking ahead

- GANs are good sample generators if they can be stably optimised (which is not always the case)
  - But they do not have (tractable, or any) densities over the data space, so not suitable for all applications

Conclusion and looking ahead

- GANs are good sample generators if they can be stably optimised (which is not always the case)
  - But they do not have (tractable, or any) densities over the data space, so not suitable for all applications
- Next lecture (after the break): how to evaluate models when we cannot compute likelihoods

Conclusion and looking ahead

- GANs are good sample generators if they can be stably optimised (which is not always the case)
  - But they do not have (tractable, or any) densities over the data space, so not suitable for all applications
- Next lecture (after the break): how to evaluate models when we cannot compute likelihoods
- Then: diffusion and flow-based models, which combine ideas from the algorithm families seen so far