Generative Adversarial Networks

Learn to generate by playing a game against a critic

Prof. Nipun Batra

ES 667 · Deep Learning · IIT Gandhinagar

From a latent noise vector to a fake image

Recall the generative task

Q QUESTION

We want to turn simple noise into a realistic sample:

Recall the generative task

Q QUESTION

We want to turn simple noise into a realistic sample:

$$z \sim p(z) \quad \xrightarrow{G_\theta} \quad \tilde{x}$$

Recall the generative task

Q QUESTION

We want to turn simple noise into a realistic sample:

$$z \sim p(z) \quad \xrightarrow{G_\theta} \quad \tilde{x}$$

We need a **training signal** that says whether the sample $\tilde{x}$ looks like real data — but for a freshly drawn z there is **no target image** to compare against.

Recall the generative task

Q QUESTION

We want to turn simple noise into a realistic sample:

$$z \sim p(z) \quad \xrightarrow{G_\theta} \quad \tilde{x}$$

We need a **training signal** that says whether the sample $\tilde{x}$ looks like real data — but for a freshly drawn z there is **no target image** to compare against.

so where does the loss come from?

Why not use only pixel reconstruction?

Q QUESTION

The obvious idea — minimize pixel distance to some target — fails twice:

Why not use only pixel reconstruction?

Q QUESTION

The obvious idea — minimize pixel distance to some target — fails twice:

- a newly sampled z has **no** paired target image to reconstruct;

Why not use only pixel reconstruction?

Q QUESTION

The obvious idea — minimize pixel distance to some target — fails twice:

- a newly sampled z has **no** paired target image to reconstruct;
- even with a target, pixel distance rewards the **average** of plausible images, giving **blurry** results (exactly the VAE symptom from L19).

Why not use only pixel reconstruction?

Q QUESTION

The obvious idea — minimize pixel distance to some target — fails twice:

- a newly sampled z has **no** paired target image to reconstruct;
- even with a target, pixel distance rewards the **average** of plausible images, giving **blurry** results (exactly the VAE symptom from L19).

we need a signal for **realism**, not for pixel-wise proximity

The two-player setup

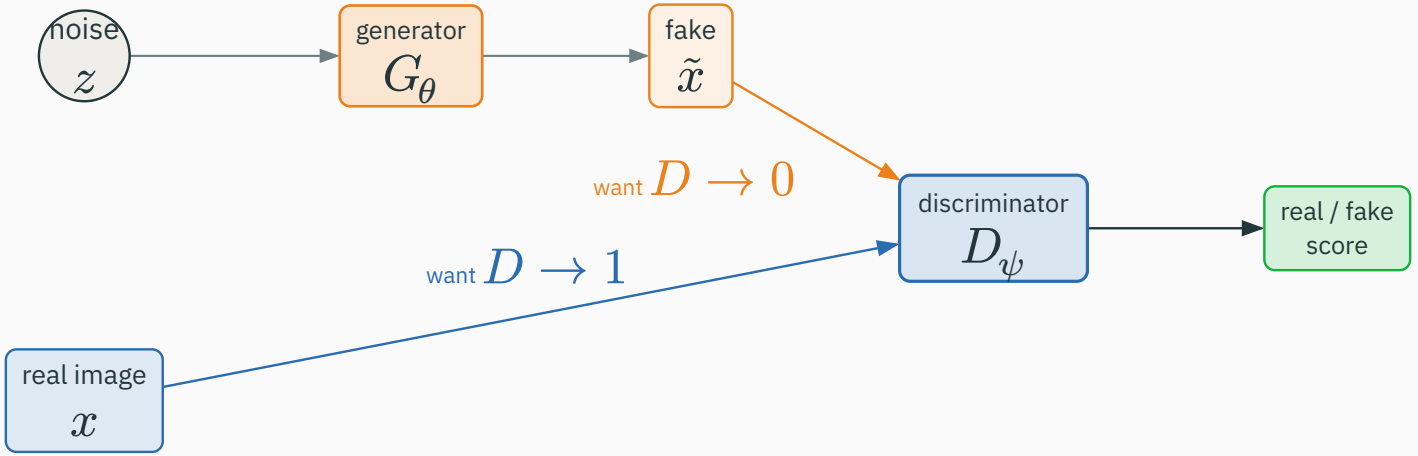
V VISUAL

Turn generation into a game between two networks:

The two-player setup

V VISUAL

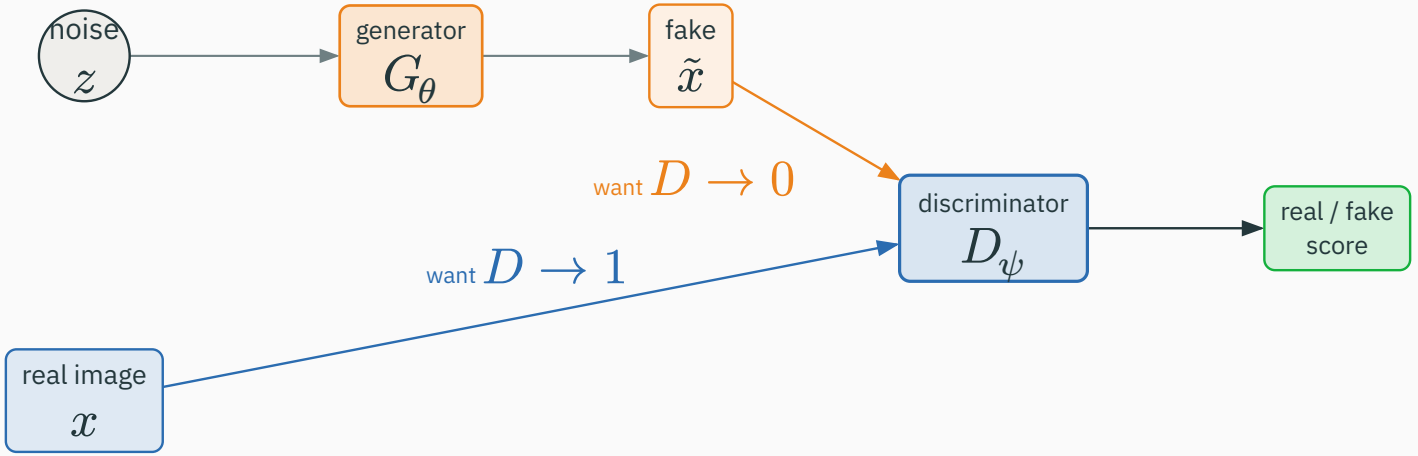
Turn generation into a game between two networks:



The two-player setup

V VISUAL

Turn generation into a game between two networks:



the generator makes fakes; the discriminator scores real vs fake — neither loss makes sense alone

The two roles

Two networks with opposing goals:

The two roles

D DERIVATION

Two networks with opposing goals:

G_θ — the counterfeiter. Maps simple noise z to samples $\tilde{x}$; wants the detective to call them real.

D_ψ — the detective. Separates real data from generated samples; wants to catch every fake.

The two roles

D DERIVATION

Two networks with opposing goals:

G_θ — the counterfeiter. Maps simple noise z to samples $\tilde{x}$; wants the detective to call them real.

D_ψ — the detective. Separates real data from generated samples; wants to catch every fake.

their **competition** supplies the learning signal — a learned notion of realism

A discriminator is a classifier

D DERIVATION

The discriminator is an ordinary binary classifier with a probability output:

A discriminator is a classifier

D DERIVATION

The discriminator is an ordinary binary classifier with a probability output:

$$D_{\psi(x)} \in (0,1) \approx P(\text{real} \mid x)$$

A discriminator is a classifier

D DERIVATION

The discriminator is an ordinary binary classifier with a probability output:

$$D_{\psi(x)} \in (0,1) \approx P(\text{real} \mid x)$$

label real examples **1**, generated examples **0** — the one twist is that the negative class keeps changing

Discriminator objective

D DERIVATION

Push D up on real data and down on fakes:

Discriminator objective

D DERIVATION

Push D up on real data and down on fakes:

$$\max_{\psi} \mathbb{E}_{x \sim p_{\text{data}}} \log D_{\psi(x)} + \mathbb{E}_{z \sim p(z)} \log\Big(1 - D_{\psi\left(G_{\theta(z)}\right)}\Big)$$

Discriminator objective

D DERIVATION

Push D up on real data and down on fakes:

$$\max_{\psi} \mathbb{E}_{x \sim p_{\text{data}}} \log D_{\psi(x)} + \mathbb{E}_{z \sim p(z)} \log\left(1 - D_{\psi\left(G_{\theta(z)}\right)}\right)$$

the first term rewards calling real data real; the second rewards calling fakes fake

The generator’s original objective

D DERIVATION

Freeze D; change the fakes so the detective calls them real:

The generator’s original objective

D DERIVATION

Freeze D; change the fakes so the detective calls them real:

$$\min_{\theta} \mathbb{E}_{z \sim p(z)} \log\Big(1 - D_{\psi\left(G_{\theta(z)}\right)}\Big)$$

The generator’s original objective

D DERIVATION

Freeze D; change the fakes so the detective calls them real:

$$\min_{\theta} \mathbb{E}_{z \sim p(z)} \log\Big(1 - D_{\psi\left(G_{\theta(z)}\right)}\Big)$$

Only the **second** term of the discriminator objective depends on θ. The generator has no access to real data — it only sees the discriminator’s verdict on its own samples.

Minimax, in words

D DERIVATION

Stack the two objectives into one value function:

Minimax, in words

D DERIVATION

Stack the two objectives into one value function:

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

Minimax, in words

D DERIVATION

Stack the two objectives into one value function:

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

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

Minimax, in words

D DERIVATION

Stack the two objectives into one value function:

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

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

the detective **maximizes**, the counterfeiter **minimizes** — the same scalar (Goodfellow et al., 2014)

Board sketch: alternating steps

I INTERACTIVE

The game plays out as an **alternation**, never a single descent:

Board sketch: alternating steps

I INTERACTIVE

The game plays out as an **alternation**, never a single descent:

1. fix D, nudge the **fakes** toward regions D currently calls real;

Board sketch: alternating steps

I INTERACTIVE

The game plays out as an **alternation**, never a single descent:

1. fix D, nudge the **fakes** toward regions D currently calls real;
2. fix G, move the **boundary** to reject those fakes again.

Board sketch: alternating steps

I INTERACTIVE

The game plays out as an **alternation**, never a single descent:

1. fix D, nudge the **fakes** toward regions D currently calls real;
2. fix G, move the **boundary** to reject those fakes again.

INTERACTIVE

↗ nipunbatra.github.io/interactive-articles/gan-minimax-dance

Step the minimax dance: watch generated samples drift toward the data while the discriminator boundary chases them — both players keep moving.

How GAN training actually works

One training iteration

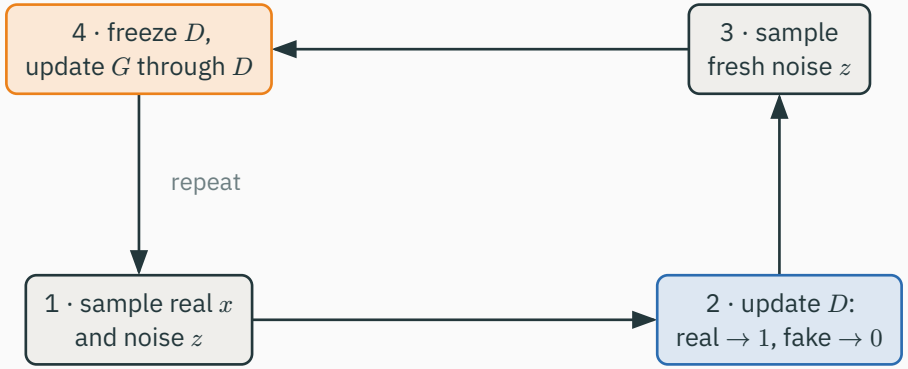
V VISUAL

Each step updates the two networks in turn:

One training iteration

V VISUAL

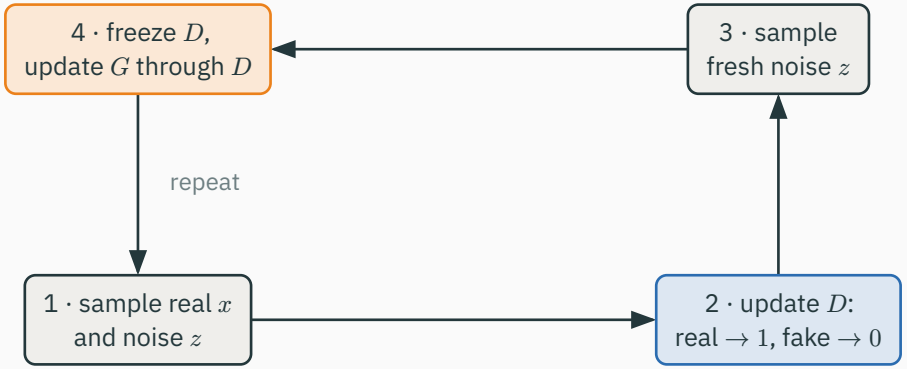
Each step updates the two networks in turn:



One training iteration

V VISUAL

Each step updates the two networks in turn:



update D on a fresh minibatch, then update G on fresh noise — then repeat

The gradient path to the generator

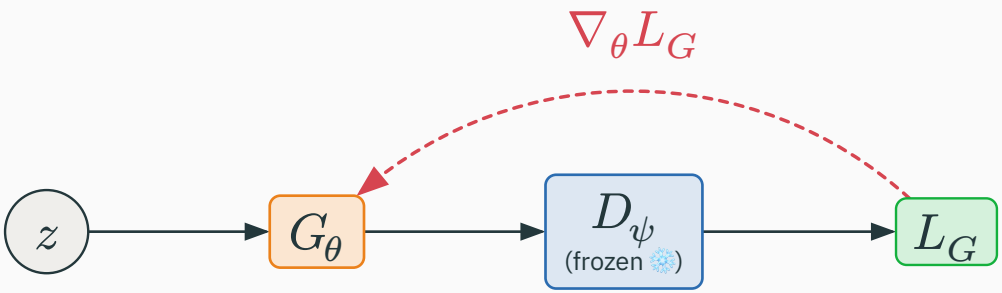
V VISUAL

The generator never touches real data; its signal arrives **through** the discriminator:

The gradient path to the generator

V VISUAL

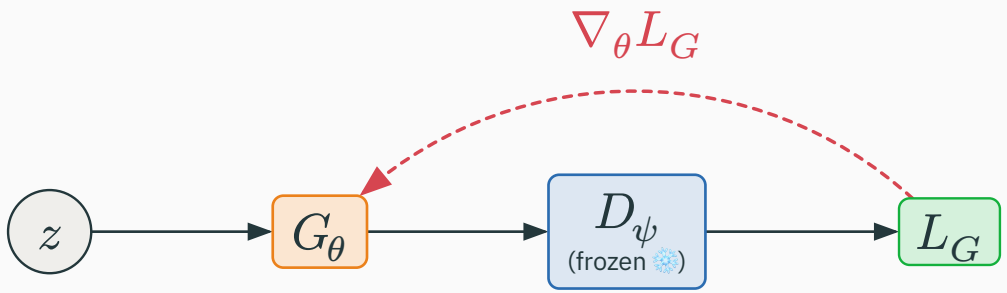
The generator never touches real data; its signal arrives **through** the discriminator:



The gradient path to the generator

V VISUAL

The generator never touches real data; its signal arrives **through** the discriminator:

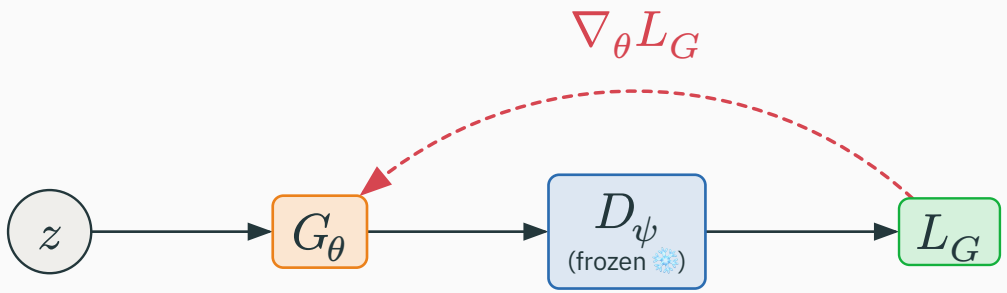


$$z \to G_{\theta(z)} \to D_{\psi\left(G_{\theta(z)}\right)} \to L_G$$

The gradient path to the generator

V VISUAL

The generator never touches real data; its signal arrives **through** the discriminator:



$$z \to G_{\theta(z)} \to D_{\psi\left(G_{\theta(z)}\right)} \to L_G$$

gradients pass through the **frozen** D into G; D's weights are held fixed, but its computation stays in the graph

The saturation problem

D DERIVATION

Early on, the detective easily spots fakes, so $D(G(z)) \approx 0$:

The saturation problem

D DERIVATION

Early on, the detective easily spots fakes, so $D(G(z)) \approx 0$:

$\log(1 - D(G(z)))$ is very **flat** near $D = 0$

The saturation problem

D DERIVATION

Early on, the detective easily spots fakes, so $D(G(z)) \approx 0$:

$\log(1 - D(G(z)))$ is very **flat** near $D = 0$

The original $\log(1 - D)$ generator objective gives a **tiny gradient** exactly when the generator is worst and needs the **strongest** push. The signal vanishes early.

Non-saturating generator loss

D DERIVATION

Keep the same goal — make fakes look real — but flip to a loss with a healthy gradient:

Non-saturating generator loss

D DERIVATION

Keep the same goal — make fakes look real — but flip to a loss with a healthy gradient:

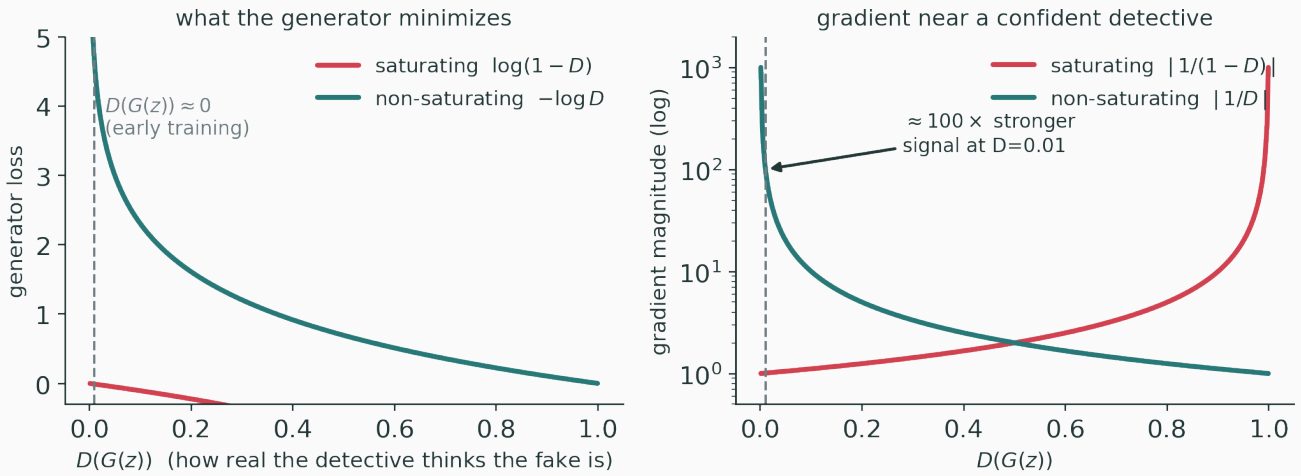
$$L_G = -\mathbb{E}_{z\sim p(z)} \log D_{\psi\left(G_{\theta(z)}\right)}$$

Non-saturating generator loss

D DERIVATION

Keep the same goal — make fakes look real — but flip to a loss with a healthy gradient:

$$L_G = -\mathbb{E}_{z\sim p(z)} \log D_{\psi}\big(G_{\theta(z)}\big)$$



Worked scalar example

D DERIVATION

Take a confident detective: $D(G(z)) = 0.01$.

Worked scalar example

D DERIVATION

Take a confident detective: $D(G(z)) = 0.01$.

Saturating $\log(1 - 0.01) = \log 0.99 \approx -0.01$.
Gradient magnitude $1/(1 - D) \approx 1.01$ — nearly flat.

Non-saturating $-\log 0.01 \approx 4.61$.
Gradient magnitude $1/D = 100$ — a strong push.

Worked scalar example

D DERIVATION

Take a confident detective: $D(G(z)) = 0.01$.

Saturating $\log(1 - 0.01) = \log 0.99 \approx -0.01$.
Gradient magnitude $1/(1 - D) \approx 1.01$ — nearly flat.

Non-saturating $-\log 0.01 \approx 4.61$.
Gradient magnitude $1/D = 100$ — a strong push.

$\approx 100 \times$ **stronger gradient at** $D = 0.01$ **— same direction, far more useful signal**

What equilibrium would mean

D DERIVATION

If the generated and real distributions matched **perfectly**, no detective could beat a coin flip:

What equilibrium would mean

D DERIVATION

If the generated and real distributions matched **perfectly**, no detective could beat a coin flip:

$$p_G = p_{\text{data}} \implies D(x) = \frac{1}{2} \quad \text{everywhere}$$

What equilibrium would mean

D DERIVATION

If the generated and real distributions matched **perfectly**, no detective could beat a coin flip:

$$p_G = p_{\text{data}} \implies D(x) = \frac{1}{2} \quad \text{everywhere}$$

the detective is reduced to guessing — the strongest possible evidence that the fakes match the data

Important correction

Q QUESTION

A common misreading of $D = \frac{1}{2}$:

Important correction

Q QUESTION

A common misreading of $D = \frac{1}{2}$:

$D(x) = \frac{1}{2}$ at equilibrium does **not** mean every generated image is average or blurry. It means the **two distributions are indistinguishable** — the detective cannot separate them, not that samples are bland.

Important correction

Q QUESTION

A common misreading of $D = \frac{1}{2}$:

$D(x) = \frac{1}{2}$ at equilibrium does **not** mean every generated image is average or blurry. It means the **two distributions are indistinguishable** — the detective cannot separate them, not that samples are bland.

a statement about **distributions**, not about any single image

Latent-space interpolation

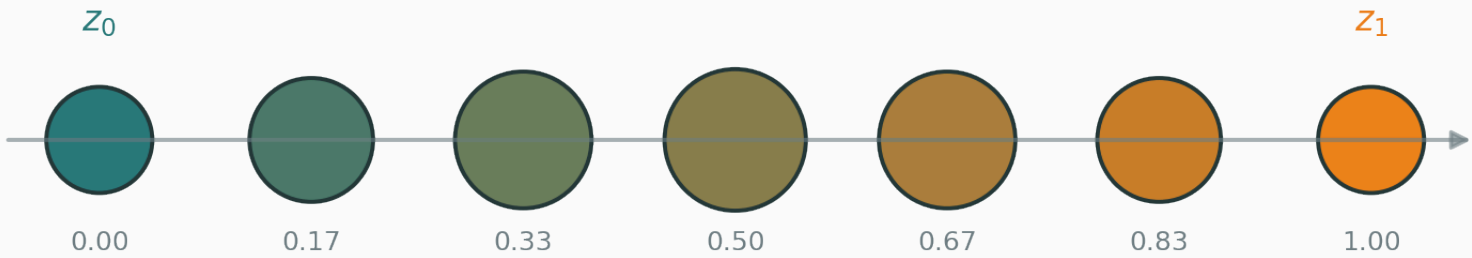
V VISUAL

Sample two codes, interpolate, and generate along the path:

Latent-space interpolation

V VISUAL

Sample two codes, interpolate, and generate along the path:

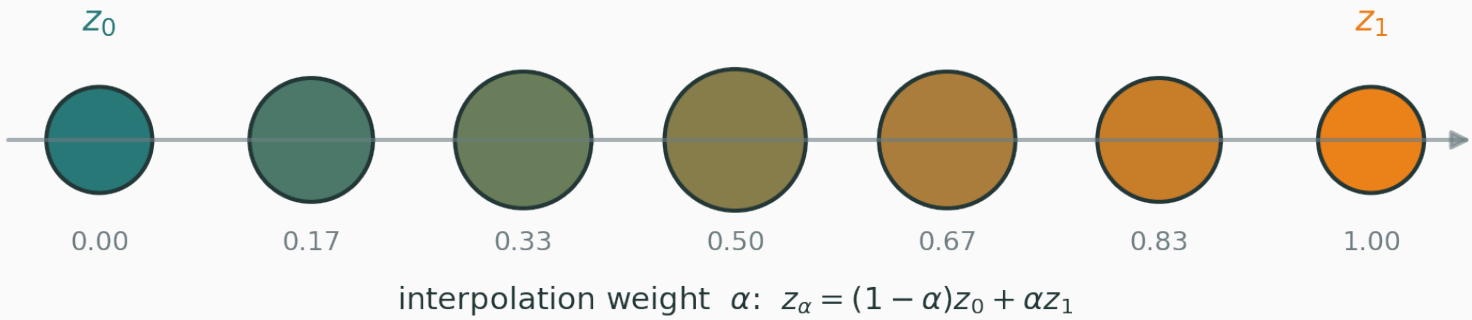


interpolation weight α: $z_\alpha = (1-\alpha)z_0 + \alpha z_1$

Latent-space interpolation

V VISUAL

Sample two codes, interpolate, and generate along the path:



smooth semantic change is evidence that G has **organized** the noise space into a usable map

Why GANs can be hard to train

The moving-target problem

D DERIVATION

Unlike ordinary supervised learning, the loss surface itself **keeps moving**:

The moving-target problem

D DERIVATION

Unlike ordinary supervised learning, the loss surface itself **keeps moving**:

Supervised — the target is **fixed**; you descend a stationary loss.

Adversarial — the opponent **learns too**; the objective shifts under you every step.

The moving-target problem

D DERIVATION

Unlike ordinary supervised learning, the loss surface itself **keeps moving**:

Supervised — the target is **fixed**; you descend a stationary loss.

Adversarial — the opponent **learns too**; the objective shifts under you every step.

game optimization $\neq$ minimizing one fixed function

Failure mode: discriminator too strong

D DERIVATION

If the detective becomes **perfect**, it separates real from fake with total confidence:

Failure mode: discriminator too strong

D DERIVATION

If the detective becomes **perfect**, it separates real from fake with total confidence:

$$D(x) \approx 1 \text{ on real}, \quad D(G(z)) \approx 0 \text{ on fake}$$

Failure mode: discriminator too strong

D DERIVATION

If the detective becomes **perfect**, it separates real from fake with total confidence:

$$D(x) \approx 1 \text{ on real}, \quad D(G(z)) \approx 0 \text{ on fake}$$

A perfect discriminator leaves the generator with **flat, uninformative** gradients — the game stalls because the critic gives no direction.

Failure mode: mode collapse

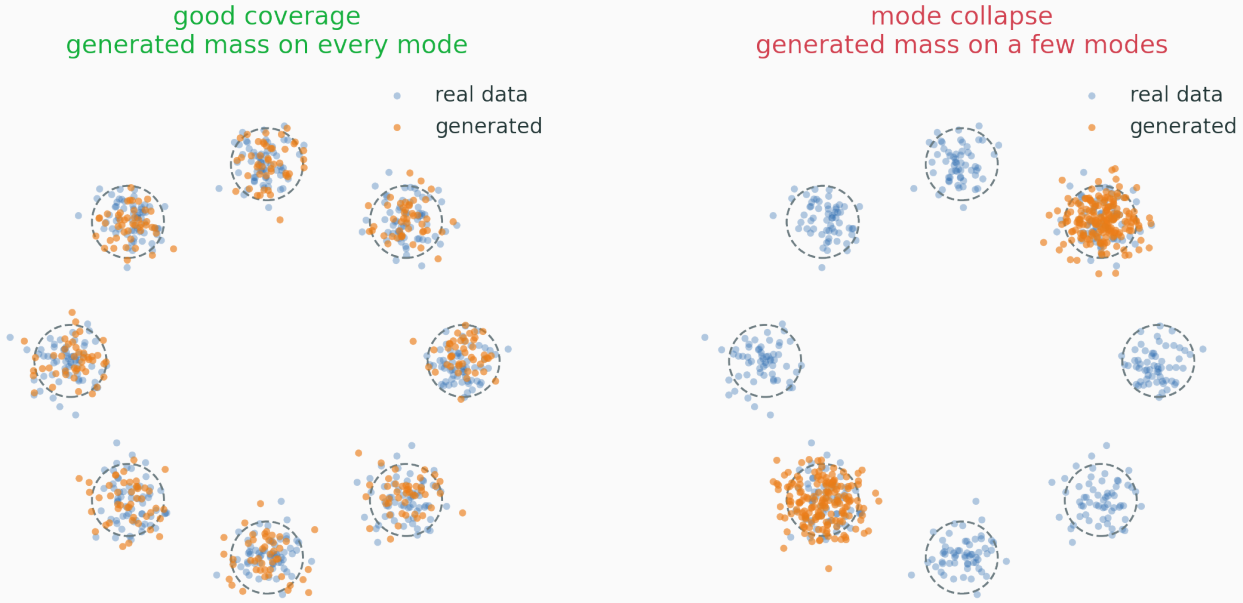
V VISUAL

The generator finds a few outputs that fool the **current** detective, and repeats them:

Failure mode: mode collapse

V VISUAL

The generator finds a few outputs that fool the **current** detective, and repeats them:



Coverage versus realism

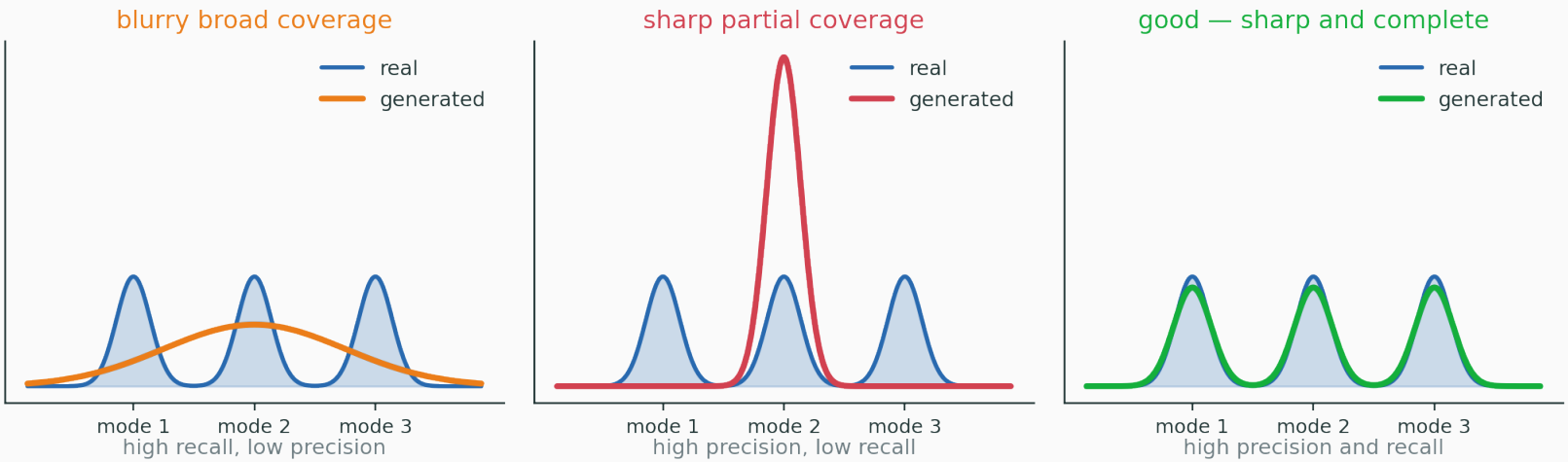
V VISUAL

Sharpness and coverage are **different** axes of quality:

Coverage versus realism

V VISUAL

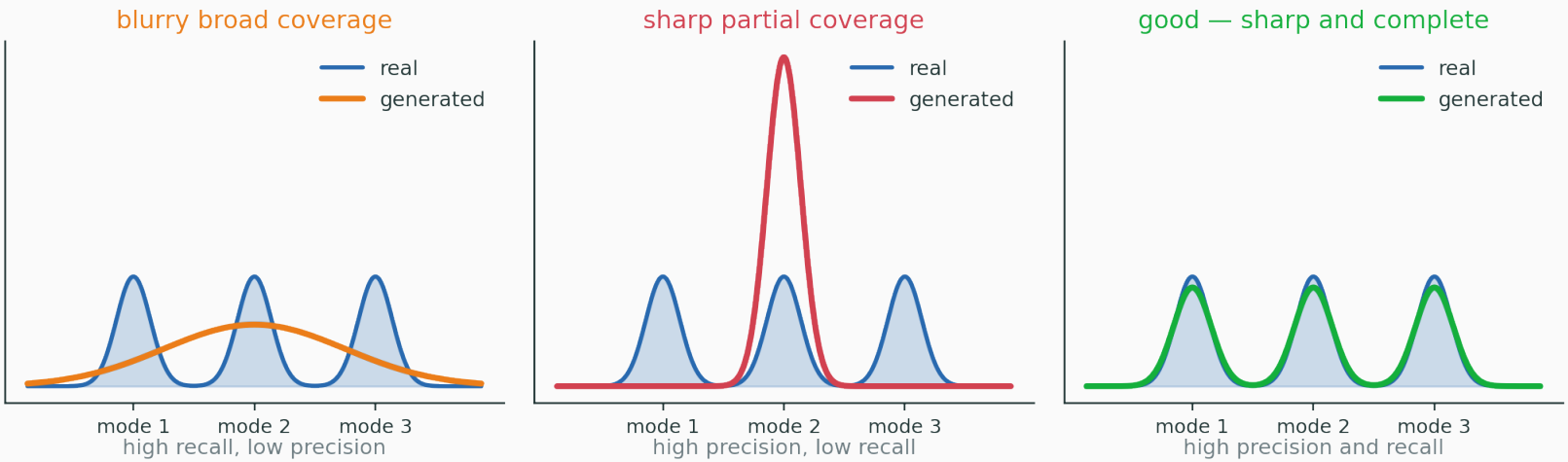
Sharpness and coverage are **different** axes of quality:



Coverage versus realism

V VISUAL

Sharpness and coverage are **different** axes of quality:



broad-but-blurry, sharp-but-partial, and sharp-and-complete are three genuinely distinct behaviours

Failure mode: oscillation

V VISUAL

The two players can **chase** each other around rather than converge:

Failure mode: oscillation

V VISUAL

The two players can **chase** each other around rather than converge:

Failure mode: oscillation

V VISUAL

The two players can **chase** each other around rather than converge:

training curves alone can mislead — a lower D loss is not a better generator

Practical stabilizers

D DERIVATION

A grab-bag of tricks that make the game better-behaved:

Practical stabilizers

D DERIVATION

A grab-bag of tricks that make the game better-behaved:

- **architecture** — convolutional up/down-sampling (DCGAN);
- **balance** — tuned update schedules and regularization;
- **better critics** — distance-like objectives (Wasserstein intuition);
- **monitor samples** — inspect generated grids, not only losses.

Practical stabilizers

D DERIVATION

A grab-bag of tricks that make the game better-behaved:

- **architecture** — convolutional up/down-sampling (DCGAN);
- **balance** — tuned update schedules and regularization;
- **better critics** — distance-like objectives (Wasserstein intuition);
- **monitor samples** — inspect generated grids, not only losses.

none is a silver bullet — GAN training is a craft as much as an algorithm

Wasserstein intuition, at a glance

D DERIVATION

A binary detective saturates; a **critic** that scores “how real” gives a smoother signal:

Wasserstein intuition, at a glance

D DERIVATION

A binary detective saturates; a **critic** that scores “how real” gives a smoother signal:

WGAN replaces the saturating classifier with a critic and measures an **earth-mover** distance between distributions. Read it as a response to **unstable geometry**, not a fourth objective to memorize.

Wasserstein intuition, at a glance

D DERIVATION

A binary detective saturates; a **critic** that scores “how real” gives a smoother signal:

WGAN replaces the saturating classifier with a critic and measures an **earth-mover** distance between distributions. Read it as a response to **unstable geometry**, not a fourth objective to memorize.

we return to the earth-mover picture in Part V

Conditional GANs

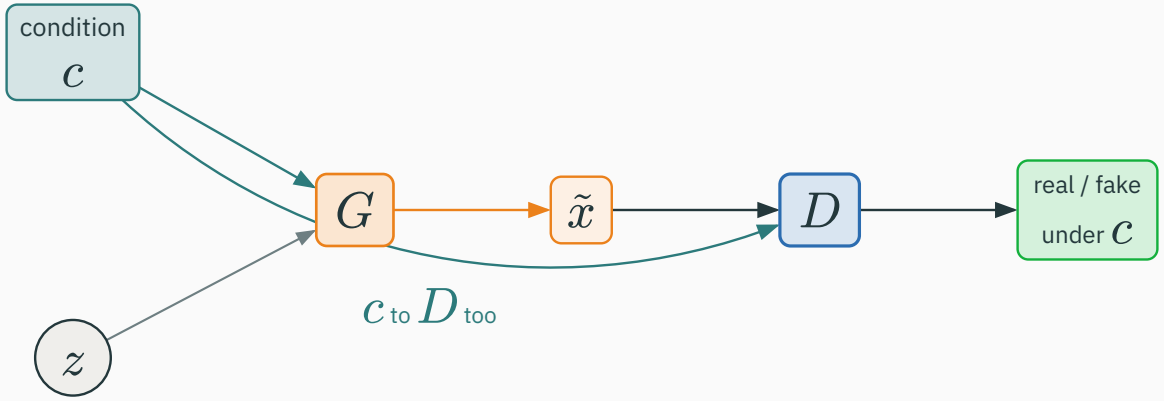
V VISUAL

Feed a **condition** c so generation becomes controllable:

Conditional GANs

V VISUAL

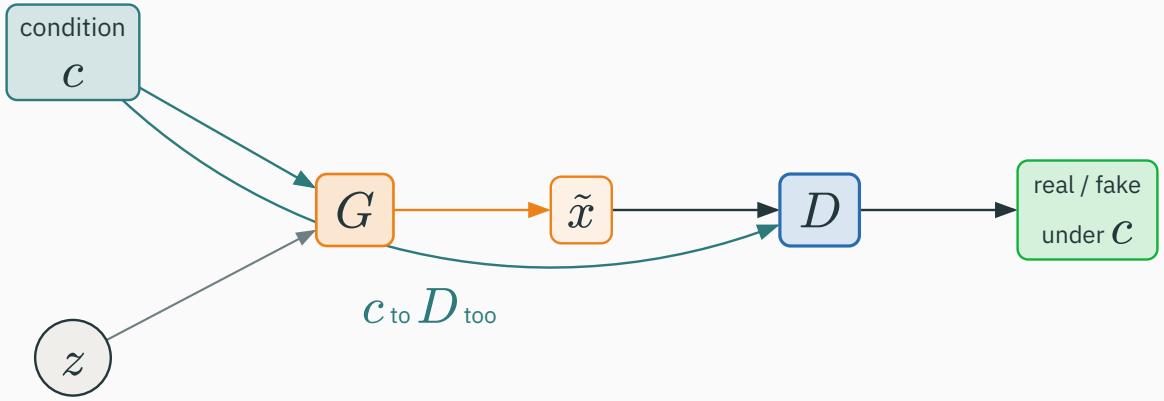
Feed a **condition** c so generation becomes controllable:



Conditional GANs

V VISUAL

Feed a **condition** c so generation becomes controllable:

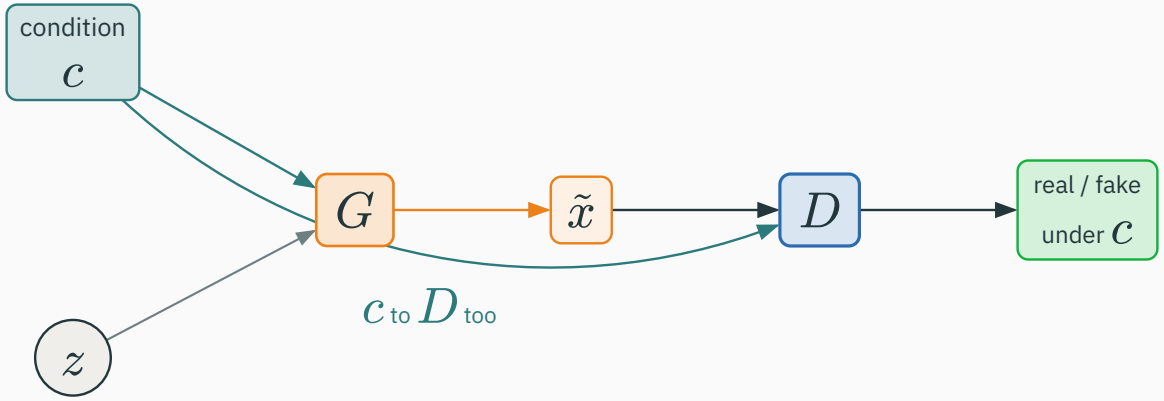


$$z, c \xrightarrow{G} \tilde{x}, \qquad D(x, c)$$

Conditional GANs

V VISUAL

Feed a **condition** c so generation becomes controllable:



$$z, c \xrightarrow{G} \tilde{x}, \qquad D(x, c)$$

c can be a class, a caption, a pose, or a segmentation map

GANs in the generative-model landscape

VAE versus GAN

V VISUAL

	VAE	GAN
training signal	likelihood bound (ELBO)	learned adversarial critic
inference encoder	yes	not required
typical trade-off	coverage, smoother samples	sharpness, instability / collapse
density estimate	explicit approximate model	implicit sampler

VAE versus GAN

V VISUAL

	VAE	GAN
training signal	likelihood bound (ELBO)	learned adversarial critic
inference encoder	yes	not required
typical trade-off	coverage, smoother samples	sharpness, instability / collapse
density estimate	explicit approximate model	implicit sampler

a probabilistic bound versus a learned game — two very different sources of gradient

Where GANs remain useful

D DERIVATION

GANs shine where structure is **paired** or **conditional** and a fast single pass matters:

Where GANs remain useful

D DERIVATION

GANs shine where structure is **paired** or **conditional** and a fast single pass matters:

- **image-to-image translation** (day $\rightarrow$ night, sketch $\rightarrow$ photo);
- **super-resolution** and restoration;
- **domain adaptation** with an adversarial alignment loss.

Where GANs remain useful

D DERIVATION

GANs shine where structure is **paired** or **conditional** and a fast single pass matters:

- **image-to-image translation** (day $\rightarrow$ night, sketch $\rightarrow$ photo);
- **super-resolution** and restoration;
- **domain adaptation** with an adversarial alignment loss.

strong conditional signal + one-shot sampling = a natural GAN fit

Why diffusion displaced many GAN uses

D DERIVATION

Diffusion (next lecture) trades one adversarial pass for **many** stable denoising steps:

Why diffusion displaced many GAN uses

D DERIVATION

Diffusion (next lecture) trades one adversarial pass for **many** stable denoising steps:

- **coverage** — a stationary regression target avoids mode collapse;
- **stability** — no moving opponent, so training is far more reliable;
- **cost** — the price is **many** sampling steps instead of one.

Why diffusion displaced many GAN uses

D DERIVATION

Diffusion (next lecture) trades one adversarial pass for **many** stable denoising steps:

- **coverage** — a stationary regression target avoids mode collapse;
- **stability** — no moving opponent, so training is far more reliable;
- **cost** — the price is **many** sampling steps instead of one.

better coverage and stability, at a sampling-time cost

Quick conceptual check

Q QUESTION

“If samples look **sharp** but all resemble the **same** face, which failure mode is this?”

Quick conceptual check

Q QUESTION

“If samples look **sharp** but all resemble the **same** face, which failure mode is this?”

and — would a discriminator loss **alone** have diagnosed it?

Quick conceptual check

Q QUESTION

"If samples look **sharp** but all resemble the **same** face, which failure mode is this?"

and — would a discriminator loss **alone** have diagnosed it?

mode collapse — and no, sharp fakes can keep D loss looking healthy

Closing line of the first pass

D DERIVATION

One sentence to carry forward:

Closing line of the first pass

D DERIVATION

One sentence to carry forward:

GANs learn by fooling a critic; diffusion learns by reversing noise.

Closing line of the first pass

D DERIVATION

One sentence to carry forward:

GANs learn by fooling a critic; diffusion learns by reversing noise.

Part V now derives the game’s optimum, the losses per minibatch, and the failure modes in detail

Detailed adversarial training

Binary cross-entropy from labels

D DERIVATION

For real data label $y = 1$ and generated data label $y = 0$:

Binary cross-entropy from labels

D DERIVATION

For real data label $y = 1$ and generated data label $y = 0$:

$$L_D = -[\, y \log D(x) + (1 - y) \log(1 - D(x)) \,]$$

Binary cross-entropy from labels

D DERIVATION

For real data label $y = 1$ and generated data label $y = 0$:

$$L_D = -[\, y \log D(x) + (1 - y) \log(1 - D(x)) \,]$$

Concretely, if $D(x_{\text{real}}) = 0.9$ and $D(G(z)) = 0.2$:

$$L_D = -(\log 0.9 + \log 0.8) \approx -(-0.105 - 0.223) = \mathbf{0.329}$$

Binary cross-entropy from labels

D DERIVATION

For real data label $y = 1$ and generated data label $y = 0$:

$$L_D = -[\, y \log D(x) + (1 - y) \log(1 - D(x)) \,]$$

Concretely, if $D(x_{\text{real}}) = 0.9$ and $D(G(z)) = 0.2$:

$$L_D = -(\log 0.9 + \log 0.8) \approx -(-0.105 - 0.223) = \mathbf{0.329}$$

ordinary classification — the twist is that the **negative class** moves as G changes

A minibatch discriminator loss

V VISUAL

Average the BCE over a batch of B real images and B fakes:

A minibatch discriminator loss

V VISUAL

Average the BCE over a batch of B real images and B fakes:

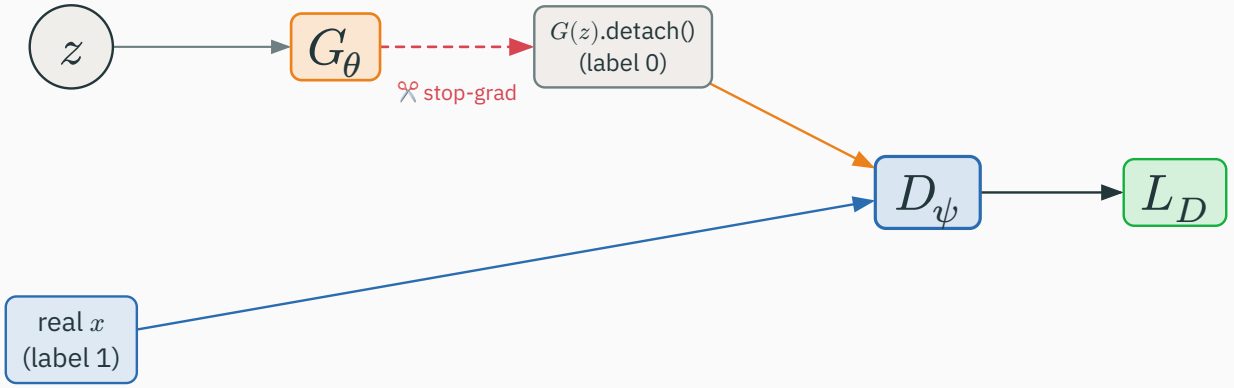
$$L_D = -\frac{1}{B}\sum_{i=1}^{B}\log D(x_i) - \frac{1}{B}\sum_{i=1}^{B}\log(1 - D(G(z_i)))$$

A minibatch discriminator loss

V VISUAL

Average the BCE over a batch of B real images and B fakes:

$$L_D = -\frac{1}{B}\sum_{i=1}^{B} \log D(x_i) - \frac{1}{B}\sum_{i=1}^{B} \log(1 - D(G(z_i)))$$

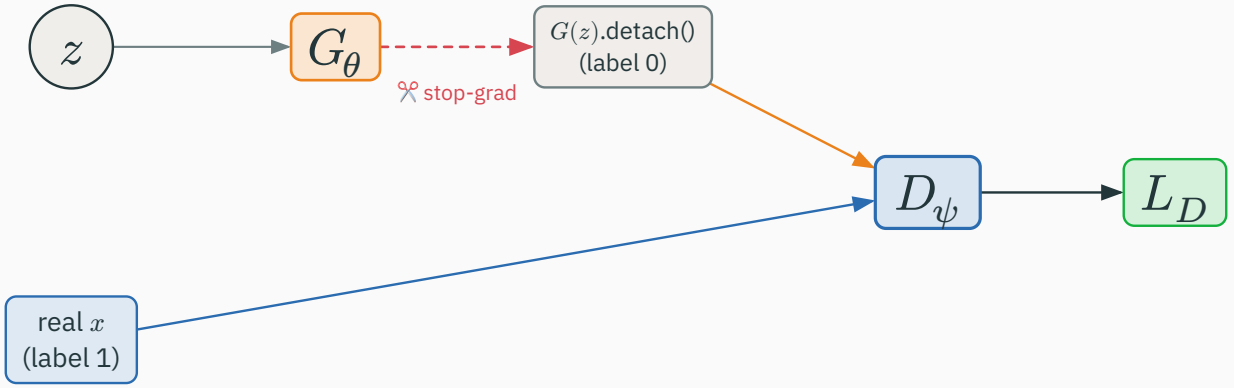


A minibatch discriminator loss

V VISUAL

Average the BCE over a batch of B real images and B fakes:

$$L_D = -\frac{1}{B}\sum_{i=1}^{B} \log D(x_i) - \frac{1}{B}\sum_{i=1}^{B} \log(1 - D(G(z_i)))$$



when updating D, the fakes are **detached** — no gradient leaks back into G

A minibatch generator loss

D DERIVATION

Now update G with the non-saturating loss, discriminator frozen:

A minibatch generator loss

D DERIVATION

Now update G with the non-saturating loss, discriminator frozen:

$$L_G = -\frac{1}{B}\sum_{i=1}^{B} \log D(G(z_i))$$

A minibatch generator loss

D DERIVATION

Now update G with the non-saturating loss, discriminator frozen:

$$L_G = -\frac{1}{B}\sum_{i=1}^{B} \log D(G(z_i))$$

The discriminator's weights are **frozen**, but its computation **remains in the graph** — that is how G receives a direction for changing its pixels.

The generator is not trained on “fake” labels

D DERIVATION

A subtle but crucial point about what G optimizes:

The generator is not trained on “fake” labels

D DERIVATION

A subtle but crucial point about what G optimizes:

```
update D:  real       -> 1 ,   generated -> 0
update G:  generated  -> make D output 1
```

The generator is not trained on “fake” labels

D DERIVATION

A subtle but crucial point about what G optimizes:

```
update D:  real       -> 1 ,   generated -> 0
update G:  generated  -> make D output 1
```

G does **not** become a classifier — it changes its images so D’s learned features regard them as real

The discriminator’s boundary moves

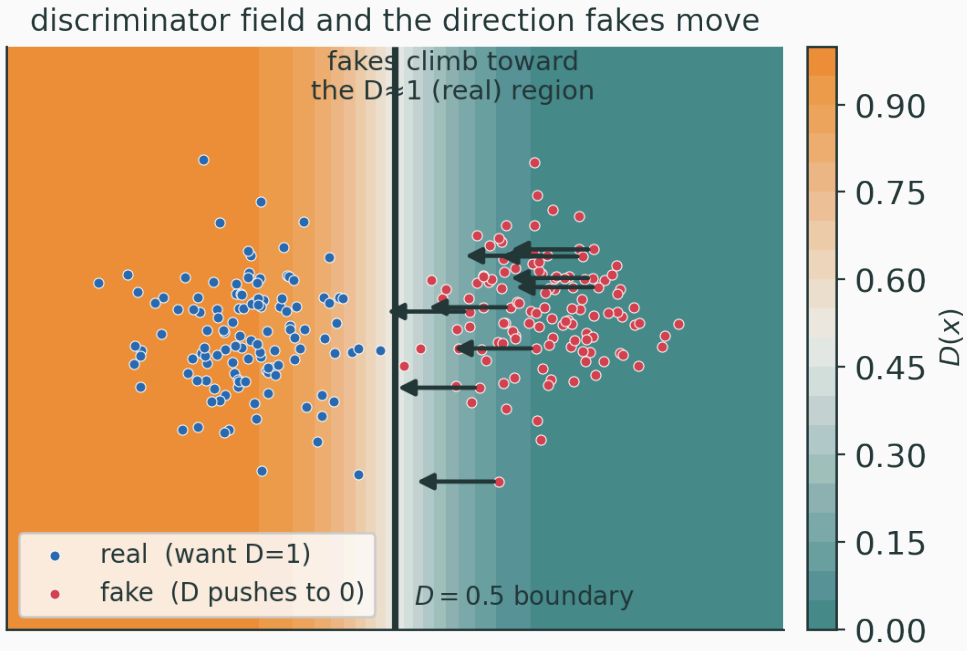
V VISUAL

Two real-ish clusters and a fake cluster: after each update the picture shifts.

The discriminator’s boundary moves

V VISUAL

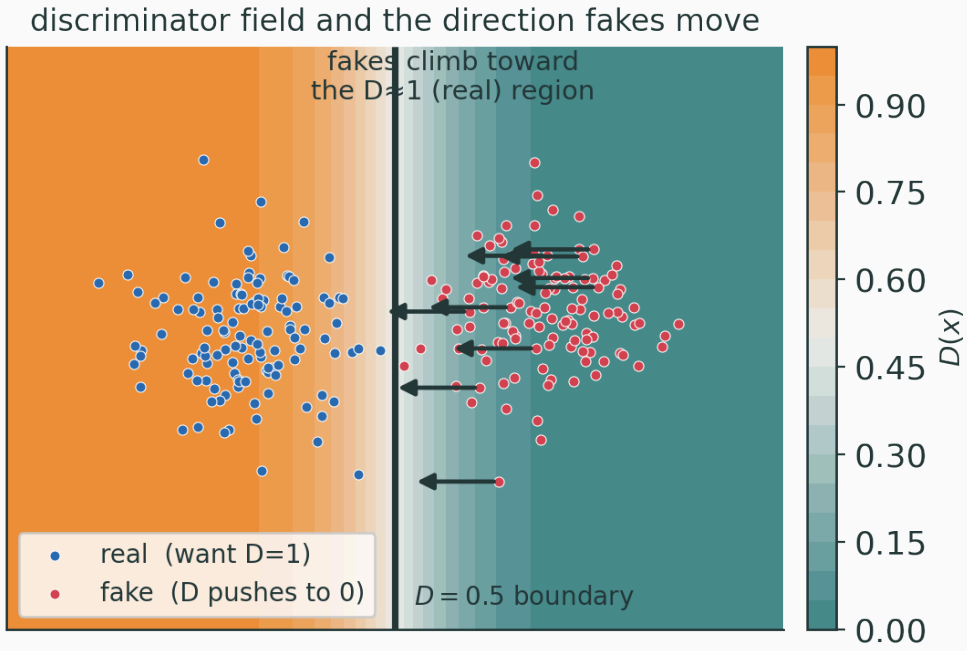
Two real-ish clusters and a fake cluster: after each update the picture shifts.



The discriminator’s boundary moves

V VISUAL

Two real-ish clusters and a fake cluster: after each update the picture shifts.



this moving-boundary picture is worth more than a formal convergence proof on a first pass

Ideal equilibrium, carefully stated

D DERIVATION

For a **fixed** generator, the optimal discriminator has a closed form:

Ideal equilibrium, carefully stated

D DERIVATION

For a **fixed** generator, the optimal discriminator has a closed form:

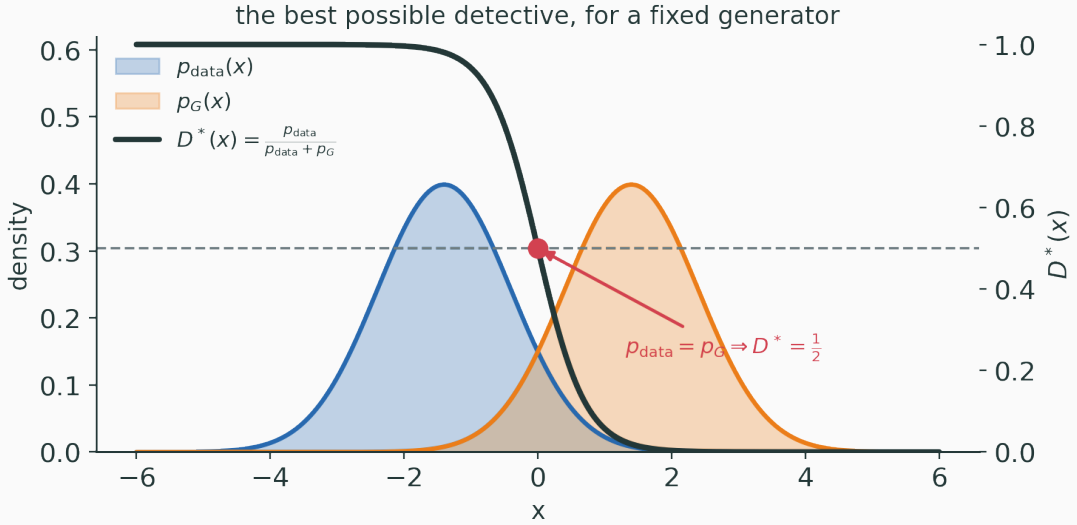
$$D^*(x) = \frac{p_{\text{data}(x)}}{p_{\text{data}(x)} + p_{G(x)}}$$

Ideal equilibrium, carefully stated

D DERIVATION

For a **fixed** generator, the optimal discriminator has a closed form:

$$D^*(x) = \frac{p_{\text{data}(x)}}{p_{\text{data}(x)} + p_{G(x)}}$$



Divergence intuition

D DERIVATION

Substituting D^* back turns the game value into a **divergence** between distributions:

Divergence intuition

D DERIVATION

Substituting D^* back turns the game value into a **divergence** between distributions:

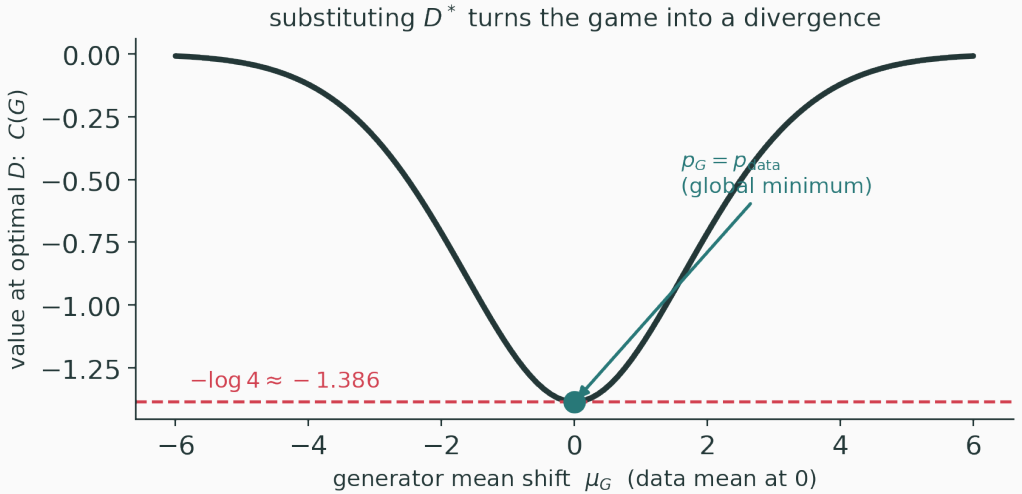
$$C(G) = -\log 4 + 2 \cdot \mathrm{JSD}(p_{\text{data}} \,\|\, p_G)$$

Divergence intuition

D DERIVATION

Substituting D^* back turns the game value into a **divergence** between distributions:

$$C(G) = -\log 4 + 2 \cdot \mathrm{JSD}(p_{\text{data}} \,\|\, p_G)$$



The classroom message

Q QUESTION

Strip away the algebra and one idea remains:

The classroom message

Q QUESTION

Strip away the algebra and one idea remains:

GAN training tries to match distributions — not to pair each fake with one real target.

The classroom message

Q QUESTION

Strip away the algebra and one idea remains:

GAN training tries to match distributions — not to pair each fake with one real target.

the minimum $-\log 4 \approx -1.386$ is reached exactly when $p_G = p_{\text{data}}$, where JSD $= 0$

Why early generator gradients can fail

D DERIVATION

When the detective assigns $D(G(z)) \approx 0$ everywhere:

Why early generator gradients can fail

D DERIVATION

When the detective assigns $D(G(z)) \approx 0$ everywhere:

- the original minimax generator loss **flattens** — a vanishing gradient;

Why early generator gradients can fail

D DERIVATION

When the detective assigns $D(G(z)) \approx 0$ everywhere:

- the original minimax generator loss **flattens** — a vanishing gradient;
- the non-saturating loss changes the **gradient behaviour** while keeping the **same desired direction**.

Why early generator gradients can fail

D DERIVATION

When the detective assigns $D(G(z)) \approx 0$ everywhere:

- the original minimax generator loss **flattens** — a vanishing gradient;
- the non-saturating loss changes the **gradient behaviour** while keeping the **same desired direction**.

same target, different curvature — the fix is about gradient magnitude, not about goals

Alternating optimization pseudocode

D DERIVATION

The whole training loop in five lines:

Alternating optimization pseudocode

D DERIVATION

The whole training loop in five lines:

```
repeat:
    x <- real minibatch ;  z <- random noise
    update discriminator using x and G(z).detach()
    z <- fresh random noise
    update generator using  -log D(G(z))
```

Alternating optimization pseudocode

D DERIVATION

The whole training loop in five lines:

```
repeat:
    x <- real minibatch ;  z <- random noise
    update discriminator using x and G(z).detach()
    z <- fresh random noise
    update generator using  -log D(G(z))
```

the two updates use **different** colours of parameter — D frozen for one, G frozen for the other

Update ratio is a design decision

D DERIVATION

One D step per G step is **common**, not a law:

Update ratio is a design decision

D DERIVATION

One D step per G step is **common**, not a law:

- some critic objectives take **several** discriminator updates per generator update, to keep the signal informative;

Update ratio is a design decision

D DERIVATION

One D step per G step is **common**, not a law:

- some critic objectives take **several** discriminator updates per generator update, to keep the signal informative;
- others balance learning rates or add regularization instead.

Update ratio is a design decision

D DERIVATION

One D step per G step is **common**, not a law:

- some critic objectives take **several** discriminator updates per generator update, to keep the signal informative;
- others balance learning rates or add regularization instead.

a first example of why **game** optimization is more delicate than minimizing one fixed loss

Conditioning enters both players

D DERIVATION

Supply the condition c to the generator **and** the discriminator:

Conditioning enters both players

D DERIVATION

Supply the condition c to the generator **and** the discriminator:

$$G(z, c) \to \tilde{x}, \qquad D(x, c) \to \text{real} \;/\; \text{fake}$$

Conditioning enters both players

D DERIVATION

Supply the condition c to the generator **and** the discriminator:

$$G(z, c) \to \tilde{x}, \qquad D(x, c) \to \text{real} \;/\; \text{fake}$$

- for **digit generation**, c is a class label;
- for **image translation**, c is an input image, pose, or segmentation map.

Conditioning enters both players

D DERIVATION

Supply the condition c to the generator **and** the discriminator:

$$G(z, c) \to \tilde{x}, \qquad D(x, c) \to \text{real} \;/\; \text{fake}$$

- for **digit generation**, c is a class label;
- for **image translation**, c is an input image, pose, or segmentation map.

D must now judge realism **and** agreement with c — a fake-but-mismatched sample is still rejected

Conditional generation versus selection

V VISUAL

Mechanism	**What it does**
latent z	chooses **one** diverse sample
condition c	specifies the **desired content**
discriminator / critic	judges **realism** under c

Conditional generation versus selection

V VISUAL

Mechanism	**What it does**
latent z	chooses **one** diverse sample
condition c	specifies the **desired content**
discriminator / critic	judges **realism** under c

many different z values can produce different **valid** samples for the **same** c

Architecture: DCGAN principles

D DERIVATION

The practical pattern that first made GANs train reliably on images:

Architecture: DCGAN principles

D DERIVATION

The practical pattern that first made GANs train reliably on images:

- **convolutional** up / down-sampling (no large dense hidden layers);
- careful **normalization** choices;
- **progressively** build spatial resolution.

Architecture: DCGAN principles

D DERIVATION

The practical pattern that first made GANs train reliably on images:

- **convolutional** up / down-sampling (no large dense hidden layers);
- careful **normalization** choices;
- **progressively** build spatial resolution.

treat DCGAN as an architectural **stabilization milestone**, not a paper to memorize

Generator geometry

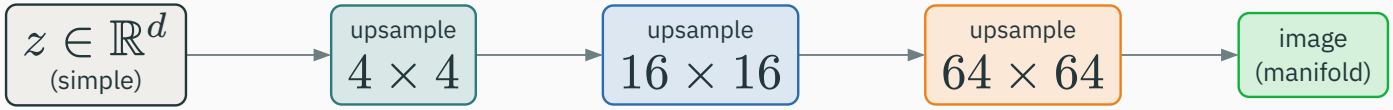
V VISUAL

The generator maps a **simple** connected latent onto a **complicated** data manifold:

Generator geometry

V VISUAL

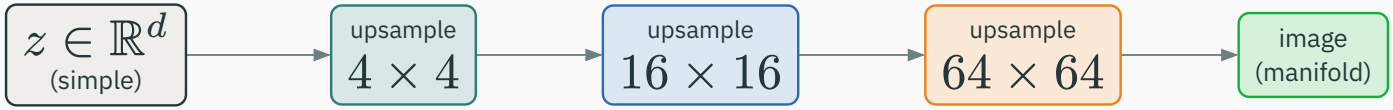
The generator maps a **simple** connected latent onto a **complicated** data manifold:



Generator geometry

V VISUAL

The generator maps a **simple** connected latent onto a **complicated** data manifold:



interpolation between two codes traces a **path** along this learned map

Mode collapse, expanded

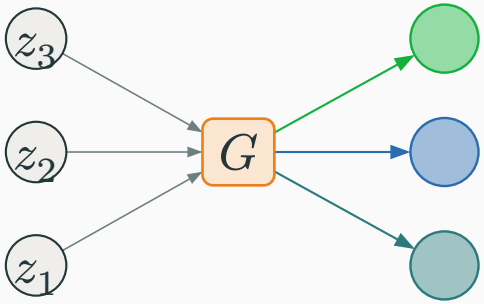
V VISUAL

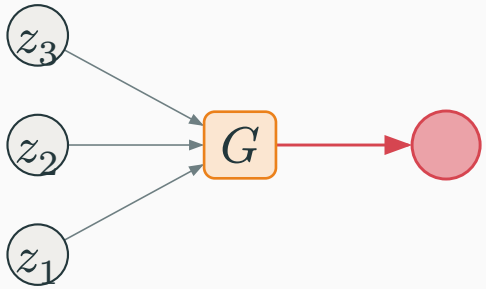
Send several z values through G and look at the outputs:

Mode collapse, expanded

V VISUAL

Send several z values through G and look at the outputs:





Mode collapse, expanded

V VISUAL

Send several z values through G and look at the outputs:

left: a **healthy** generator fans out to varied outputs — right: a **collapsed** one maps everything to a single convincing sample

Mode dropping is subtler

Q QUESTION

Collapse is the obvious failure; **dropping** is the quiet one:

Mode dropping is subtler

Q QUESTION

Collapse is the obvious failure; **dropping** is the quiet one:

A generator may cover **several** modes yet **omit rare** regions. The sample grid looks impressive while quietly failing fairness, safety, or scientific-use requirements.

Mode dropping is subtler

Q QUESTION

Collapse is the obvious failure; **dropping** is the quiet one:

A generator may cover **several** modes yet **omit rare** regions. The sample grid looks impressive while quietly failing fairness, safety, or scientific-use requirements.

quality and coverage must be evaluated **separately**

Loss curves are not enough

D DERIVATION

GAN losses oscillate and change scale as the opponent changes. Always also inspect:

Loss curves are not enough

D DERIVATION

GAN losses oscillate and change scale as the opponent changes. Always also inspect:

- **fixed-noise sample grids** over the course of training;
- **diversity** of the generated set;
- **task-specific** downstream measures.

Loss curves are not enough

D DERIVATION

GAN losses oscillate and change scale as the opponent changes. Always also inspect:

- **fixed-noise sample grids** over the course of training;
- **diversity** of the generated set;
- **task-specific** downstream measures.

a lower discriminator loss does **not** monotonically mean better samples

Precision and recall for generation

D DERIVATION

Split “quality” into two measurable axes:

Precision and recall for generation

D DERIVATION

Split “quality” into two measurable axes:

Precision — do generated samples lie **near** the data manifold? (fidelity)

Recall — does the generator **cover** the manifold? (diversity)

Precision and recall for generation

D DERIVATION

Split “quality” into two measurable axes:

Precision — do generated samples lie **near** the data manifold? (fidelity)

Recall — does the generator **cover** the manifold? (diversity)

sharp, collapsed outputs can score **high precision** and **poor recall**

FID, one careful slide

D DERIVATION

Fréchet-style metrics compare **feature statistics** of real and generated images:

FID, one careful slide

D DERIVATION

Fréchet-style metrics compare **feature statistics** of real and generated images:

$$\text{one scalar metric} \quad \neq \quad \text{complete judgment of quality, diversity, bias, memorization}$$

FID, one careful slide

D DERIVATION

Fréchet-style metrics compare **feature statistics** of real and generated images:

$$\text{one scalar metric} \quad \neq \quad \text{complete judgment of quality, diversity, bias, memorization}$$

FID is a **useful summary**, not a verdict. A single number hides mode dropping, bias, and near-copying — inspect samples too.

Wasserstein intuition

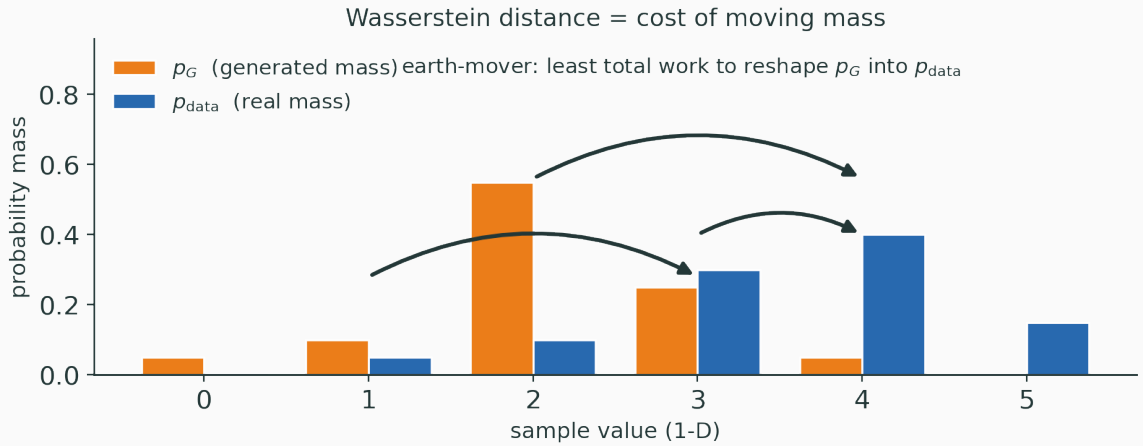
V VISUAL

If two distributions have **disjoint support**, a binary detective becomes overconfident. A distance-like critic measures **how far mass must move**:

Wasserstein intuition

V VISUAL

If two distributions have **disjoint support**, a binary detective becomes overconfident. A distance-like critic measures **how far mass must move**:



The Lipschitz constraint, at a glance

D DERIVATION

Wasserstein critics need **controlled sensitivity** — a bounded slope:

The Lipschitz constraint, at a glance

D DERIVATION

Wasserstein critics need **controlled sensitivity** — a bounded slope:

$$|f(x) - f(y)| \leq K\|x - y\|$$

The Lipschitz constraint, at a glance

D DERIVATION

Wasserstein critics need **controlled sensitivity** — a bounded slope:

$$|f(x) - f(y)| \leq K\|x - y\|$$

Weight clipping and **gradient penalties** are two ways to enforce this approximate 1-Lipschitz property. We name them, and stop short of deriving WGAN-GP.

Style-based generation, one placement slide

D DERIVATION

Where the latent is **injected** changes what it controls:

Style-based generation, one placement slide

D DERIVATION

Where the latent is **injected** changes what it controls:

- separate latent processing can steer **coarse-to-fine** attributes at **different layers**;
- pose and layout at coarse scales, colour and texture at fine scales.

Style-based generation, one placement slide

D DERIVATION

Where the latent is **injected** changes what it controls:

- separate latent processing can steer **coarse-to-fine** attributes at **different layers**;
- pose and layout at coarse scales, colour and texture at fine scales.

a clean example of **architecture shaping the meaning** of a latent — not a required implementation

Image-to-image GANs

V VISUAL

Condition on a **source image** and translate it:

Image-to-image GANs

Condition on a **source image** and translate it:

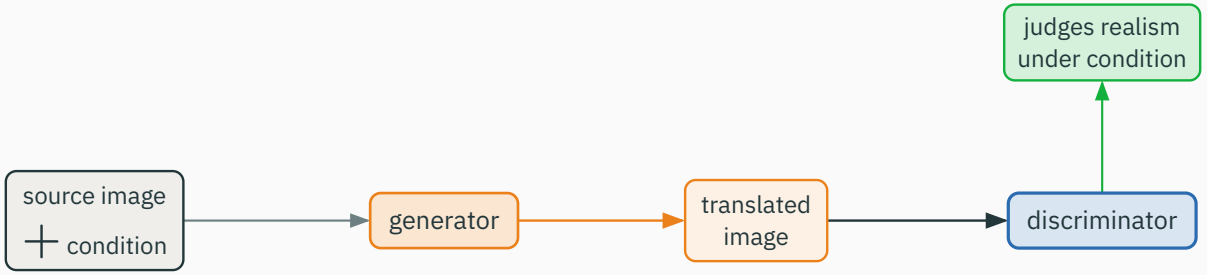
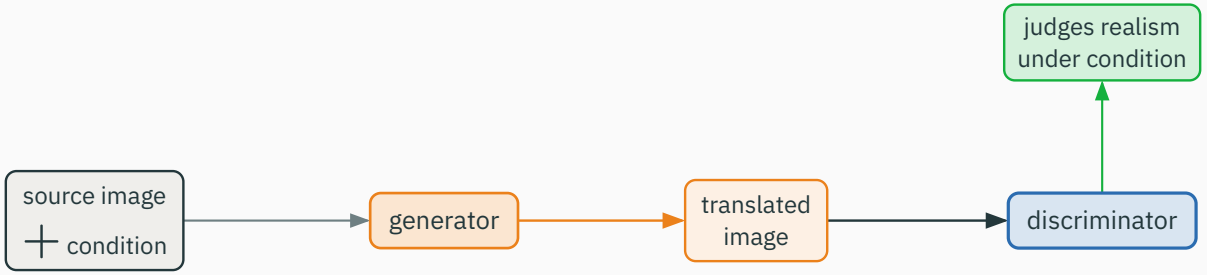


Image-to-image GANs

V VISUAL

Condition on a **source image** and translate it:



super-resolution · day-to-night · domain adaptation — contrast with diffusion-based editing later

Limited data and memorization

Q QUESTION

With small or narrow datasets, a generator can **memorize** or nearly copy examples:

Limited data and memorization

Q QUESTION

With small or narrow datasets, a generator can **memorize** or nearly copy examples:

A visually persuasive sample grid does **not** by itself establish generalization. Run **nearest-neighbour** checks against the training set.

Limited data and memorization

Q QUESTION

With small or narrow datasets, a generator can **memorize** or nearly copy examples:

A visually persuasive sample grid does **not** by itself establish generalization. Run **nearest-neighbour** checks against the training set.

memorization is a **data-governance** issue, not only a quality one

GANs and representations

D DERIVATION

The discriminator’s intermediate features often become **useful representations**:

GANs and representations

D DERIVATION

The discriminator’s intermediate features often become **useful representations**:

- separating real from fake requires **semantic** and **texture-sensitive** features;
- those features transfer to downstream tasks — echoing L17.

GANs and representations

D DERIVATION

The discriminator’s intermediate features often become **useful representations**:

- separating real from fake requires **semantic** and **texture-sensitive** features;
- those features transfer to downstream tasks — echoing L17.

many generative objectives **incidentally** learn representations — but their evaluation targets differ

VAE, GAN, diffusion: first comparison

V VISUAL

Question	VAE	GAN	Diffusion
what trains it?	likelihood bound	adversarial signal	known-noise regression
what is hard?	latent / recon trade-off	game stability	iterative sampling
what we emphasize	probabilistic latent	realism and coverage	denoising dynamics

VAE, GAN, diffusion: first comparison

V VISUAL

Question	VAE	GAN	Diffusion
what trains it?	likelihood bound	adversarial signal	known-noise regression
what is hard?	latent / recon trade-off	game stability	iterative sampling
what we emphasize	probabilistic latent	realism and coverage	denoising dynamics

three routes to the same goal — draw samples that look like the data

Classroom diagnosis

Q QUESTION

Match each **symptom** to a likely issue and one intervention category:

Classroom diagnosis

Q QUESTION

Match each **symptom** to a likely issue and one intervention category:

Symptom	**Likely issue**
fake samples repeat	mode collapse — diversify / rebalance
D accuracy near perfect	D too strong — weaken / critic loss
samples drift around training	oscillation — schedule / regularize
diverse but implausible	under-trained G — more capacity / steps

Classroom diagnosis

Q QUESTION

Match each **symptom** to a likely issue and one intervention category:

Symptom	**Likely issue**
fake samples repeat	mode collapse — diversify / rebalance
D accuracy near perfect	D too strong — weaken / critic loss
samples drift around training	oscillation — schedule / regularize
diverse but implausible	under-trained G — more capacity / steps

no symptom has a **unique** fix — name a category, not a magic cure

Common misconceptions

V VISUAL

Misconception	**Correction**
D is used at generation time	usually only $G(z)$ is needed after training
$D = \frac{1}{2}$ means a bad classifier	it is **ideal** only when distributions match
sharp samples imply good generation	mode collapse can be sharp and incomplete
GANs maximize likelihood	standard GANs are **implicit** adversarial models

Retrieval questions

Q QUESTION

Test yourself before the next lecture:

Retrieval questions

Q QUESTION

Test yourself before the next lecture:

1. Why does the generator need gradients **through** a frozen discriminator?

Retrieval questions

Q QUESTION

Test yourself before the next lecture:

1. Why does the generator need gradients **through** a frozen discriminator?
2. What exactly changes in the **non-saturating** loss?

Retrieval questions

Q QUESTION

Test yourself before the next lecture:

1. Why does the generator need gradients **through** a frozen discriminator?
2. What exactly changes in the **non-saturating** loss?
3. How does **mode collapse** differ from poor visual fidelity?

Retrieval questions

Q QUESTION

Test yourself before the next lecture:

1. Why does the generator need gradients **through** a frozen discriminator?
2. What exactly changes in the **non-saturating** loss?
3. How does **mode collapse** differ from poor visual fidelity?
4. Why do **game** losses resist a simple interpretation?

Student takeaways

Student takeaways

D DERIVATION

1. A GAN is a **two-player optimization problem**, not a generator plus a conventional label.

Student takeaways

D DERIVATION

1. A GAN is a **two-player optimization problem**, not a generator plus a conventional label.
2. The discriminator supplies a **learned realism signal**, and its gradients train the generator.

Student takeaways

D DERIVATION

1. A GAN is a **two-player optimization problem**, not a generator plus a conventional label.
2. The discriminator supplies a **learned realism signal**, and its gradients train the generator.
3. Sharp samples do **not** guarantee coverage — **mode collapse** is the central warning.

Student takeaways

D DERIVATION

1. A GAN is a **two-player optimization problem**, not a generator plus a conventional label.
2. The discriminator supplies a **learned realism signal**, and its gradients train the generator.
3. Sharp samples do **not** guarantee coverage — **mode collapse** is the central warning.
4. Diffusion solves a **different** training problem: denoising known corruptions, not fooling a moving critic.

A GAN learns a **critic** to tell fake from real; diffusion learns to **reverse a known noise process**.

Next: **Diffusion Models** — a simple corruption process turns generation into **supervised denoising**.