DDA4220 · Fall 2026

Lecture 18

Deep Learning and Applications

Conditional generation and flow matching

Zhen Liu

CUHK-Shenzhen

Lecture outline

01 Choosing the path

02 Flow matching

03 Paths, steps and reflow

04 Conditional generation

05 Classifier-free guidance

06 Generating large images

07 A text-to-image system and its uses

* denotes optional slides

Images from a sentence



Stable Diffusion 3: a 1024×1024 image from a text prompt

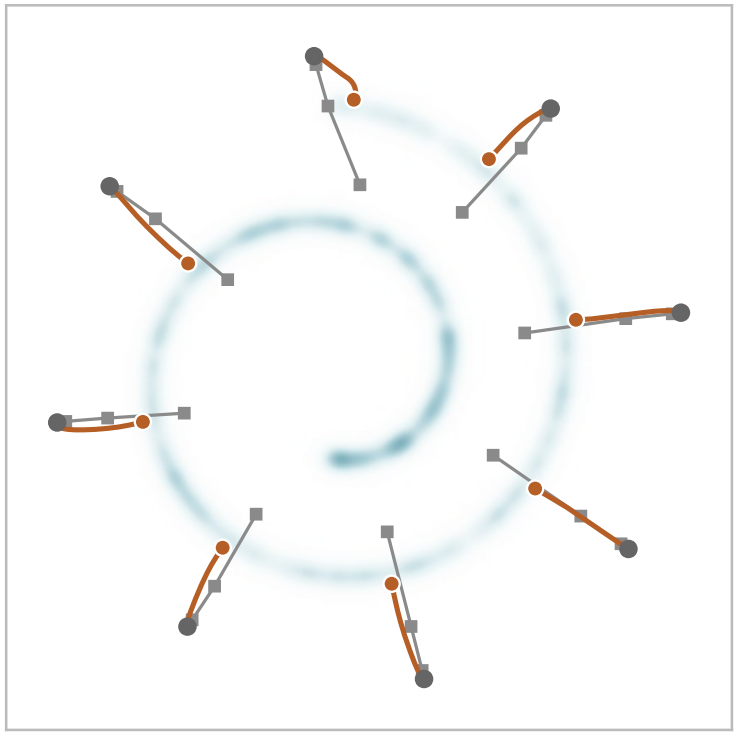
Today's three questions

1. An easy path from noise
2. Following the prompt
3. A million pixels per step

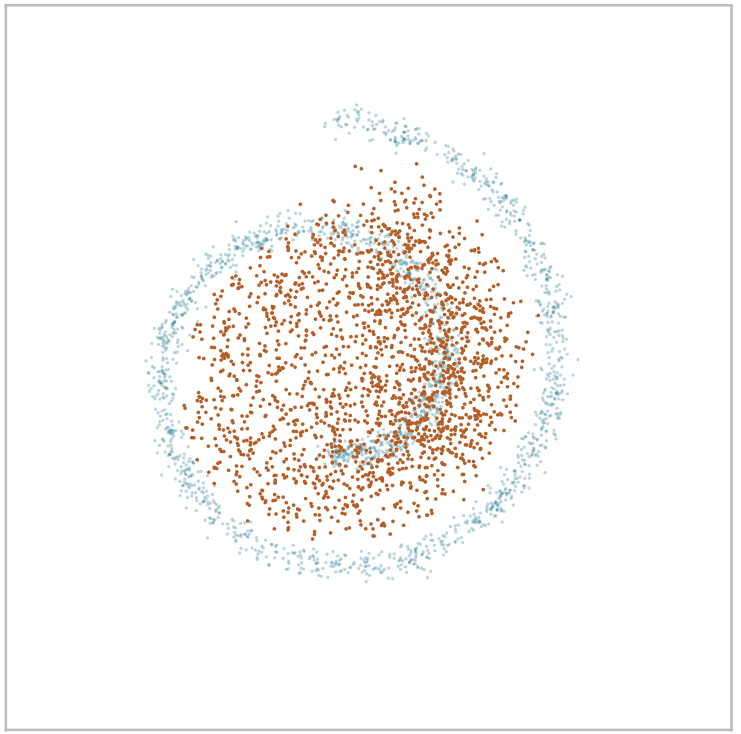
Source: Stable Diffusion 3 (2024)

L17: sampling follows a path from noise

1000 steps (orange), 4 steps (gray)



2000 samples, 4 DDIM steps



Exact noise predictions: no training error

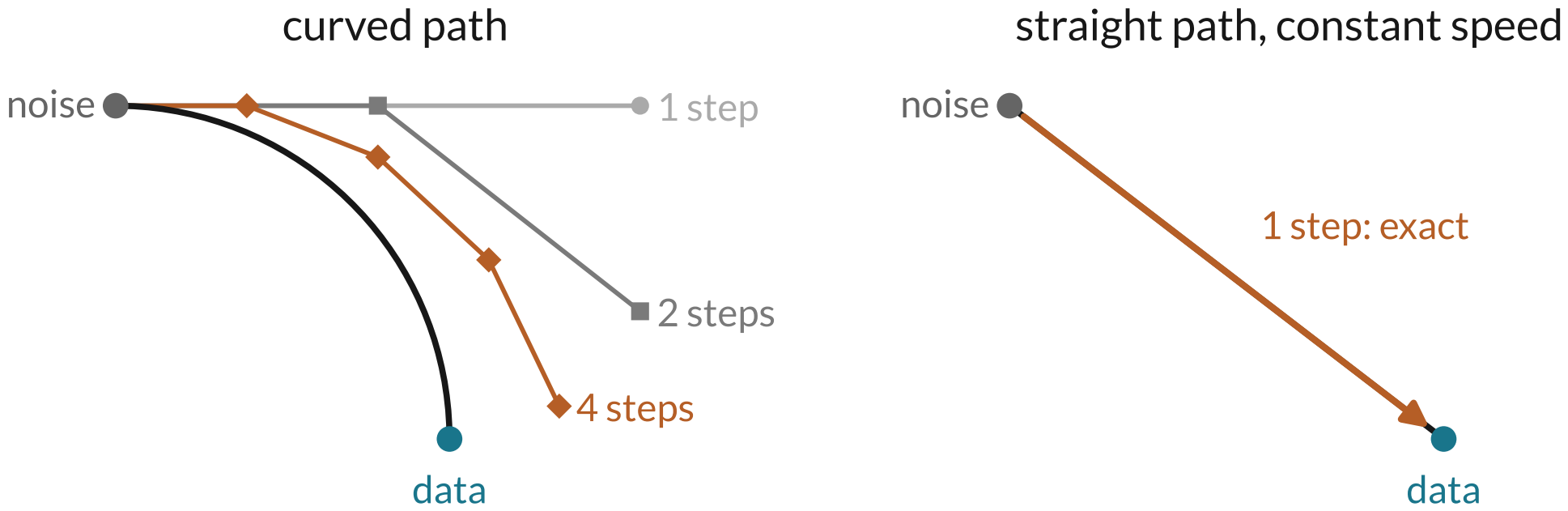
DDIM: one fixed path per noise sample

One network call per step

Gray: 4 big steps land between the arms

Few steps need an easy path

Big steps leave a curved path

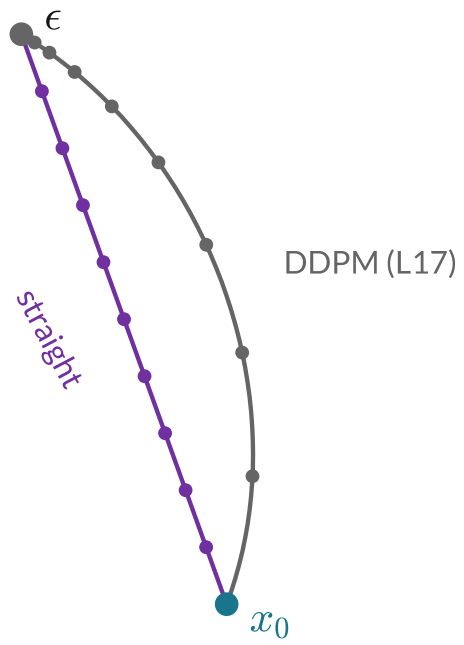


A step moves in the direction at its start; on a curve, the errors add up

Straight path, constant speed: one step is exact

The path L17 inherited

$$x_t = \underbrace{\sqrt{\bar{\alpha}_t}}_{\text{noise schedule}}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$$



One pair (x_0, ϵ); gray dots every 100 of 1000 steps

Curved and uneven: from the noise end, 400 steps cover 10% of the way

Set by the noise schedule, not chosen for sampling

Wanted: a path we choose ourselves

Simplest choice: straight, at constant speed (purple)

Lecture outline

01 Choosing the path

02 Flow matching

03 Paths, steps and reflow

04 Conditional generation

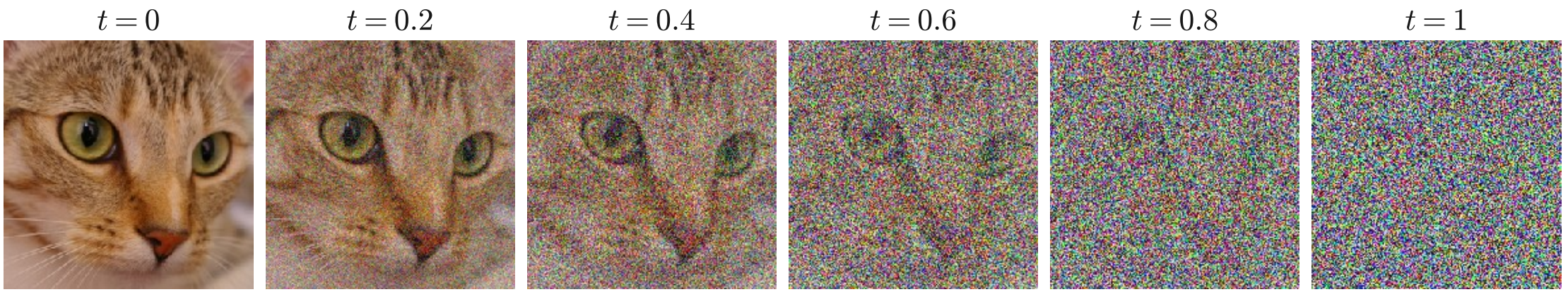
05 Classifier-free guidance

06 Generating large images

07 A text-to-image system and its uses

A straight path from an image to noise

$$x_t = \underbrace{(1-t)}_{\text{image weight}}\, x_0 + \underbrace{t}_{\text{noise weight}}\, \epsilon$$



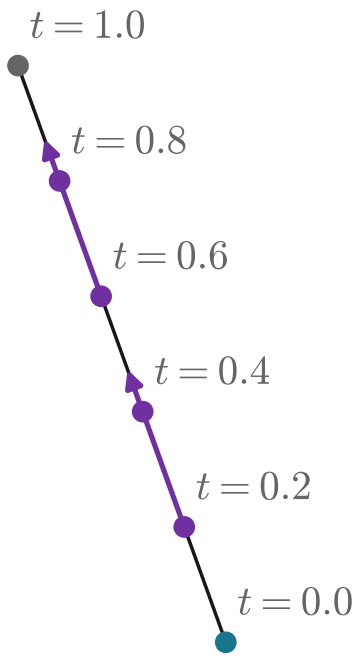
$t = 0$: the image x_0. $t = 1$: pure noise ϵ. In between: a straight blend

Here t is a number in $[0, 1]$, not a step count as in L17

Source: photo: scikit-image (CC0)

Its velocity is known

$$\frac{dx_t}{dt} = \underbrace{\epsilon - x_0}_{\text{the same at every t}}$$



Velocity: rate of change of x_t in t

One pair: straight line, constant speed

Generation runs it backwards, $t = 1$ to 0

Knowing its pair, one step would be exact

The sampler does not know the pair

Many pairs: the lines cross



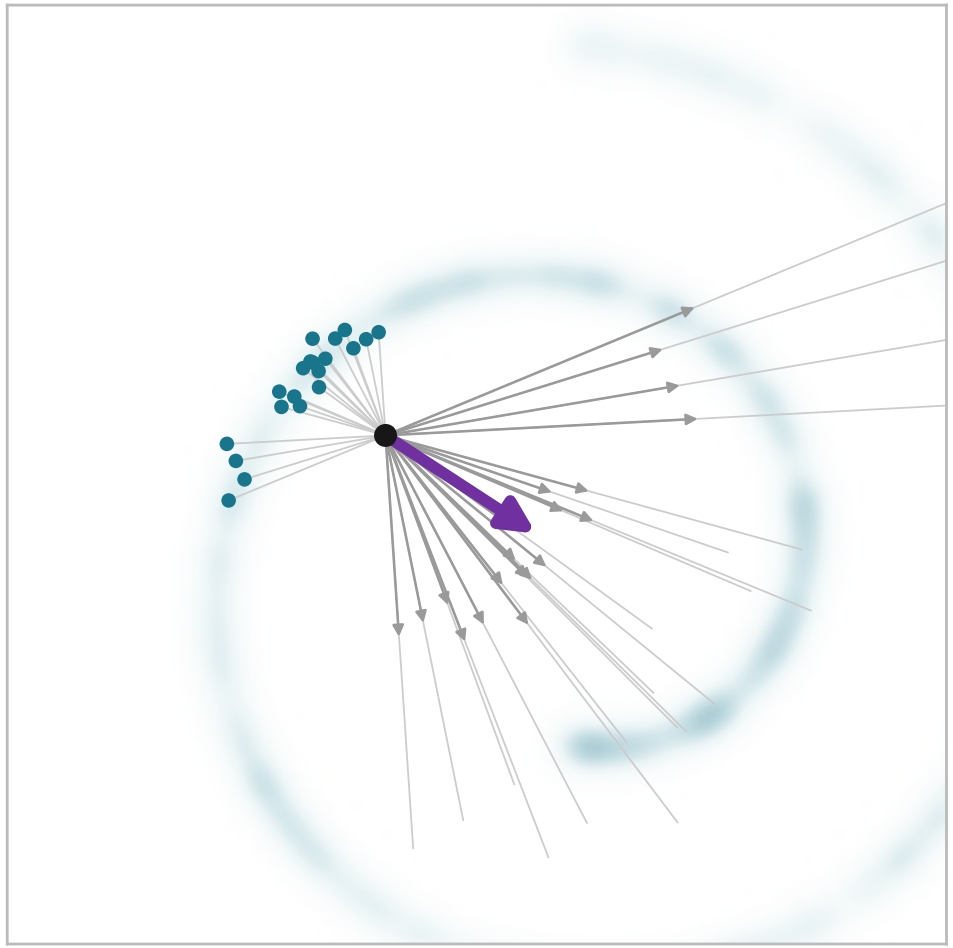
Training draws random pairs: an image x_0 (teal) and a noise ϵ (gray)

Each pair has its own straight line

Lines cross: at a crossing, two velocities

The sampler sees only x_t and t: which way?

One point, many velocities



Black dot: one point x at $t = 0.2$

Gray: velocities of 20 pairs whose lines pass through it

Teal: their images x_0, all near x

Purple: the average velocity

One answer per point and time: the average

Next: learn it by regression, then check that it reaches the data

The flow matching loss

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,\epsilon,t}\Big\| \underbrace{v_\theta(x_t, t)}_{\text{network}} - \underbrace{(\epsilon - x_0)}_{\text{known for the pair}} \Big\|^2$$

x_0 : training image. $\epsilon \sim \mathcal{N}(0, I)$: same-size noise. t: uniform in $[0, 1]$

$x_t = (1 - t)\, x_0 + t\, \epsilon$: the point on their line

L17 (DDPM) for comparison

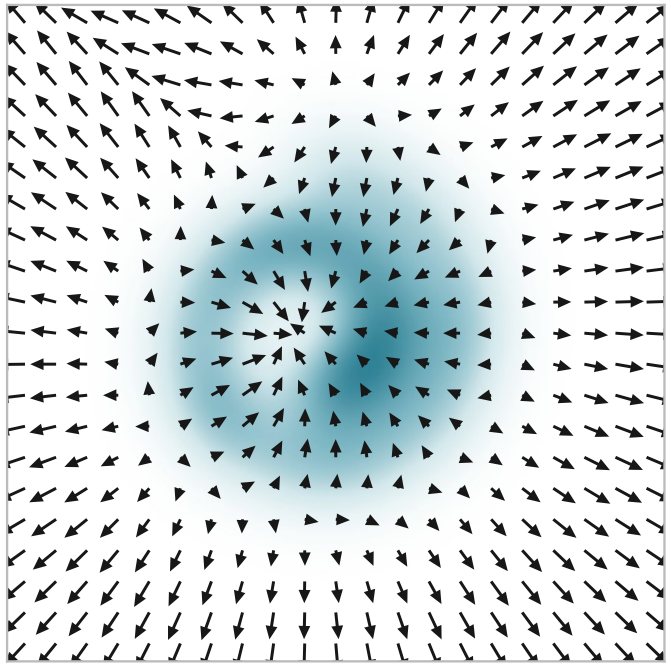
$$\big\|\epsilon_\theta(x_t, t) - \epsilon\big\|^2, \qquad x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$$

Same recipe: a pair, a time, a known target. New path, new target.

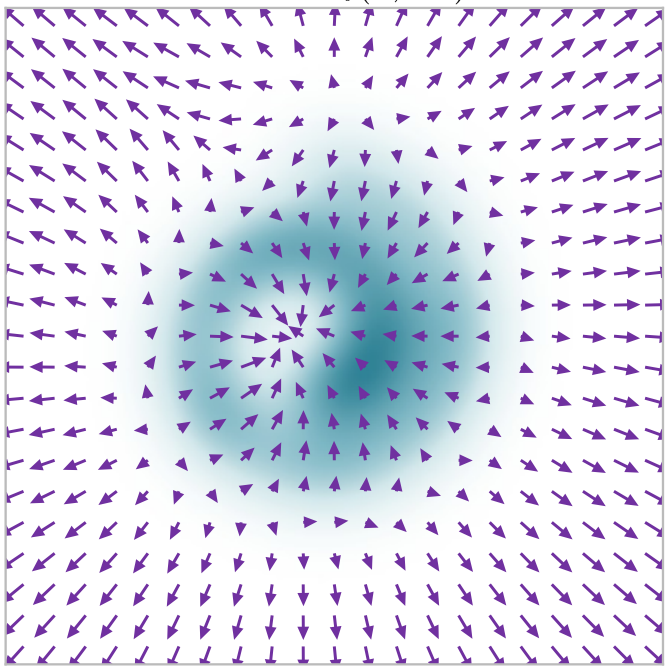
The loss learns the average velocity

$$v^*(x,t) = \underbrace{\mathbb{E}\big[\,\epsilon - x_0 \;\big|\; x_t = x\,\big]}_{\text{average over the pairs whose line passes } x \text{ at time } t}$$

exact average, $t = 0.3$



learned $v_\theta(x, 0.3)$



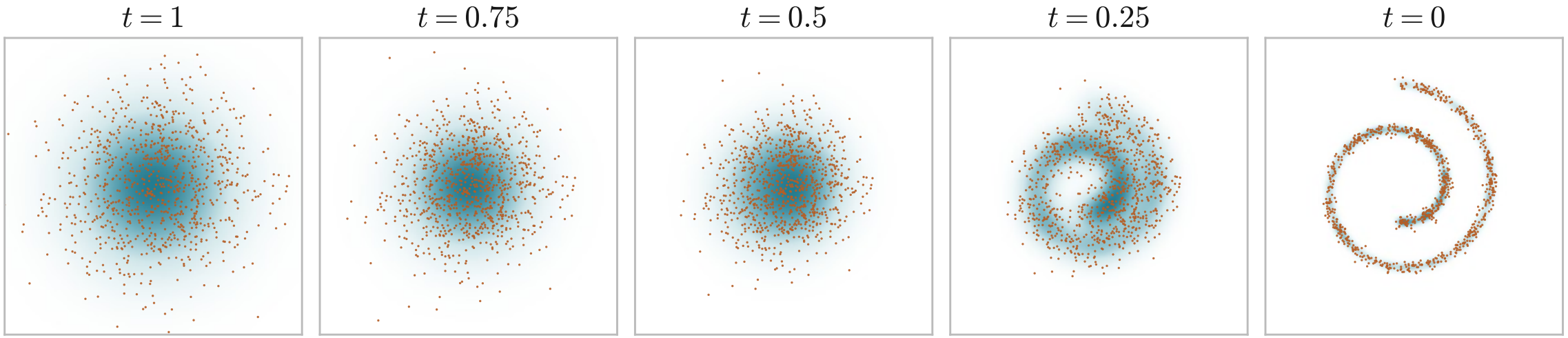
Squared error: best answer is the mean (as in L17)

Left: exact, $t = 0.3$

Right: trained network

Median angle between arrows: under 2°

Following the average reproduces every snapshot



Teal: training points x_t at each time

Orange: 2000 samples following v^* from noise

They agree at every t, so also at $t = 0$: the data

True for any data distribution

Training algorithm

Flow matching training

1 repeat

2 $x_0 \leftarrow$ a training image, $\epsilon \sim \mathcal{N}(0, I)$, $t \sim U(0, 1)$

3 $x_t = (1 - t)\, x_0 + t\, \epsilon$

4 gradient step on $\left\| v_\theta(x_t, t) - (\epsilon - x_0) \right\|^2$

5 until samples look good

L17 DDPM differs in two lines

Line 3:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$$

Line 4: target ϵ

Minibatch: own ϵ and t per image

Sampling: follow the velocity back to the data

$$x_{t-h} = x_t - \underbrace{h}_{\text{step size}} \underbrace{v_\theta(x_t, t)}_{\text{velocity here}}$$

Sampling with N Euler steps

1. $x \sim \mathcal{N}(0, I)$ (time $t = 1$)
2. for $k = N, N-1, \ldots, 1$
3. $\quad t = k/N$
4. $\quad x \leftarrow x - \frac{1}{N} v_\theta(x, t)$
5. return x (time $t = 0$)

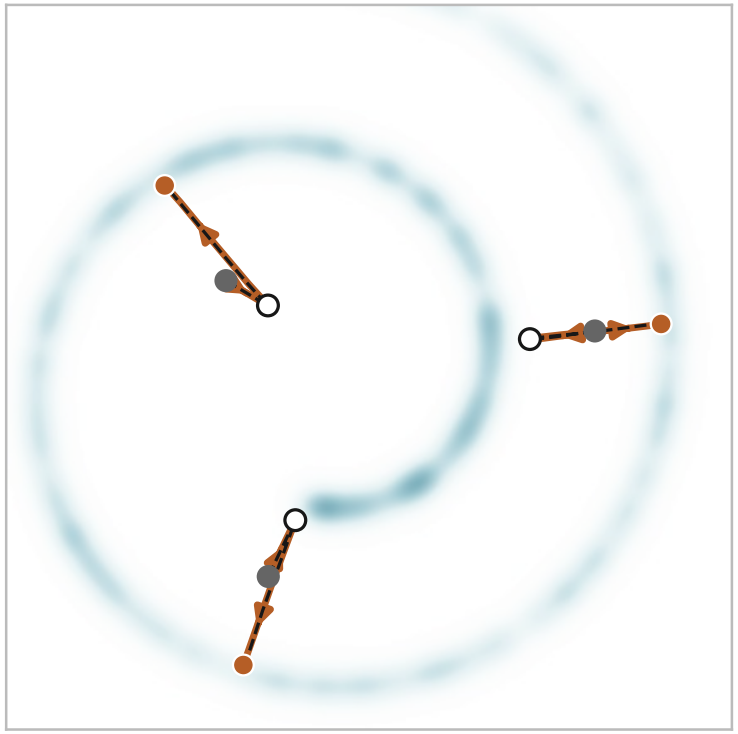
$h = 1/N$: one network call per step

N chosen at sampling time

No fresh noise, as in DDIM

Flow on the spiral

learned (orange), exact (dashed), zoomed



2000 samples, 200 Euler steps



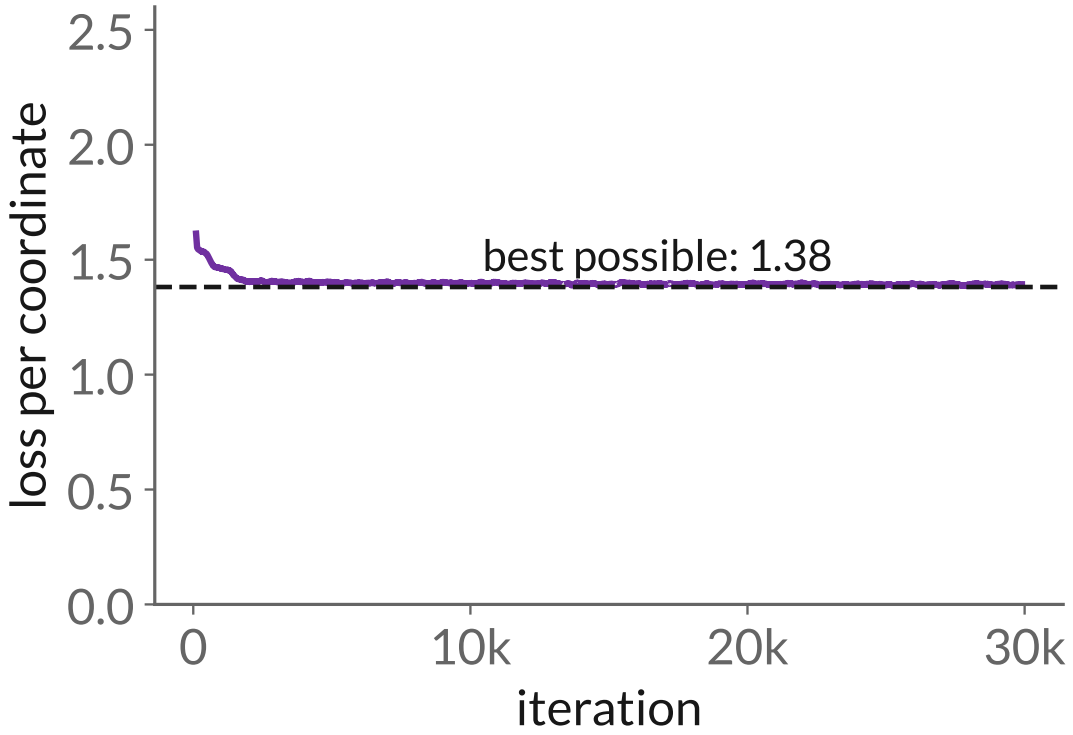
MLP, 4 layers of 256 units, trained with this loss

Left: 3 noise points; the learned paths follow the exact ones

Noise lands on the spiral

The training lines are straight; this path turns back at the ring (Section 03)

The loss does not reach zero



Target depends on the pair; input is only x_t , t

Best: the average. The spread remains

Floor 1.38 per coordinate; trained 1.39

High loss, good model: judge the samples

Reading off the clean image

$$\hat{x}_0 = x_t - t \; v_\theta(x_t, t)$$

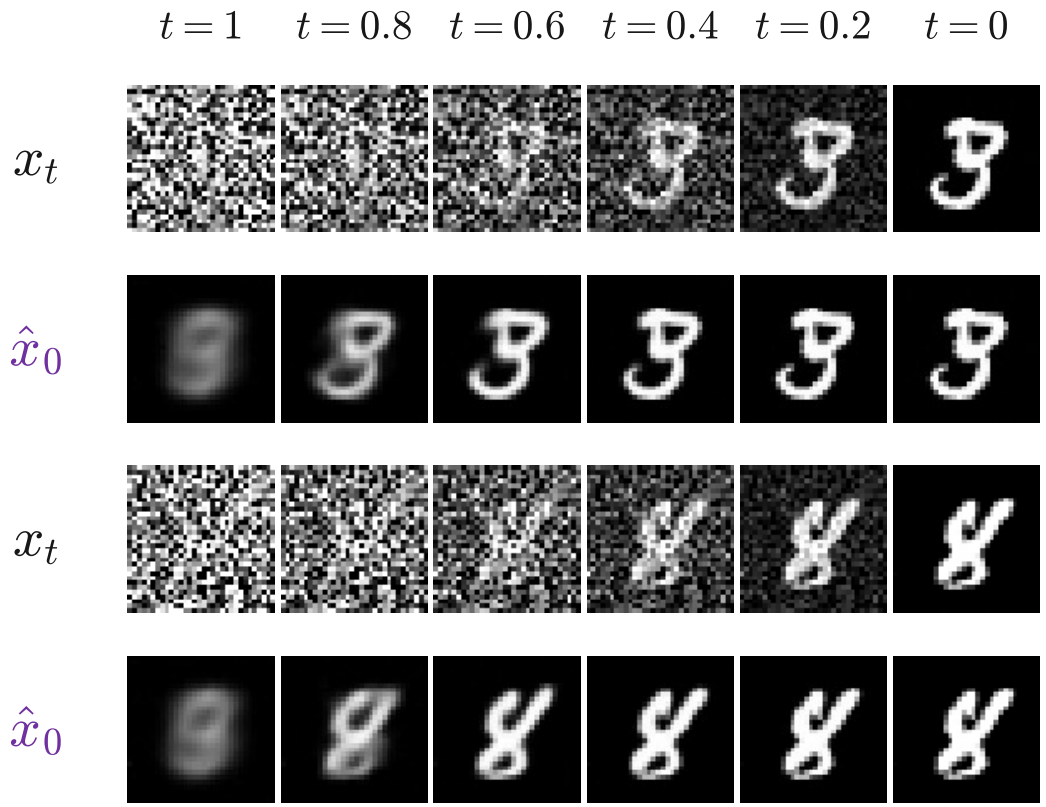
One line: $x_t = x_0 + t\,(\epsilon - x_0)$

Solve: $x_0 = x_t - t\,(\epsilon - x_0)$

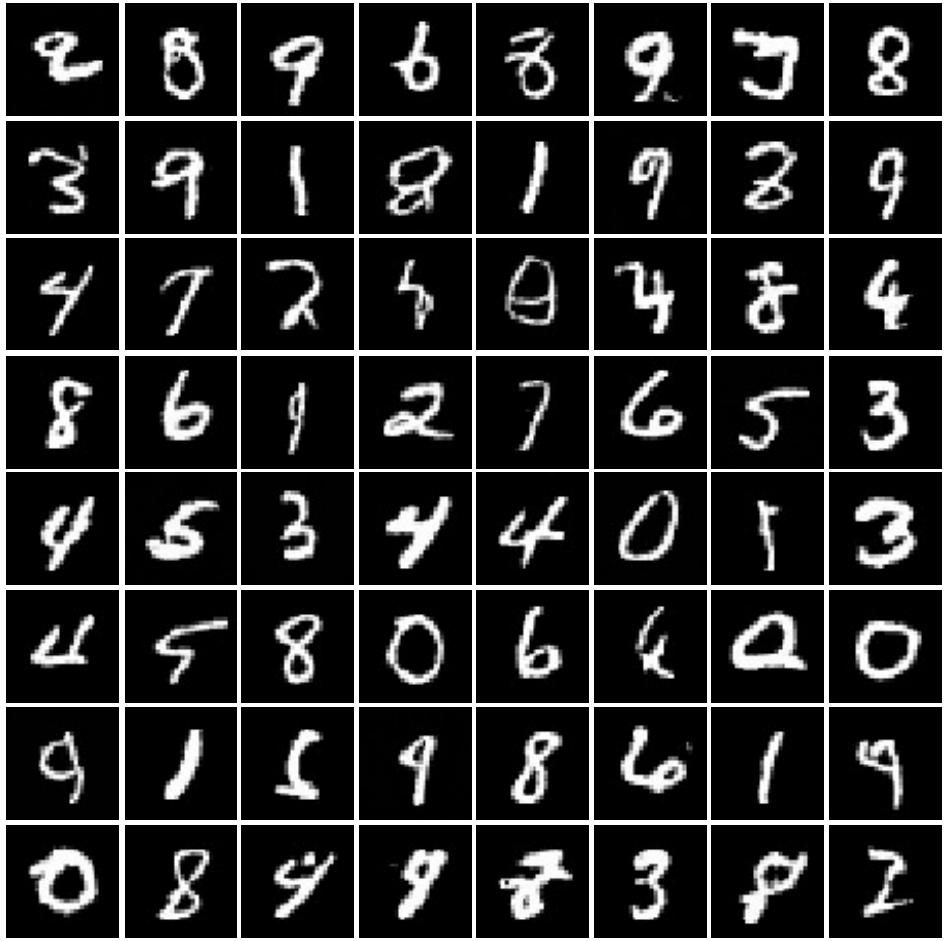
Average velocity instead: estimate $\hat{x}_0$

Near $t = 1$: blurry average digit

Sharp while x_t is still noisy



Samples from a small flow model



MNIST, 32×32

The L17 U-Net, 1.6 M parameters

20k iterations, about 1.5 hours on a laptop GPU

50 Euler steps

No label given; Section 04 adds labels

Lecture outline

01 Choosing the path

02 Flow matching

03 Paths, steps and reflow

04 Conditional generation

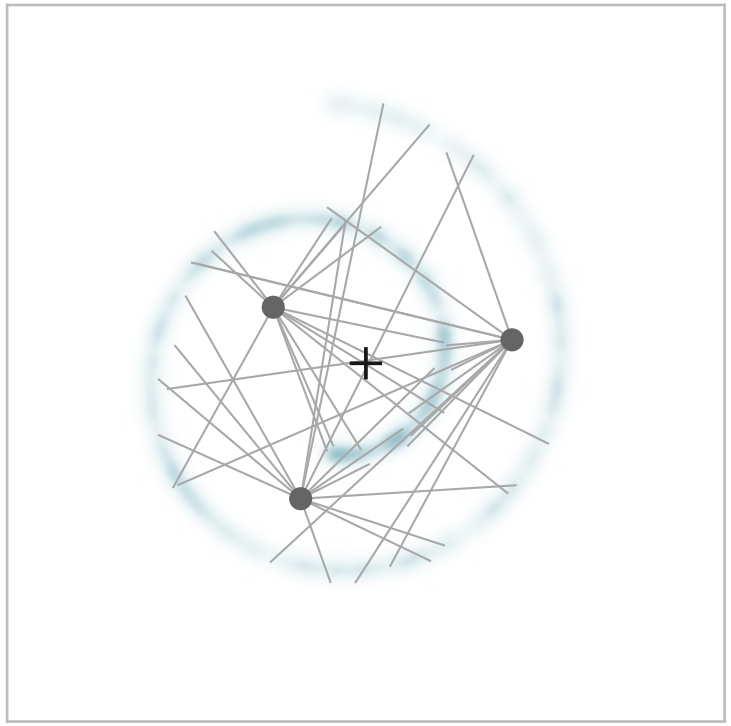
05 Classifier-free guidance

06 Generating large images

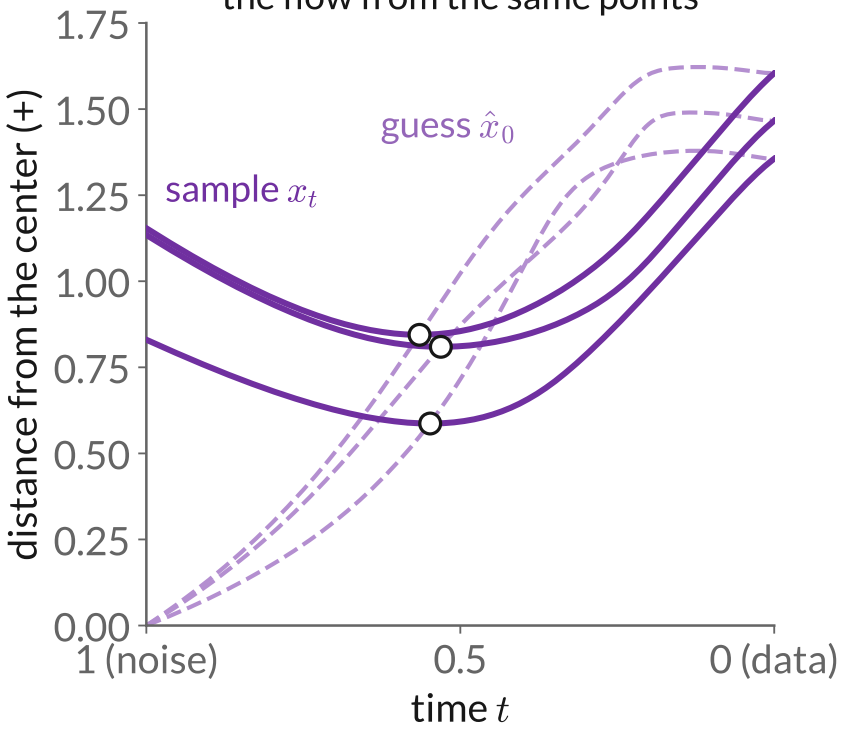
07 A text-to-image system and its uses

The flow does not follow the training lines

training lines from three noise points



the flow from the same points



Each step moves toward the current guess $\hat{x}_0$ (slide 19)

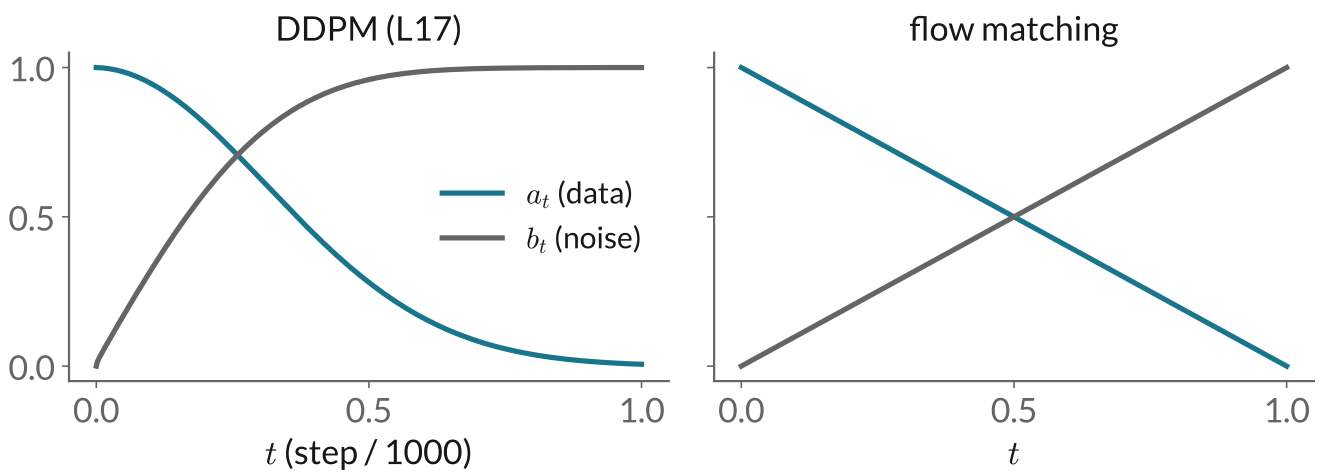
Early: the guess is near the center, so the sample heads in

Later: the guess moves out to an arm; the sample turns back (ring)

Straight lines, folded flow

Flow matching and diffusion: one family

$$x_t = \underbrace{a_t}_{\text{data weight}}\, x_0 + \underbrace{b_t}_{\text{noise weight}}\, \epsilon$$



DDPM (L17)

$a_t = \sqrt{\bar{\alpha}_t}$, $b_t = \sqrt{1 - \bar{\alpha}_t}$; predicts ϵ

Flow matching

$a_t = 1 - t$, $b_t = t$; predicts $\epsilon - x_0$

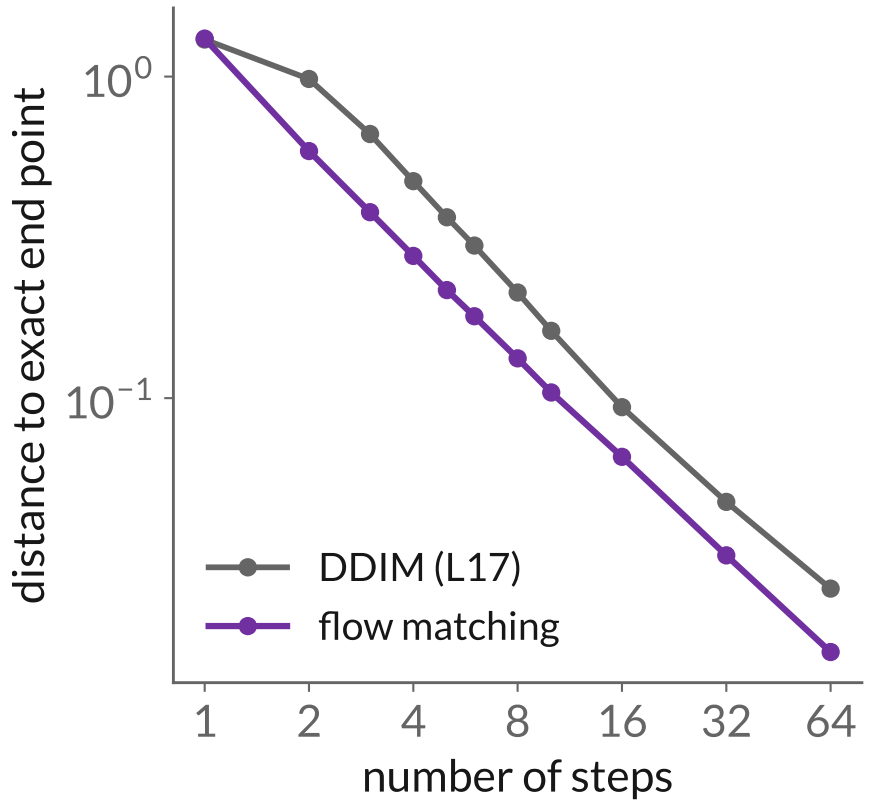
Divide by a_t: both give $x_0 + (b_t / a_t)\, \epsilon$

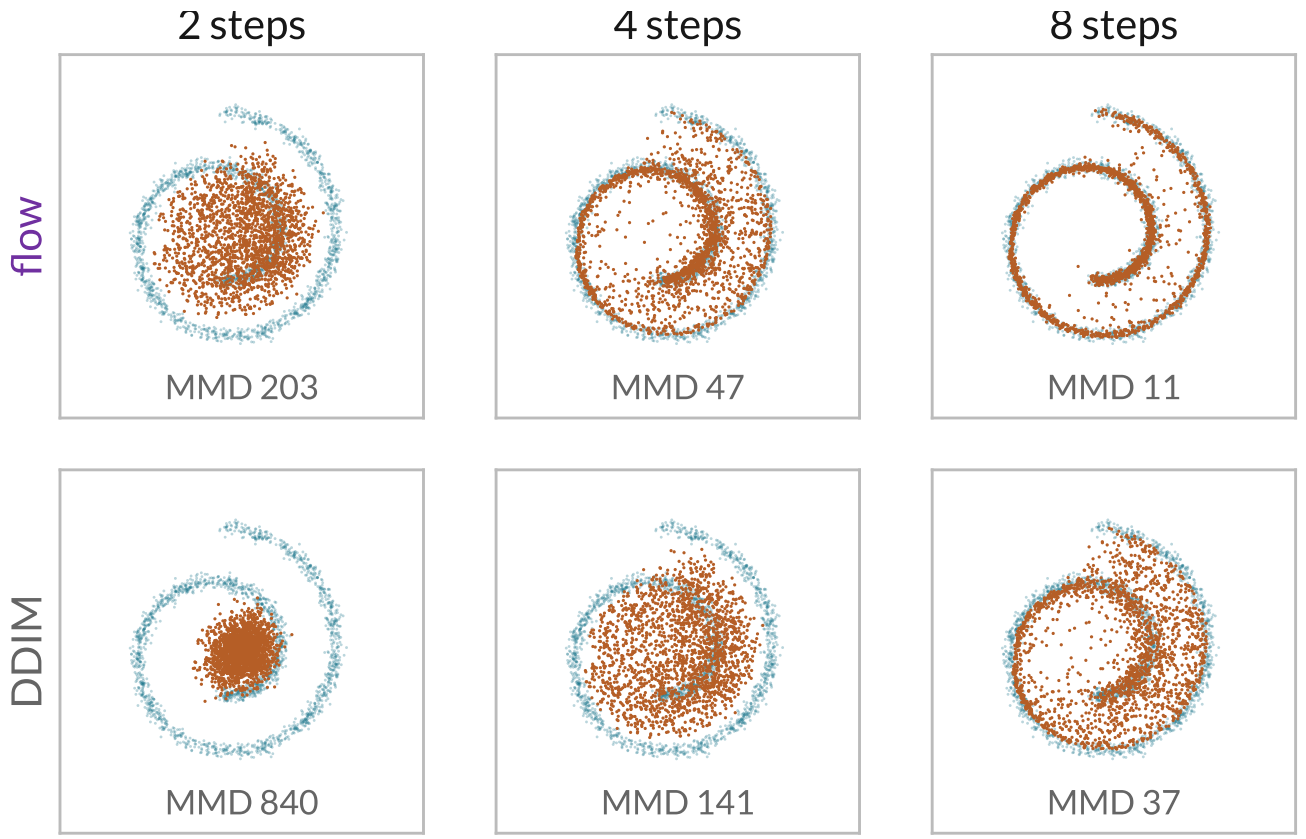
Same noisy points, a different clock

Exact samplers: same end point

Predictions convert: $\hat{x}_0 = x_t - t\, v$, $\hat{\epsilon} = x_t + (1 - t)\, v$

How many steps? The 2-D spiral



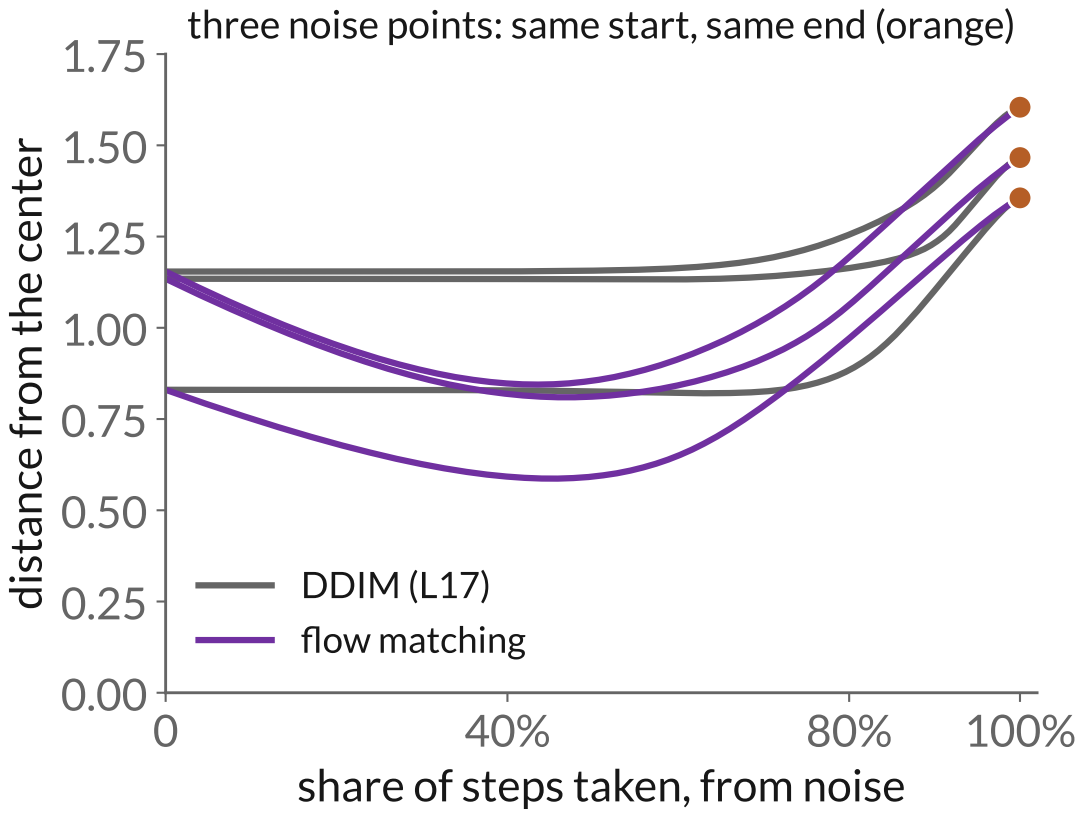


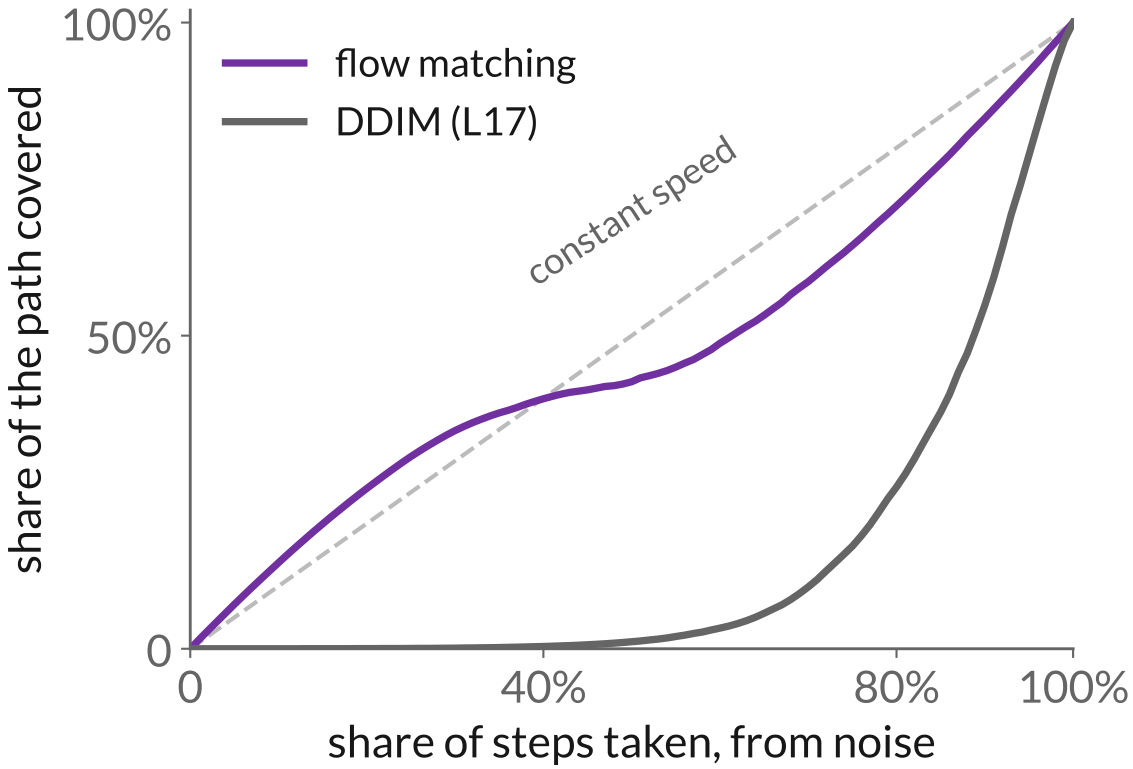
Exact velocities for both: only the path differs

Flow matching with 10 steps ≈ DDIM with 16

MMD (×1000): distance between two point clouds; lower is closer to the data

Fewer steps from even speed



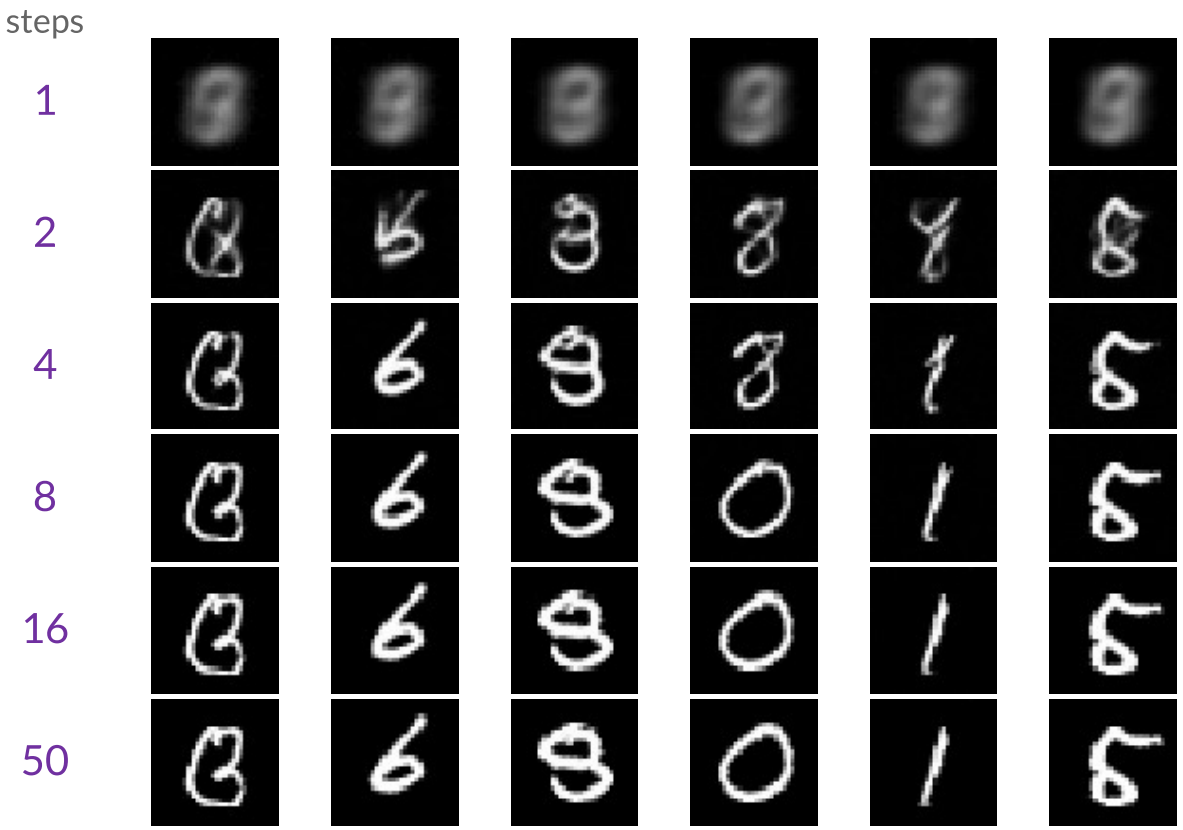


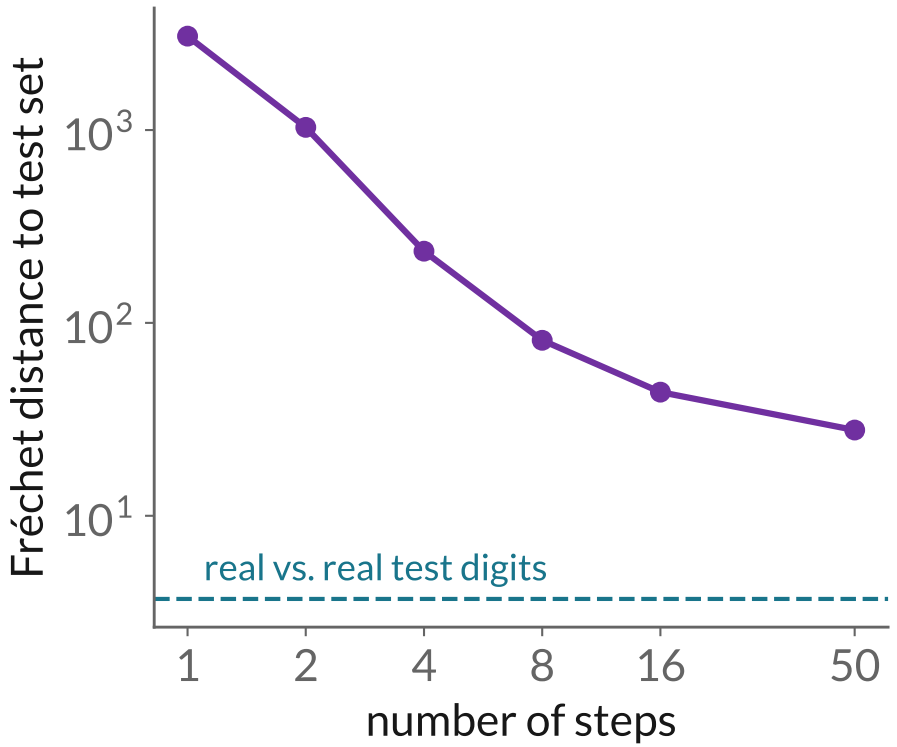
Left: exact velocities, so both reach the same end (previous slides); routes and timing differ

Right, 400 noise points: DDIM covers 1% of its path in the first half of the steps, 74% in the last 20%

Equal steps suit even speed: the flow gains from timing, not shape

How many steps? MNIST

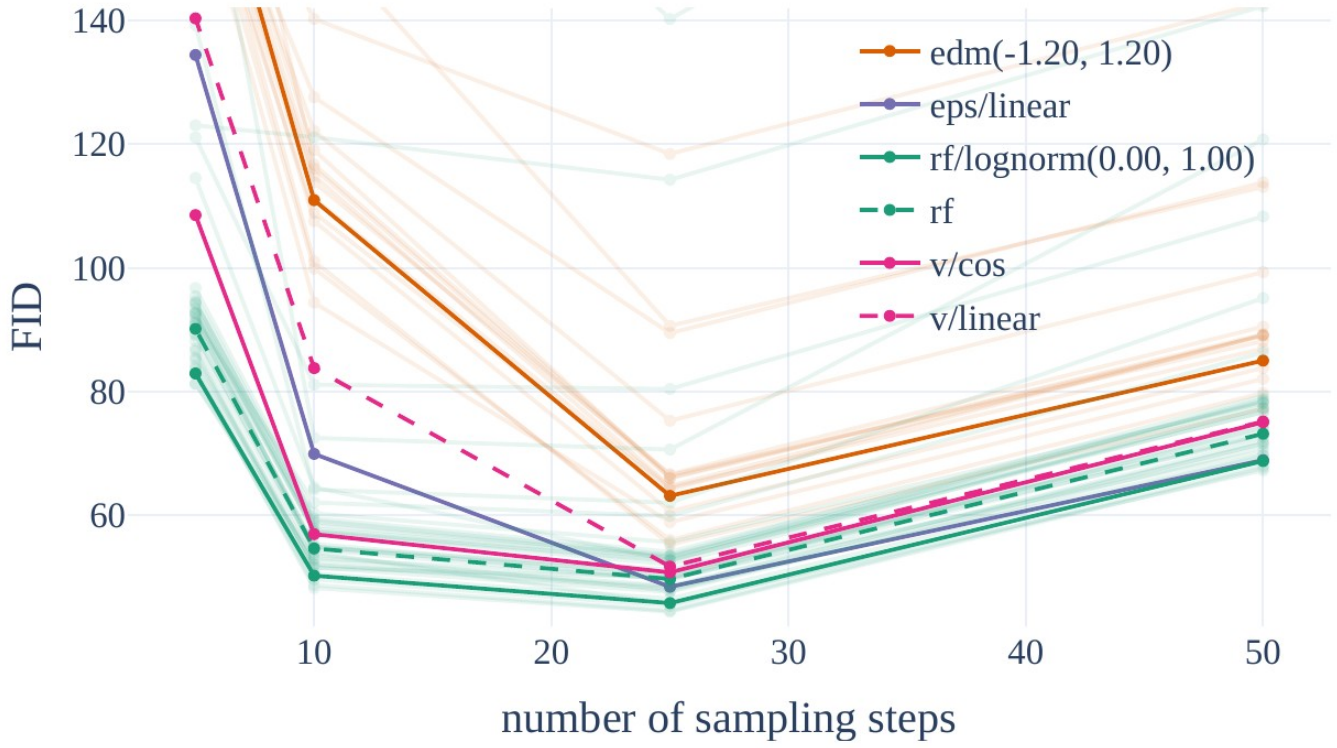




Same noise per column; few steps blur

Fréchet distance: classifier features of 2000 samples vs test digits; lower is better

At scale: flow matching needs fewer steps



Text-to-image models differing in path and target

Green (rf): flow matching. Purple: DDPM-style

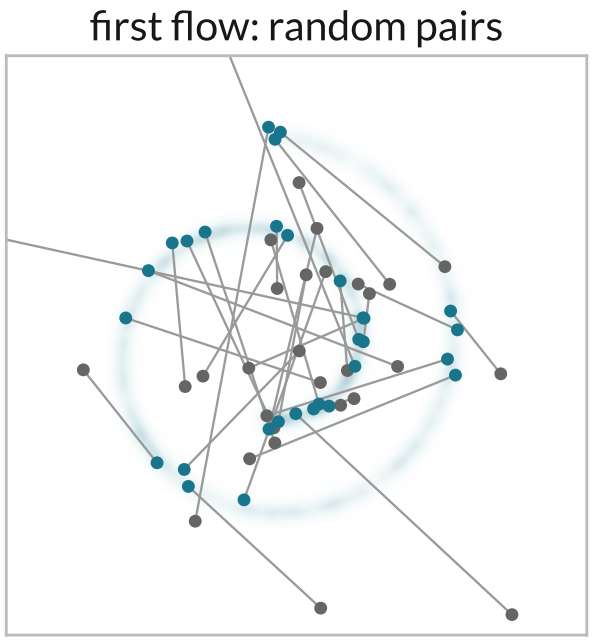
FID: Fréchet distance on photos; lower is better

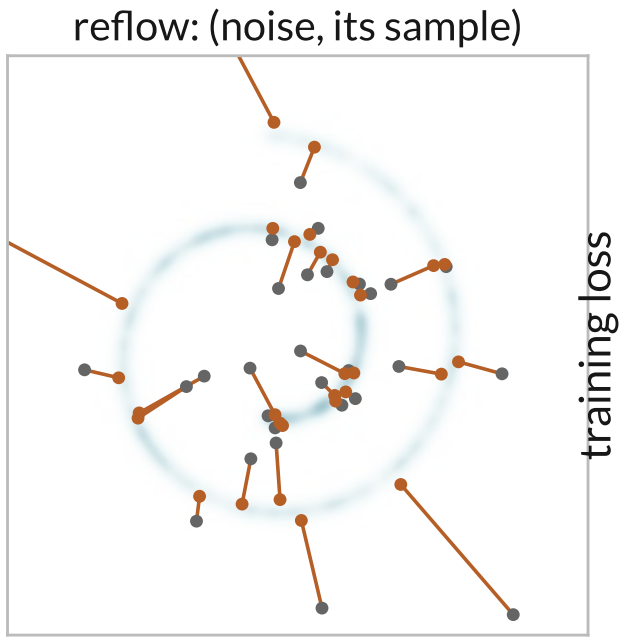
At 5–10 steps, flow matching wins clearly

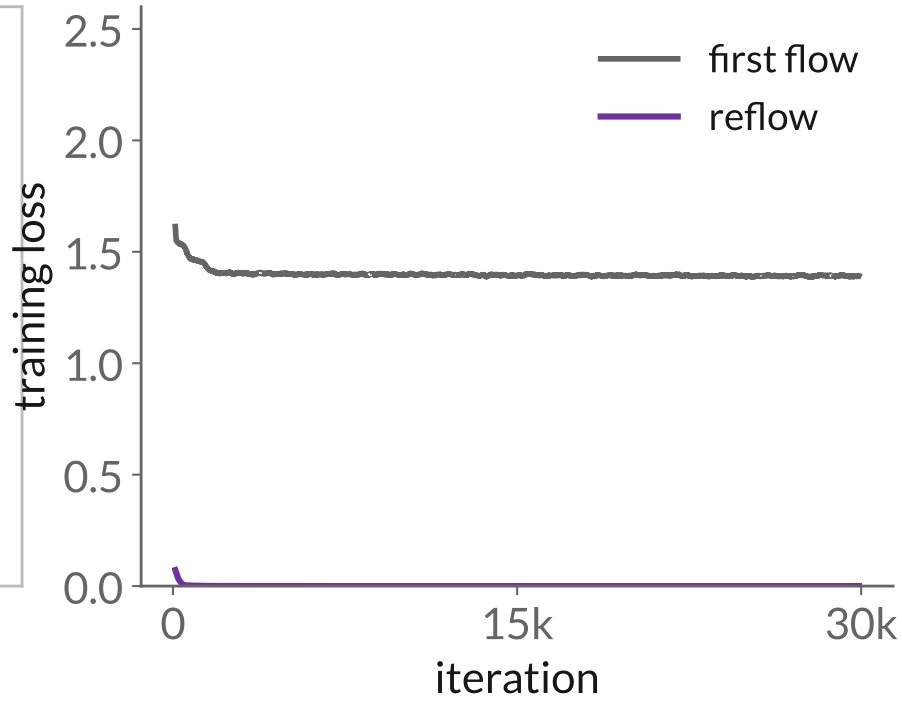
All sampled with Euler steps

Source: Stable Diffusion 3 (2024)

Straightening the paths: reflow





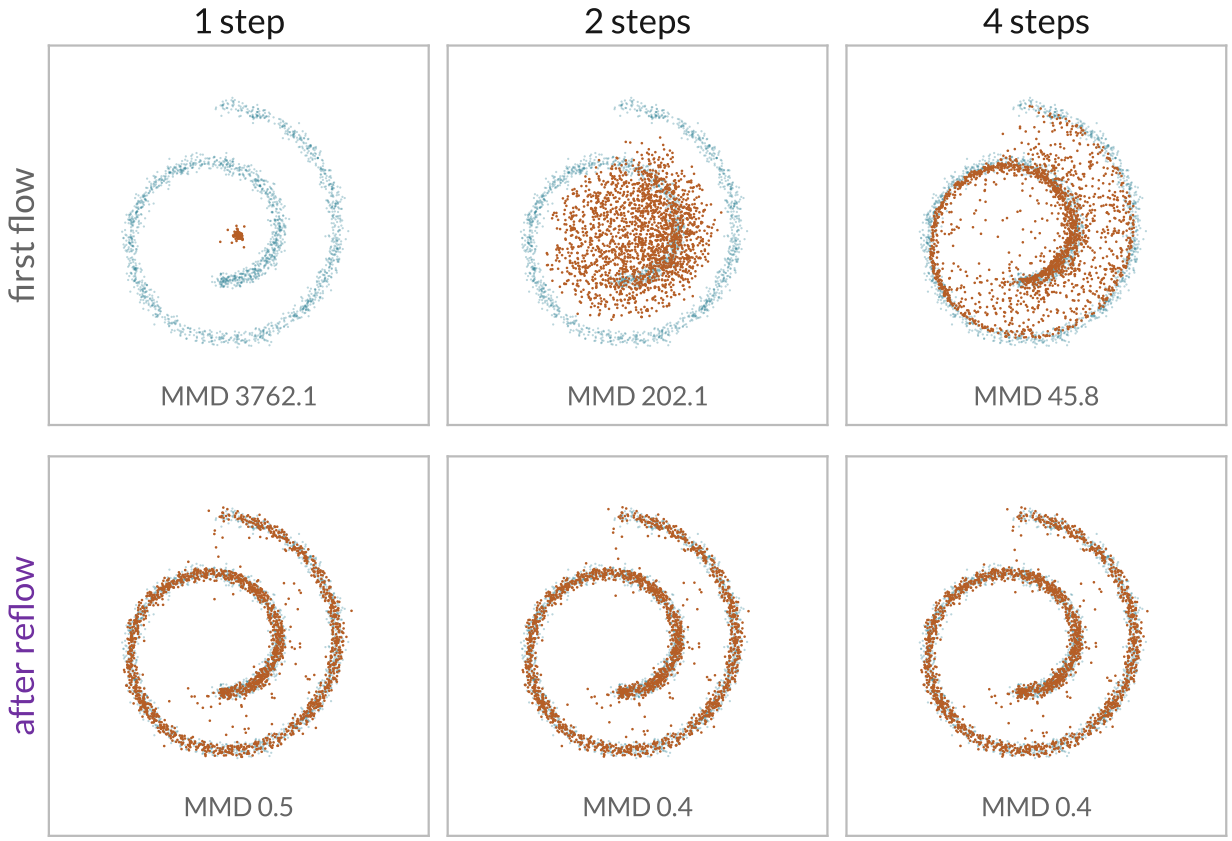


Run the first flow: each noise ϵ gets its sample x_0

Retrain on these pairs: their lines rarely cross

One velocity per point: loss near zero, straight paths

Reflow: one step is enough here



Top: first flow. Bottom: after reflow

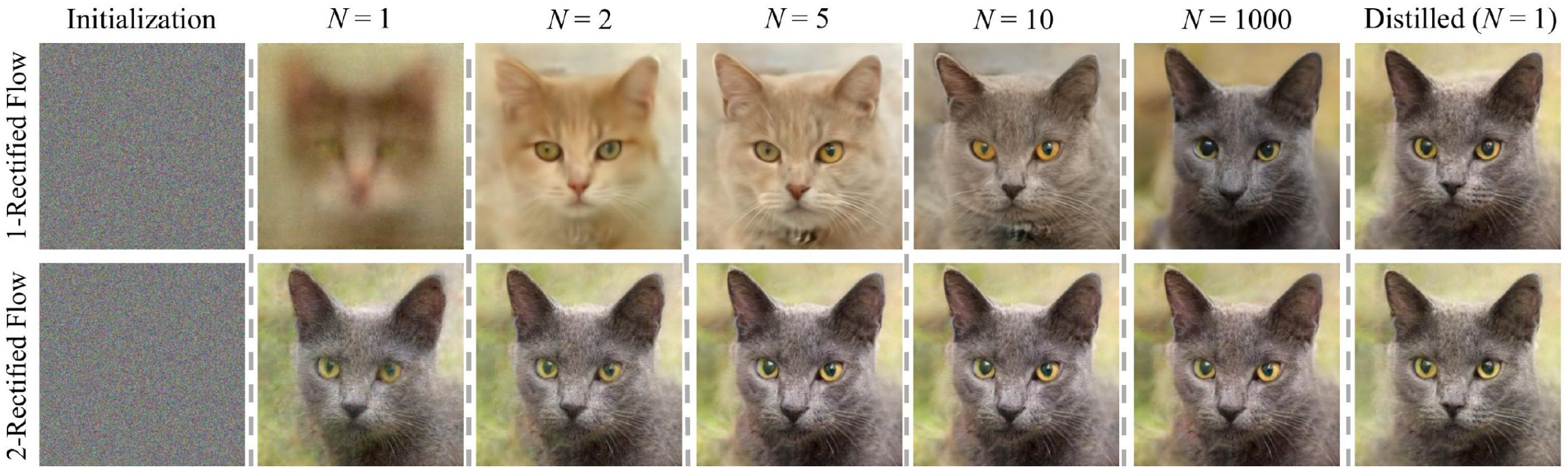
One step after reflow: MMD 0.5; first flow at 128 steps: 0.4

Price

A second training run, plus generating the pairs

Errors of the first flow are copied into the pairs

Few steps on images: reflow and distillation



Top: first flow, blurry at $N = 1$. Bottom: after one reflow, sharp at $N = 1$

Distilled: a network that jumps to the result in one call

Source: Rectified Flow (2023)

Lecture outline

01 Choosing the path

02 Flow matching

03 Paths, steps and reflow

04 Conditional generation

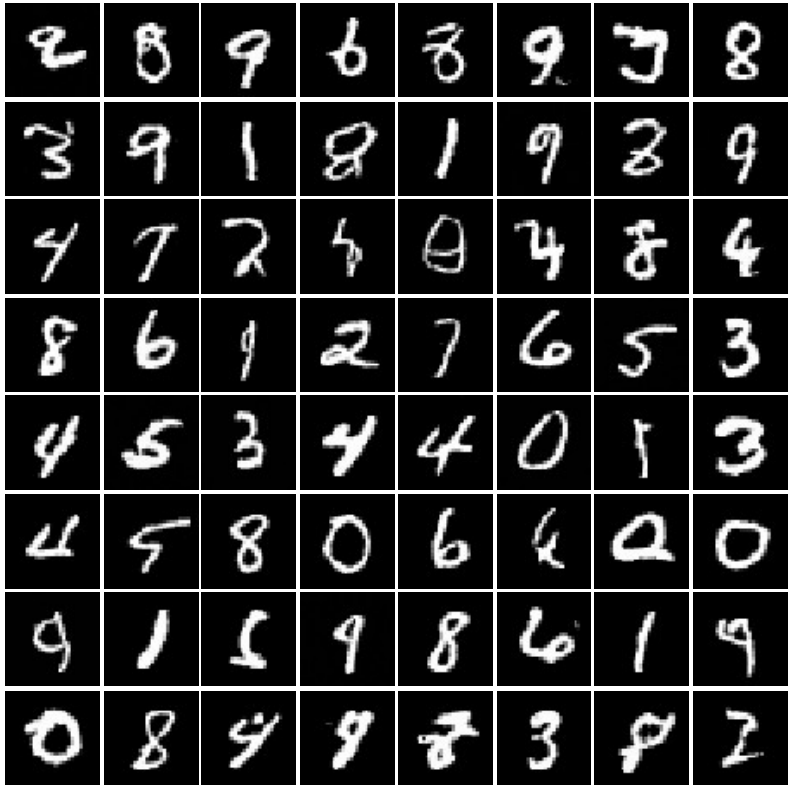
05 Classifier-free guidance

06 Generating large images

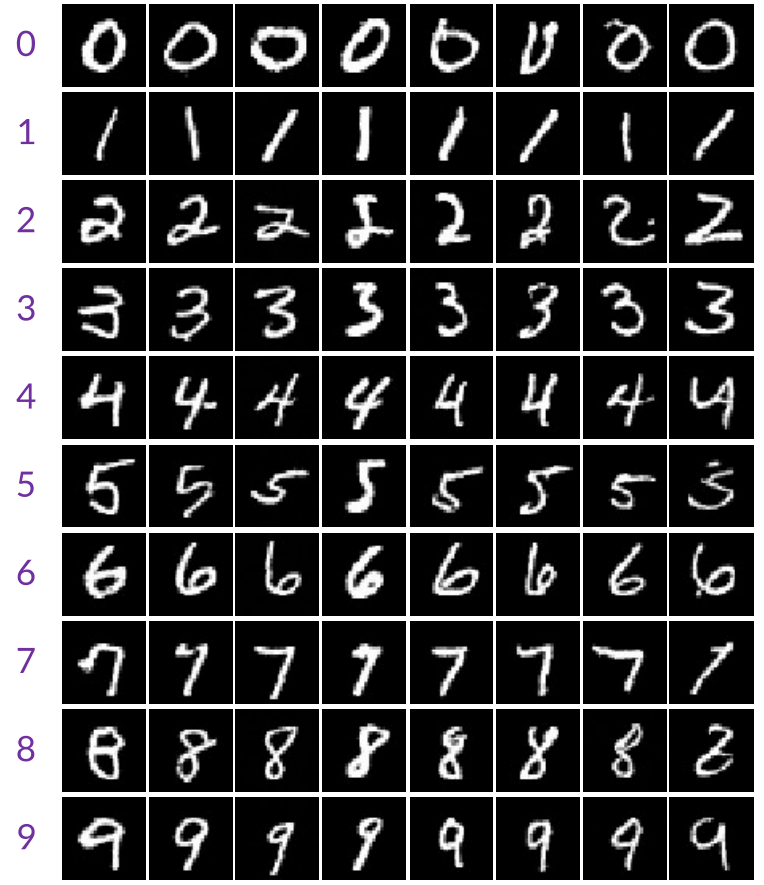
07 A text-to-image system and its uses

From random samples to requested ones

no label: a random digit



label $c = 0, \ldots, 9$ (rows)



Same noise in each column

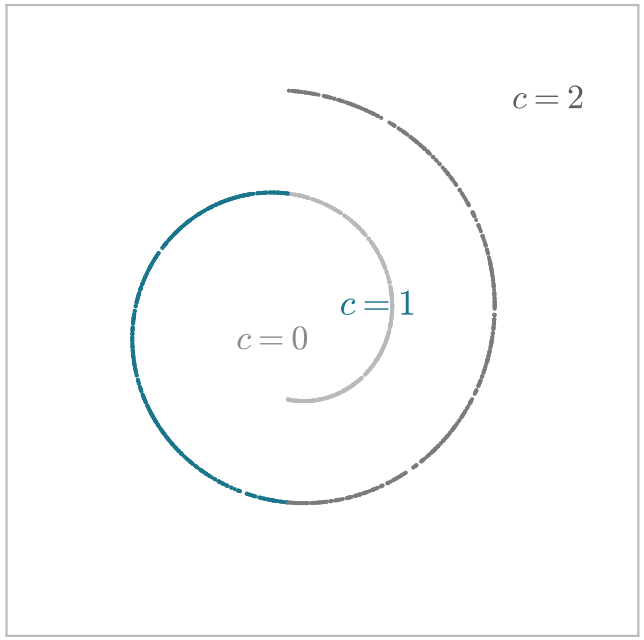
Label: which digit.
Noise: the style

Text-to-image: a sentence as c

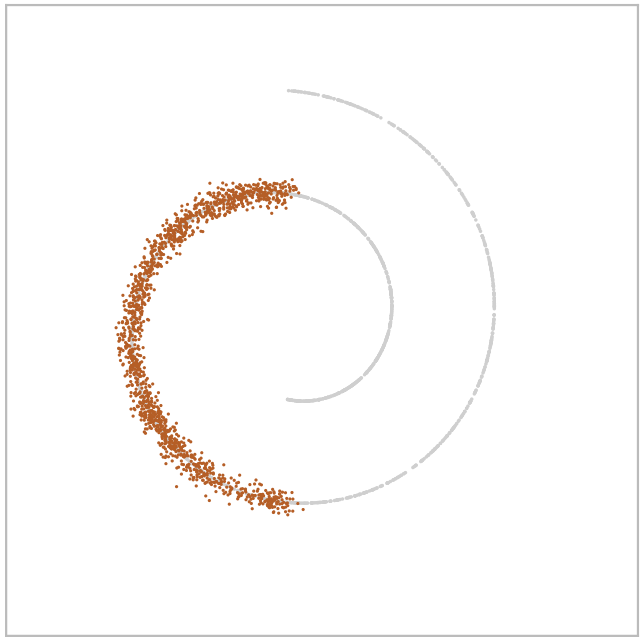
Conditional flow matching

$$\mathcal{L}(\theta) = \mathbb{E}_{(x_0,c),\epsilon,t}\left\|v_\theta(x_t, t, \underbrace{c}_{\text{the condition}}) - (\epsilon - x_0)\right\|^2$$

three labelled arcs



samples given $c = 1$



Data: pairs of image x_0 and condition c

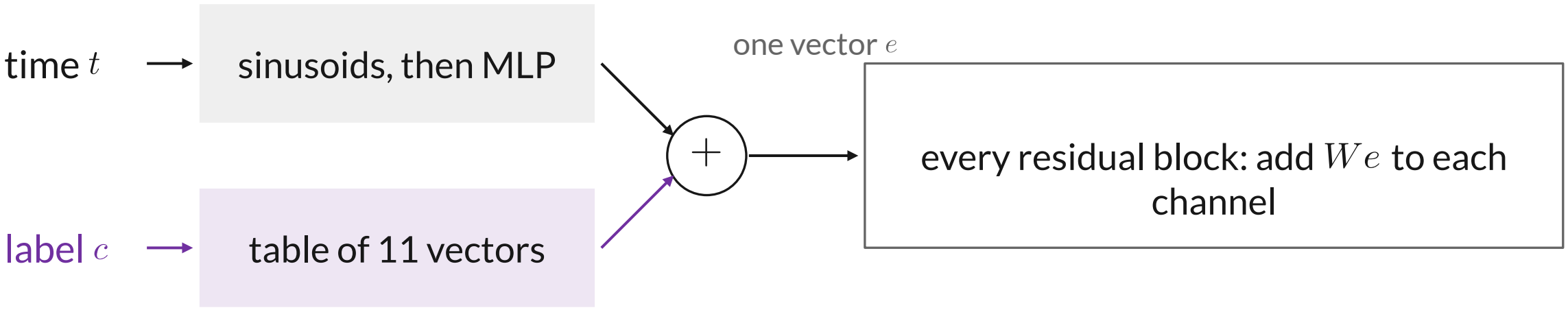
c: one more network input

Learns the average velocity for that c

Sampling: same steps, c fixed

Left: 3 classes. Right: class 1 samples

A label enters like the time



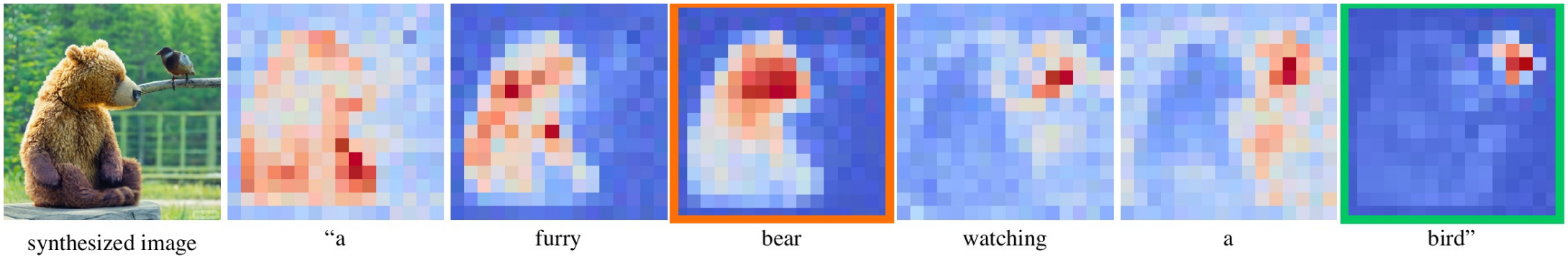
A label is a lookup: one learned vector per class, plus one for “no label” (Section 05)

Adding We shifts each channel by one number, the same at every position

A sentence needs more than one vector

A sentence enters through cross-attention

$$\text{softmax}\left(\underbrace{Q}_{\text{image}} \underbrace{K^\top}_{\text{text}} / \sqrt{d} \right) \underbrace{V}_{\text{text}}$$



Recap: tokens and embeddings (L10), cross-attention (L11), CLIP (L14), T5 (L12)

A frozen, pretrained text encoder gives one vector per token; keep them all

Queries: image positions in the velocity network. Keys and values: the token vectors

“bear” positions read “bear”; “bird” positions read “bird”

Source: Prompt-to-Prompt (2022)

Lecture outline

01 Choosing the path

02 Flow matching

03 Paths, steps and reflow

04 Conditional generation

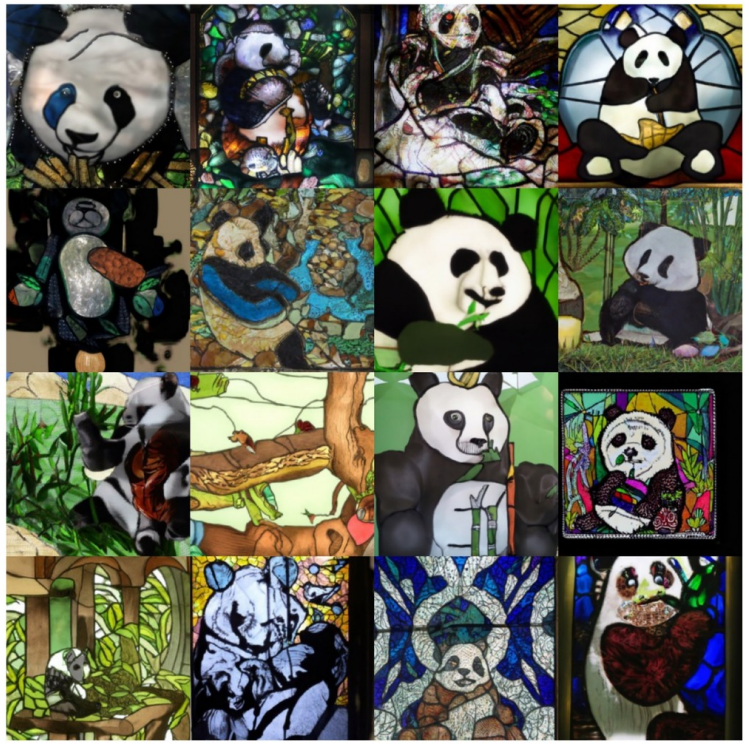
05 Classifier-free guidance

06 Generating large images

07 A text-to-image system and its uses

Conditioning alone follows the prompt loosely

conditional only



with guidance (this section)



GLIDE: “a stained glass window of a panda eating bamboo”

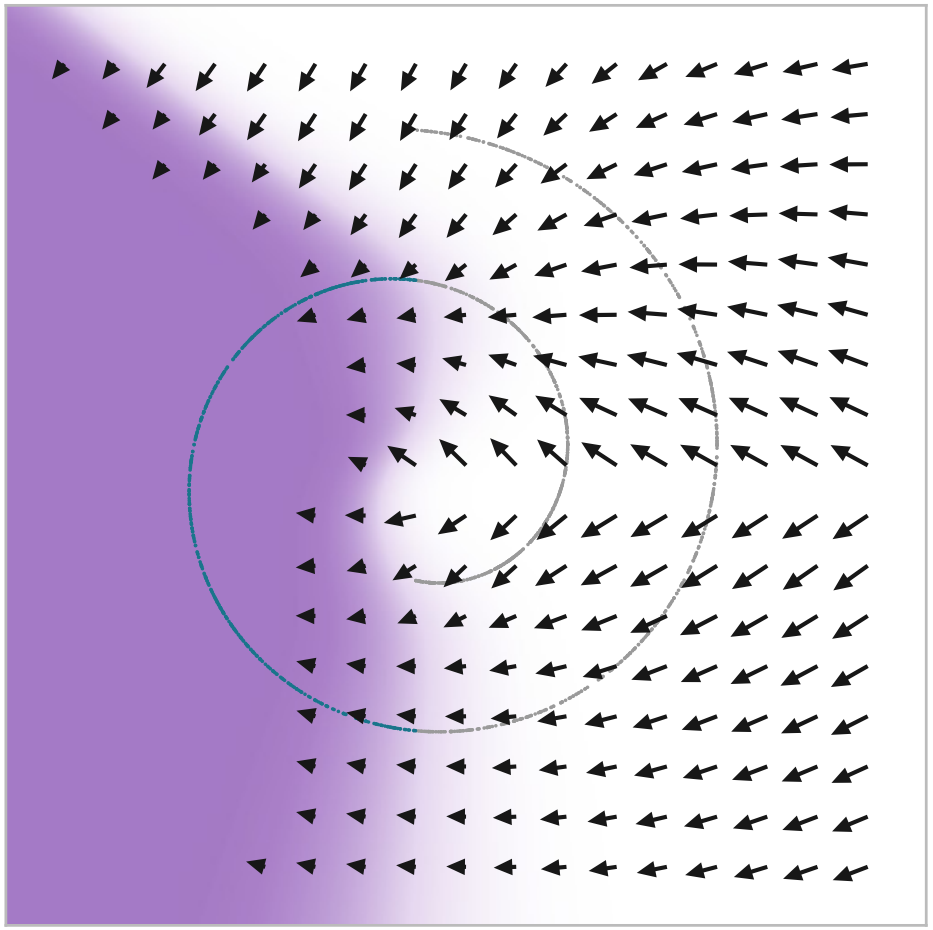
Left: many miss part of the prompt

MNIST: 2% of samples read as another digit

Wanted: a knob that sets how closely c is followed

Source: GLIDE (2022)

Classifier guidance: push toward the class



A classifier $p_\phi(c \mid x_t)$, trained on noisy inputs

Each step also climbs $\log p_\phi(c \mid x_t)$

Shading: class probability at $t = 0.3$; arrows: the push

Price

A second network, trained on noisy images

Its gradients can fool the classifier instead of improving the image

The classifier's push from two generators

Bayes' rule at time t: $p_t(c \mid x) = p_t(x \mid c)\, p(c) \,/\, p_t(x)$

Take the log, then the gradient in x; $p(c)$ does not depend on x and drops out:

$$\underbrace{\nabla_x \log p_t(c \mid x)}_{\text{the classifier's push}} = \underbrace{\nabla_x \log p_t(x \mid c)}_{\text{score of images with label } c} - \underbrace{\nabla_x \log p_t(x)}_{\text{score of all images}}$$

Score (L17): the direction in which x becomes more likely; a generator's prediction estimates it

A generator trained with c gives the first score; one trained without c gives the second

No classifier needed: two generators supply its push

One network, with and without the condition

Training examples (x_0, c): the label is replaced by $\varnothing$ about one time in ten



5 0 4 1 9 2 $\varnothing$ 3 1 4

Same loss and target $\epsilon - x_0$; only the label input changes

$v_\theta(x, t, c)$: velocity for images with label c

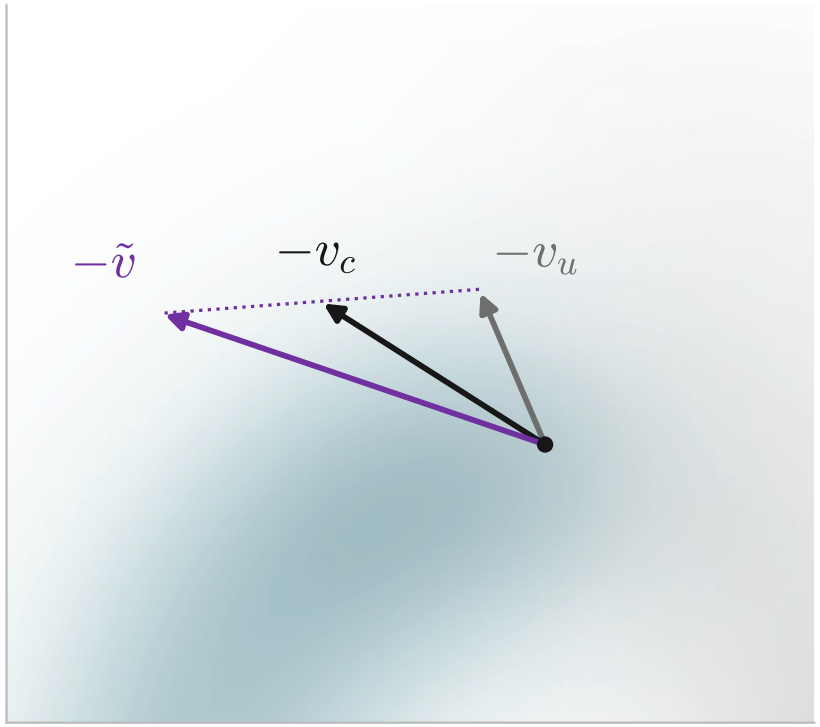
$v_\theta(x, t, \varnothing)$: velocity for all images

One network gives both velocities

$\varnothing$: the extra row of the label table; for text, the empty prompt

Classifier-free guidance

$$\tilde{v} = \underbrace{v_\theta(x, t, \varnothing)}_{\text{no condition}} + \underbrace{w}_{\text{guidance weight}} \left(\underbrace{v_\theta(x, t, c) - v_\theta(x, t, \varnothing)}_{\text{what the condition adds}} \right)$$



$w = 0$: ignore c. $w = 1$: plain conditional

$w > 1$: further in the direction c adds

Motion along $-(v_c - v_u)$: the classifier's push, rescaled

Two network calls per step

Spiral at $t = 0.3$; requested class: the middle arc

Why the velocity difference points toward the class

On the flow matching path the noise has scale t, so the score is $-\hat{\epsilon}/t$ (as in L17)

With $\hat{\epsilon} = x + (1-t)\,v$ (slide 23), velocity and score are linked:

$$v = -\frac{x + t\,\nabla_x \log p_t(x)}{1-t}$$

Same x and t, with and without c: the x terms cancel

$$v_\theta(x, t, c) - v_\theta(x, t, \varnothing) = -\frac{t}{1-t}\,\nabla_x \log p_t(c \mid x)$$

Sampling moves along $-v$: uphill on $\log p_t(c \mid x)$, the classifier hidden inside the network

Guided sampling algorithm

Sampling with guidance weight w

1 $x \sim \mathcal{N}(0, I)$

2 for $k = N, \ldots, 1$

3 $\quad t = k/N$

4 $\quad v_c = v_\theta(x, t, c), v_u = v_\theta(x, t, \varnothing)$

5 $\quad \tilde{v} = v_u + w\,(v_c - v_u)$

6 $\quad x \leftarrow x - \frac{1}{N}\,\tilde{v}$

7 return x

New: lines 4 and 5

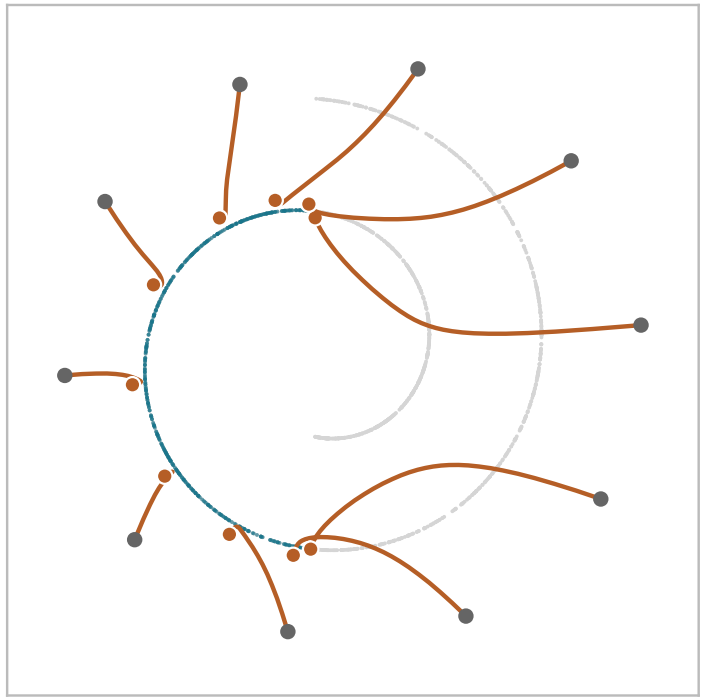
Twice the network calls

w set at sampling time

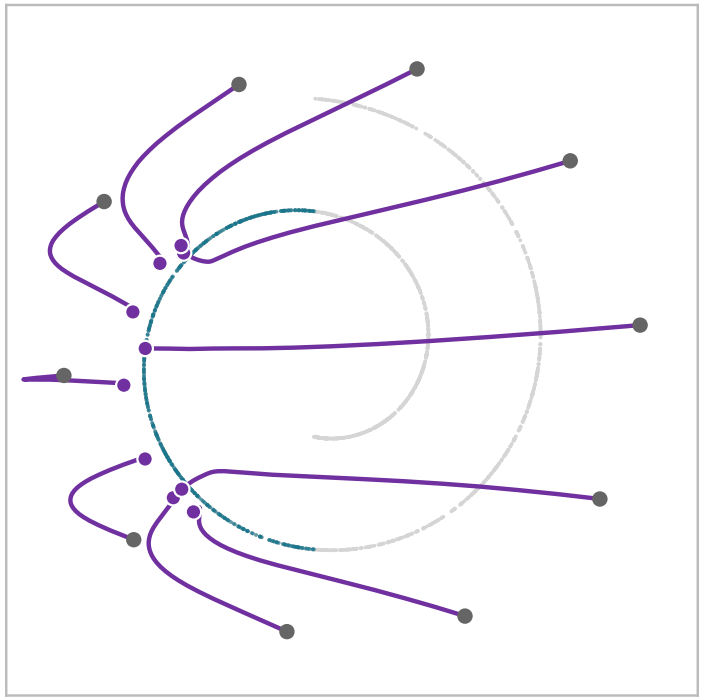
Text-to-image: often $w = 3$ to 8

Guidance bends the paths

$w = 1$



$w = 4$

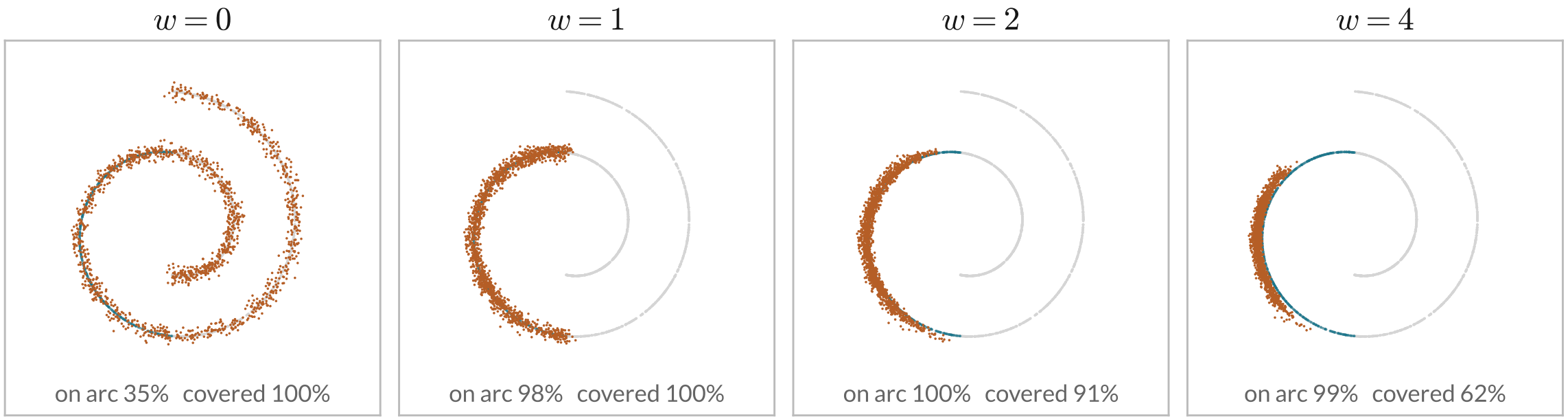


Requested: the middle arc; exact velocities

$w = 1$: paths end all along the arc

$w = 4$: pushed deep into it, away from the other classes

Guidance on the 2-D spiral

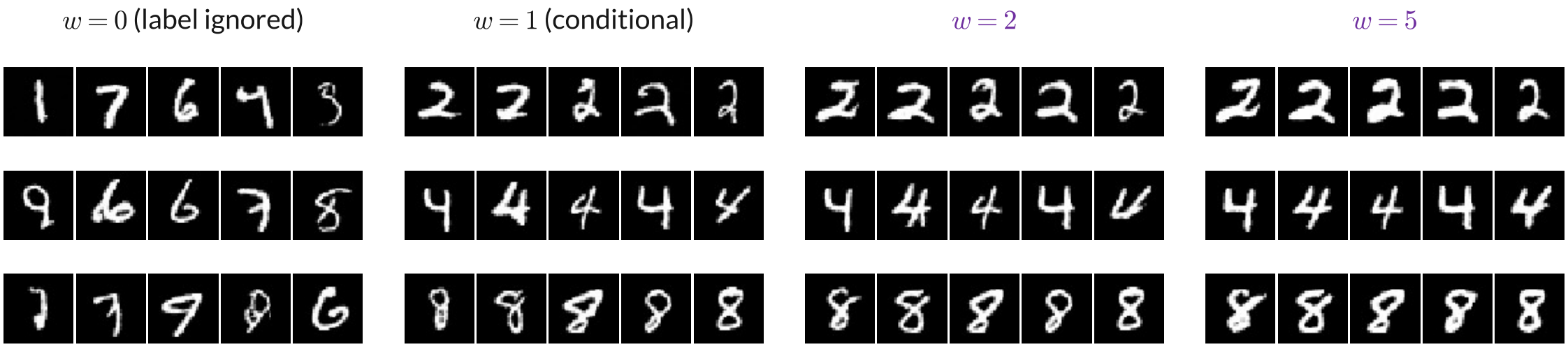


On the requested arc: 98% ($w = 1$), 100% ($w = 2$), 99% ($w = 4$)

Share of the arc that is covered: 100%, 91%, 62%

Strong guidance: samples crowd into part of the class

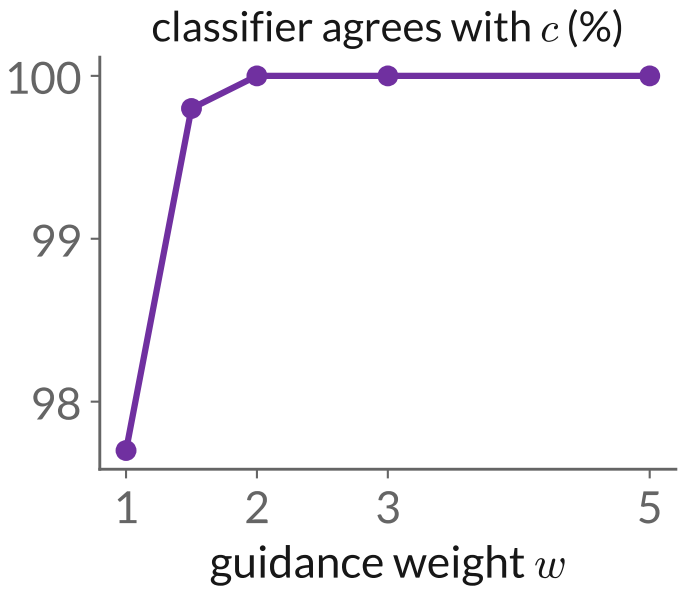
Guidance on MNIST

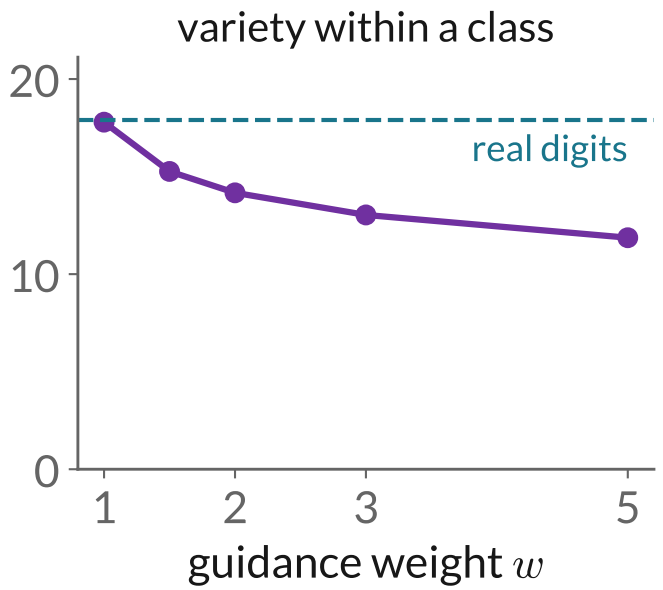


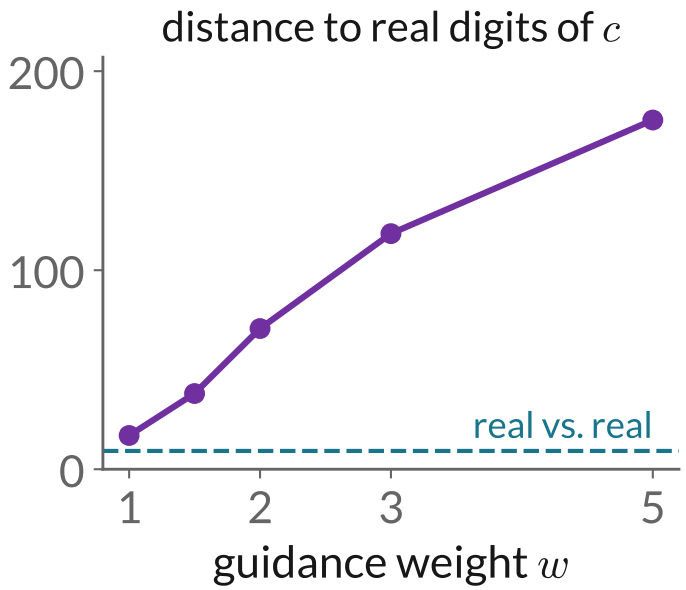
Rows: labels 2, 4, 8. Same noise in every panel

Larger w: bolder, more alike digits

Agreement versus variety







Agreement with the label: 97.7% at $w = 1$, 100% at $w = 2$

Variety: as in real digits at $w = 1$, then falls

Distance to real digits: lowest at $w = 1$, grows with w

Guidance trades variety for agreement

Guidance at scale



ImageNet “malamute”, 64×64 ; more guidance to the right

Left: varied, often unclear. Right: clear, alike

Source: Classifier-Free Guidance (2022)

Lecture outline

01 Choosing the path

02 Flow matching

03 Paths, steps and reflow

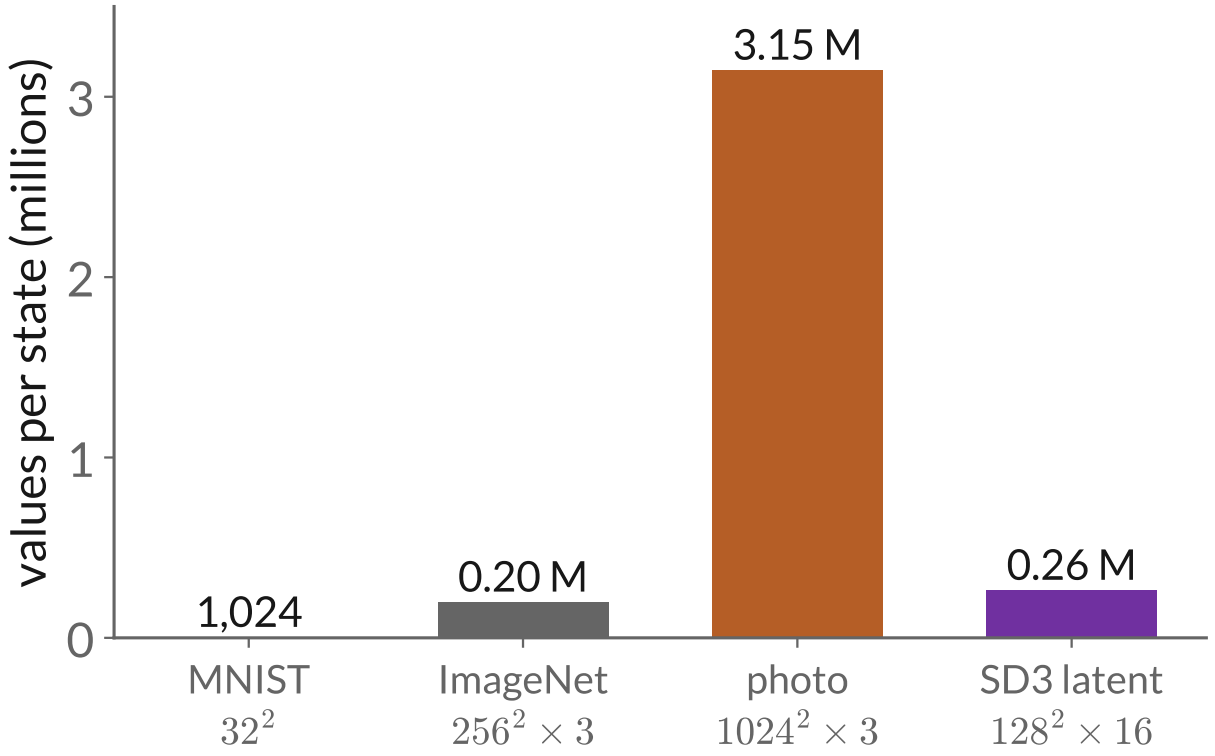
04 Conditional generation

05 Classifier-free guidance

06 Generating large images

07 A text-to-image system and its uses

Every step processes every value



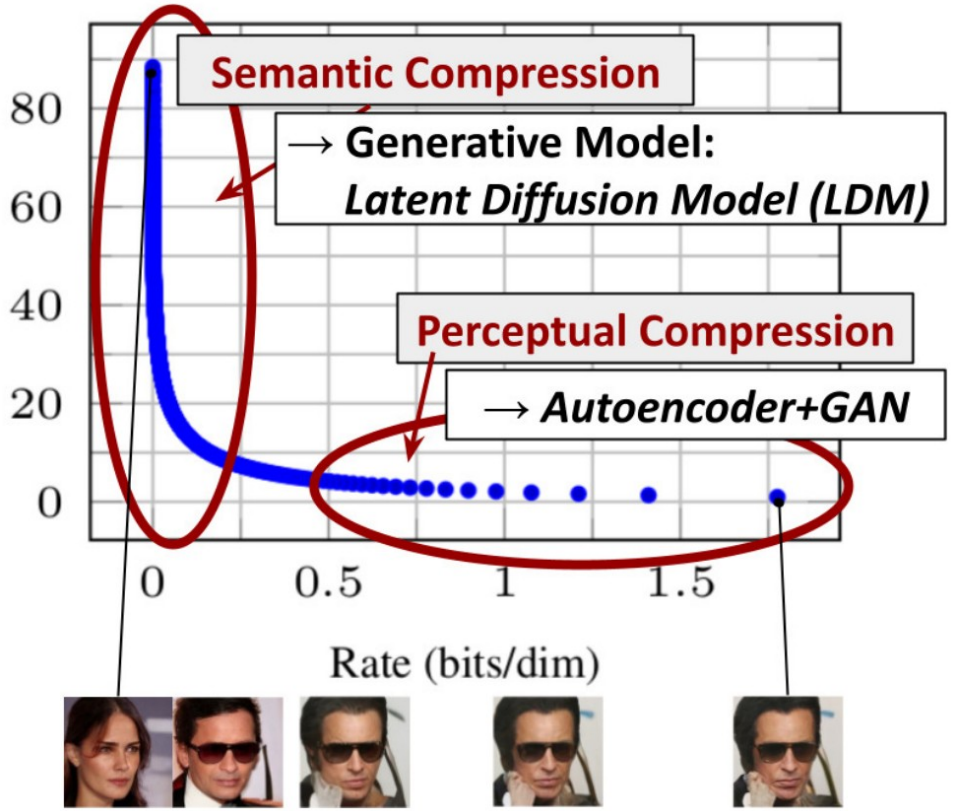
1024×1024 color image: 3.1 M values

Every step processes all of them, twice with guidance

Attention: cost grows with the square of positions

Idea: a smaller space for the flow

Most bits are imperceptible detail



Vertical: distortion. Horizontal: bits kept

Steep part: few bits carry the content

Long tail: many bits for invisible detail

Autoencoder for detail, flow for content

Source: Latent Diffusion (2022)

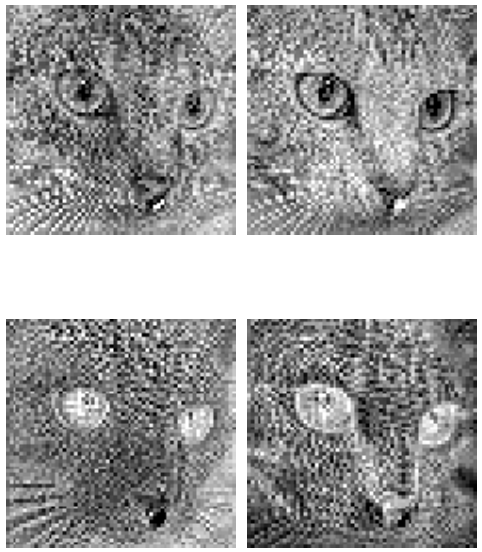
Compress once with an autoencoder

image $512 \times 512 \times 3$



E

latent $64 \times 64 \times 4$



D

decoded



E: image to latent z, 48× fewer values. D: back to an image

31.8 dB PSNR: differences hard to see

Trained once with reconstruction and GAN losses (L15, L16), then frozen

Source: autoencoder: Stable Diffusion sd-vae-ft-mse · Latent Diffusion (2022) · photo: scikit-image (CC0)

The decoder restores fine detail

original (zoom)



decoded (zoom)



128×128 crop around the eye

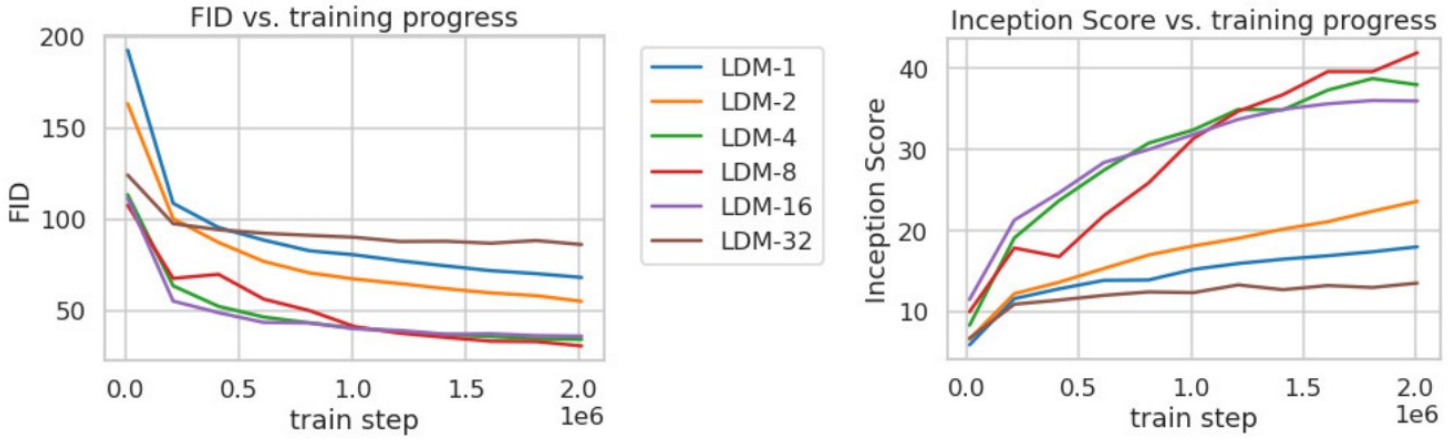
Latent: only 16×16 positions

Fur and whiskers come from the decoder

Details differ but look plausible

Source: autoencoder: Stable Diffusion sd-vae-ft-mse · photo: scikit-image (CC0)

How much to compress



Training on ImageNet; LDM-f: latent f times smaller per side

$f = 1$: slow training

$f = 32$: too much lost

$f = 4$ **to** 16 **work best**

SD3 autoencoders, $f = 8$, reconstruction quality

latent channels	4	8	16
FID (lower is better)	2.41	1.56	1.06
PSNR in dB (higher is better)	25.12	26.40	28.62

More channels, more detail

SD3: 16 channels, larger flow model

Source: Latent Diffusion (2022) · Stable Diffusion 3 (2024)

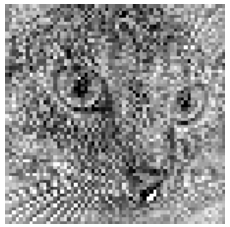
Flow matching in the latent space

Training

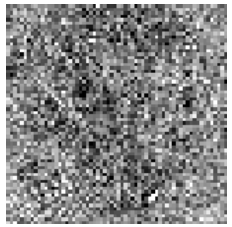


image x

E



$z_0 = E(x)$

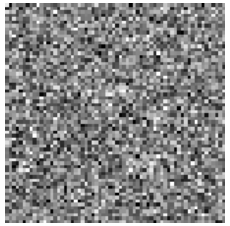


$z_t = (1-t)\, z_0 + t\, \epsilon$

v_θ

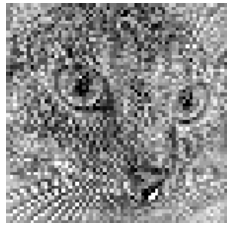
$$\left\| v_\theta(z_t, t, c) - (\epsilon - z_0) \right\|^2$$

Generation



$z_1 \sim \mathcal{N}(0, I)$

Euler steps with v_θ



z_0

D



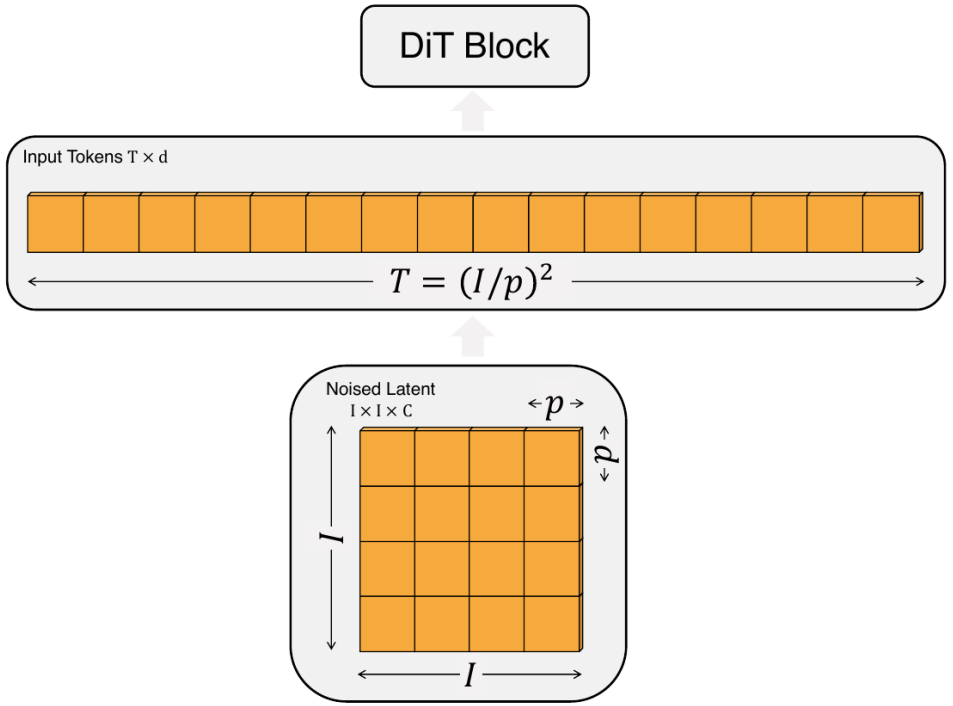
image

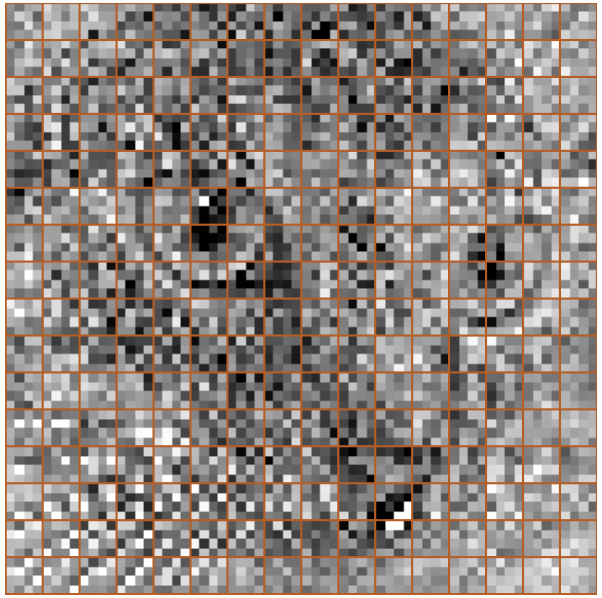
On z: same loss and sampler

E, D frozen: the flow never sees pixels

Source: photo: scikit-image (CC0)

A Transformer as the velocity network





one latent channel; patches of 4×4 positions (256 tokens)

$p \times p$ patches become tokens, as in ViT (L12)

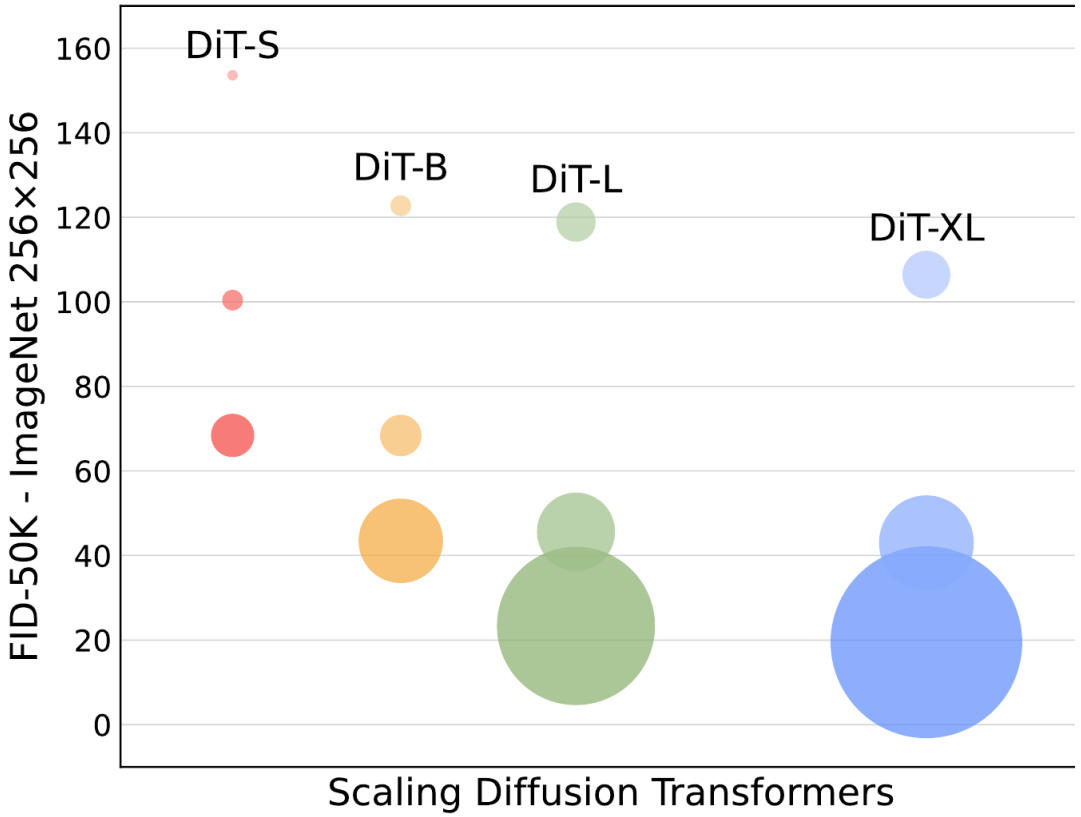
Transformer blocks (L12): all tokens attend

Linear layer: token back to patch, the velocity

Why: quality grows steadily with compute (next slide)

Source: DiT (2023) · autoencoder: Stable Diffusion sd-vae-ft-mse

Bigger Transformers, better images



FID on ImageNet 256×256 ; lower is better

S, B, L, XL: wider, deeper Transformers

Bubble area: compute per call

More compute per step, better images

Source: DiT (2023)

Time and label by scale and shift

$$h \leftarrow h + \alpha \cdot \mathrm{Attn}\big((1+\gamma)\cdot \mathrm{LN}(h) + \beta\big)$$

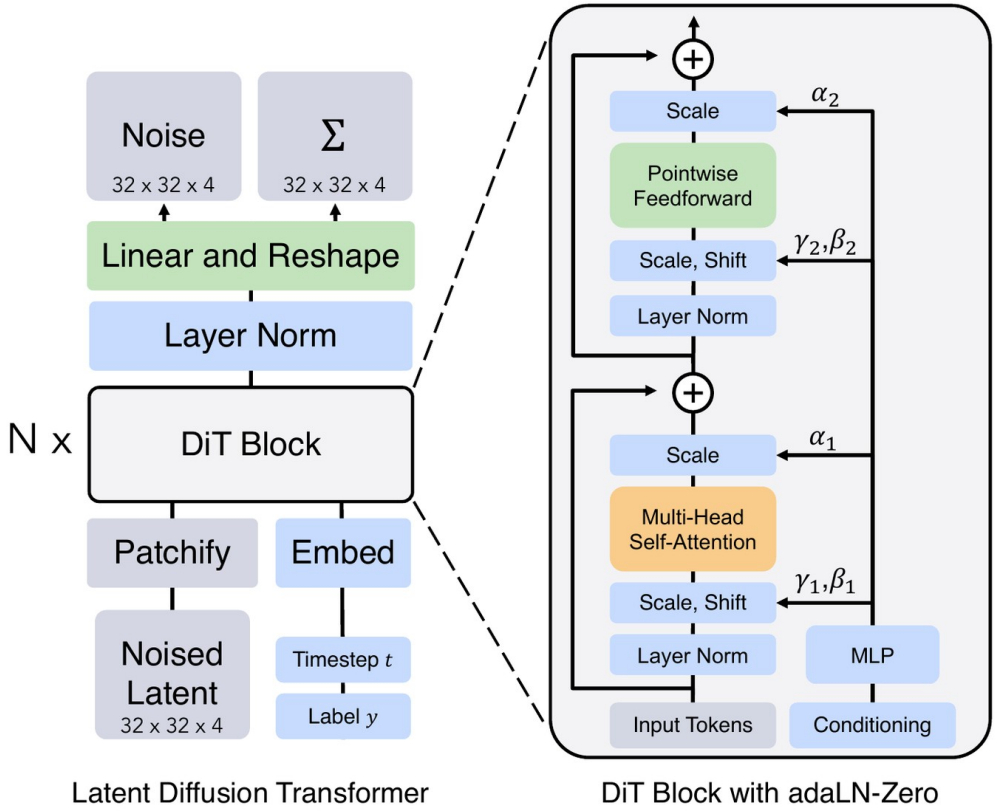
e: embedding of t and label, as in the U-Net

MLP: e to scale γ, shift β, gate α per channel

LN: layer normalization; same rule around the MLP

Gate starts at 0: each block starts as the identity

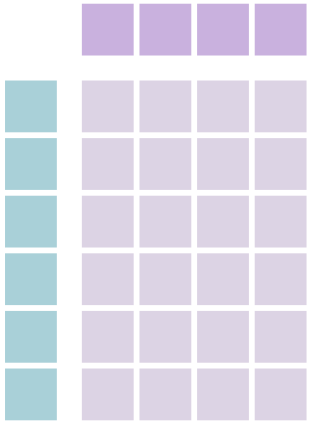
Sentences enter through attention instead



Source: DiT (2023)

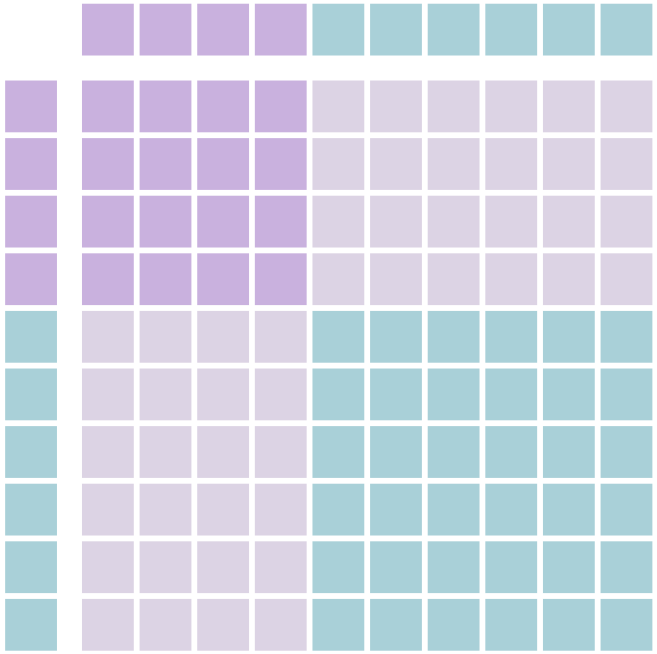
Text and image tokens in one attention

cross-attention



rows: image tokens ask; columns: text tokens answer

joint attention (SD3)



all tokens ask and answer

text token

image token

Cross: image reads text; text fixed

Joint: one attention over both sequences

Separate weights per token type

Lecture outline

01 Choosing the path

02 Flow matching

03 Paths, steps and reflow

04 Conditional generation

05 Classifier-free guidance

06 Generating large images

07 A text-to-image system and its uses

Training Stable Diffusion 3



E (frozen)

z_0, $128 \times 128 \times 16$

$z_t = (1-t)\, z_0 + t\, \epsilon$

"a cat in a hat"

CLIP-L → token vectors

CLIP-G

T5-XXL → pooled vector and t

frozen

MM-DiT v_θ

image and text tokens: joint attention

pooled + t: scale and shift

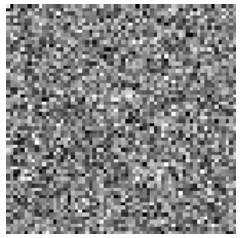
$\left\| v_\theta - (\epsilon - z_0) \right\|^2$

Each text encoder dropped 46.3% of the time: trains the unconditional velocity

Middle times t, where the target is hardest to guess, sampled more often (logit-normal)

Source: photo: scikit-image (CC0)

Generating with Stable Diffusion 3



$z_1 \sim \mathcal{N}(0, I)$

repeat for N steps

$$\tilde{v} = v_\theta(z, t, \varnothing) + w\left(v_\theta(z, t, c) - v_\theta(z, t, \varnothing)\right)$$

$$z \leftarrow z - \tfrac{1}{N}\,\tilde{v}$$

z_0 → D →



1024 × 1024

prompt c and the empty prompt $\varnothing$: encoded once, reused in every step

Cost: N steps, two calls of a large Transformer each

T5 can be skipped: most prompts still work; written text suffers

Source: Stable Diffusion 3 (2024)

Larger models need fewer steps

SD3: drop in text–image agreement (CLIP score) relative to 50 steps

model depth	5 steps	10 steps	20 steps	path length
15	4.30%	0.86%	0.21%	191.13
30	3.59%	0.70%	0.24%	187.96
38	**2.71%**	**0.14%**	**0.08%**	**185.96**

Deeper models lose less at few steps

Path length: distance travelled; shrinks with size

Authors: larger models fit straight paths better

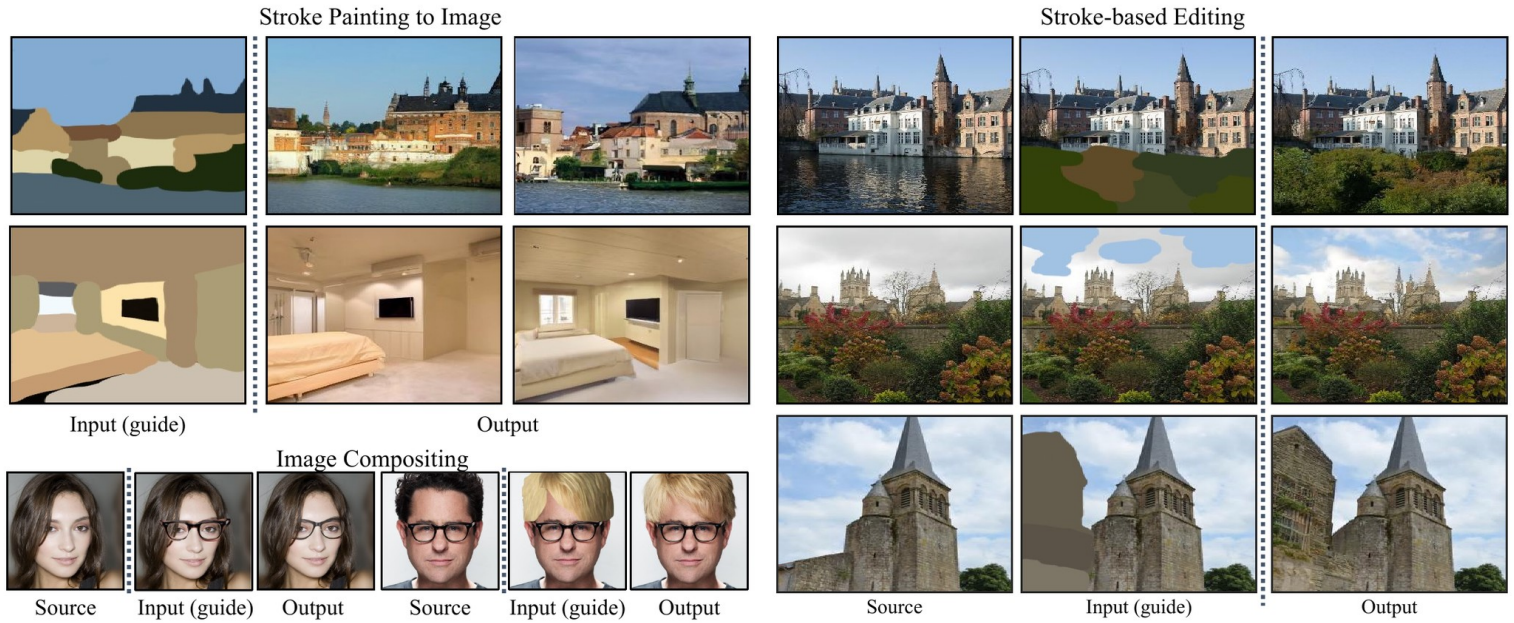
Source: Stable Diffusion 3 (2024)

Editing: start part-way along the path

$$x_{t_0} = (1 - t_0)\, x_{\text{guide}} + t_0\, \epsilon$$

start time

strokes or a photo



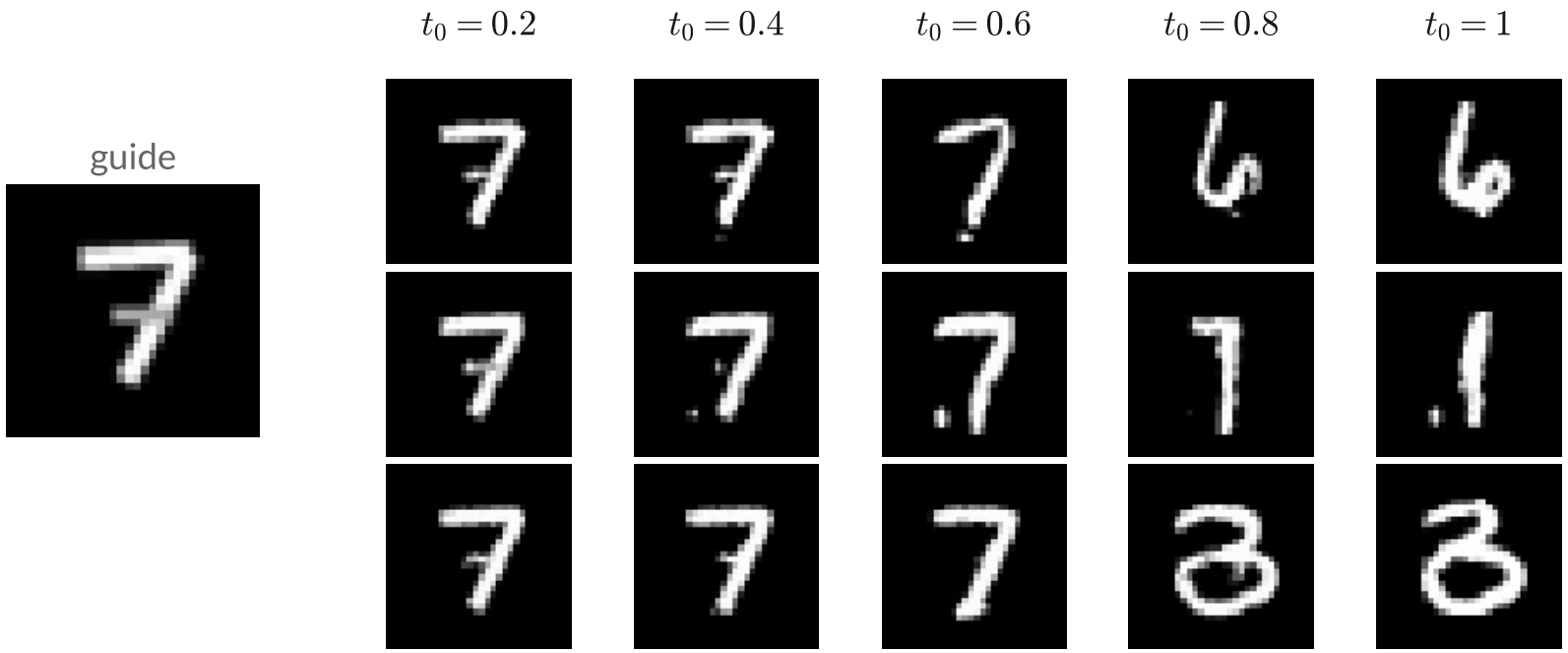
Start at t_0 : the guide plus noise

Euler steps from t_0 down to 0

No retraining

Source: SDEdit (2022)

Choosing the starting time



t_0	0.2	0.4	0.6	0.8	1
pixel distance to guide	0.02	0.04	0.08	0.18	0.24
read as a 7 (of 6 samples)	6	6	4	2	0

Guide: a rough “7”; no label

Small t_0 : keeps the strokes, crossbar included

$t_0 = 1$: guide forgotten

Larger t_0 : more freedom, less faithful

Adding a spatial condition

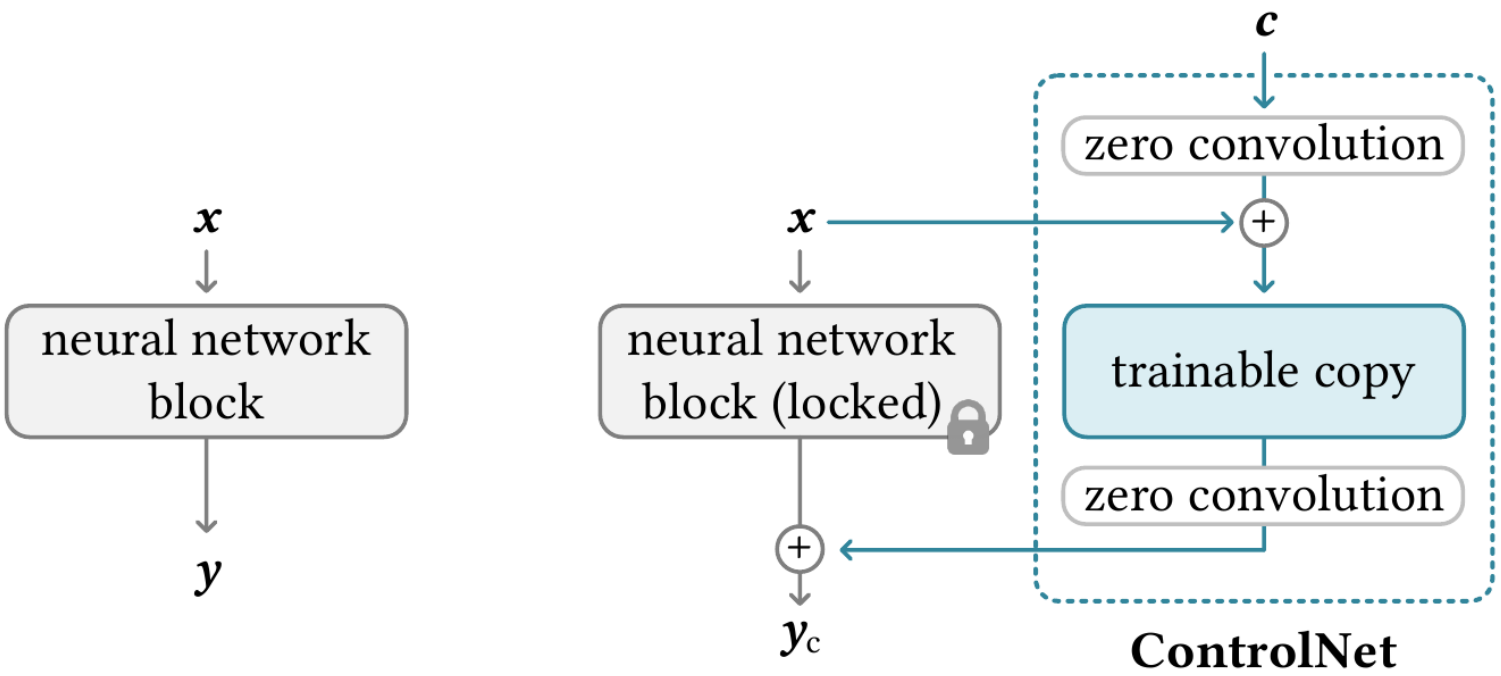


Condition: edges (top) or a pose (bottom), plus a prompt

Layout from the condition, content from the prompt

Source: ControlNet (2023)

ControlNet: a trainable copy



Lock the pretrained network

c enters a trainable copy of its encoder

Zero convolutions start at 0: training starts from the original

Control without retraining the base

Source: ControlNet (2023)

From images to video



Frames from two generated videos

Video latent with a time axis; space-time patches

Same recipe: autoencoder, flow matching, Transformer, guidance

Source: Seedance 1.0 (2025)

Summary

Few steps. Train on a straight path for each pair, $x_t = (1-t)x_0 + t\epsilon$, regressing on $\epsilon - x_0$; the network learns the average velocity

Sampling. Euler steps from noise. The averaged path can bend, but its even speed needs fewer steps than DDIM; reflow straightens it

Following the prompt. c as one more input: a label by embedding, a sentence by attention

Guidance. Bayes: a classifier's push is a difference of two scores, so $\tilde{v} = v_u + w\,(v_c - v_u)$; more agreement, less variety

Large images. Flow in an autoencoder's latent space; a Transformer predicts the velocity

Uses. Text-to-image (SD3), editing, spatial control, video

Next time: value learning and DQN

A new setting: an agent acts, and the world answers with rewards

Goal: learn how good each action is in each situation, a value

Deep Q-networks: a network predicts these values from raw input