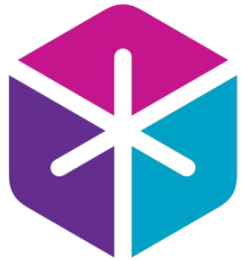
Flow and Diffusion Models - 1

Advances in Computer Vision

MIT Spring 2025 | April 03 2025

Peter Holderrieth



Generative AI - A new generation of AI systems

Artistic Images

Realistic Videos

ChatGPT

Draft Texts

These systems are “creative”: they generate new objects.

Flow and Diffusion Models: state-of-the-art models for generating images and videos!

Stable Diffusion

DALEE

OpenAI Sora

Meta MovieGen

Next 2 classes: Image and Video Generation with Flow and Diffusion Models

Example Models

Image generation:

Stable Diffusion 3

Video generation:

Meta MovieGen

Esser, Patrick, et al. "Scaling rectified flow transformers for high-resolution image synthesis." *Forty-first international conference on machine learning*. 2024.

Movie Gen: A Cast of Media Foundation Models. https://ai.meta.com/static-resource/movie-gen-research-paper

Overview - Today + Next Tuesday

1. **From Generation to Sampling**: Formalize “generating an image/etc.”
2. **Flow and diffusion models as generative models:**
   a. *Flows*: Sampling based on ODEs
   b. *Diffusion*: Sampling based on SDEs
3. **Training algorithms - 1**: Flow Matching — Today

---

4. **Training algorithms - 2**: Score Matching — Next class
5. **Guidance**: How to condition on a prompt
6. **Neural network architectures**
7. **Case studies of large-scale models**

Lecture Notes

For your reference, there are lecture notes:
https://diffusion.csail.mit.edu/docs/lecture-notes.pdf

Note that these are bit *more in-depth than required* for this class.

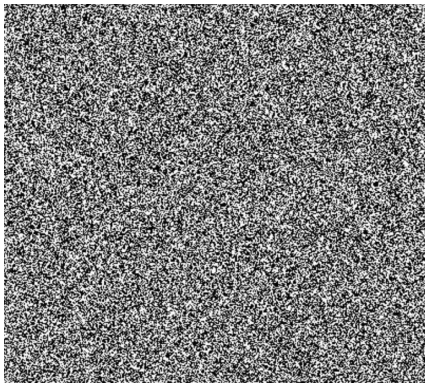
Section 1:

From Generation to Sampling

Goal: Formalize what it means to “generate an image”.

What does it mean to successfully generate an image?

Prompt: “A picture of a dog”

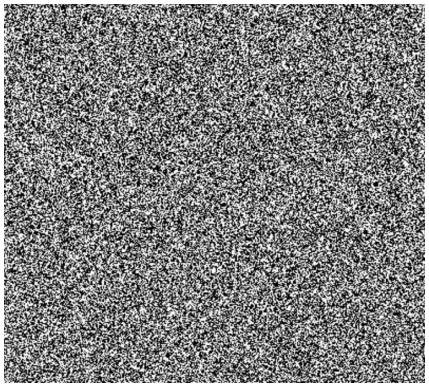



Useless < **Bad** < **Wrong animal** < **Great!**

These are subjective statements - Can we formalize this?

Data Distribution: How “likely” are we to find this picture in the internet with this caption?

Prompt: “A picture of a dog”




Impossible < Rare < Unlikely < Very likely

How good an image is ~= How likely it is under the data distribution

Generation as sampling from the data distribution

Data distribution: Distribution of objects that we want to generate: p_{data}

Probability density:

$$p_{\text{data}} : \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0},$$

$$z \mapsto p_{\text{data}}(z)$$

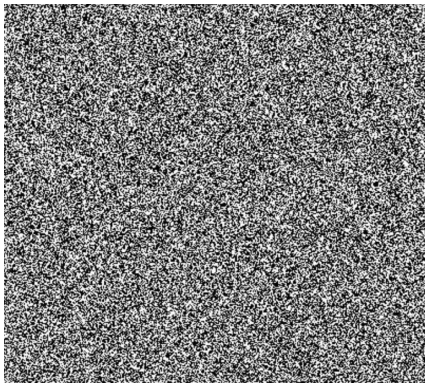
Note: We don't know the probability density!

Generation means sampling the data distribution:

$$z \sim p_{\text{data}} \Rightarrow z =$$

Data Distribution: How “likely” are we to find this picture in the internet?

Prompt: “A picture of a dog”




Small value of $p_{\text{data}}(z)$ → **High value** $p_{\text{data}}(z)$

Generative Models generate samples from data distribution

Initial distribution: p_{init} ***Default:*** $p_{\text{init}} = \mathcal{N}(0, I_d)$

A generative model converts samples from a initial distribution (e.g. Gaussian) into samples from the data distribution:

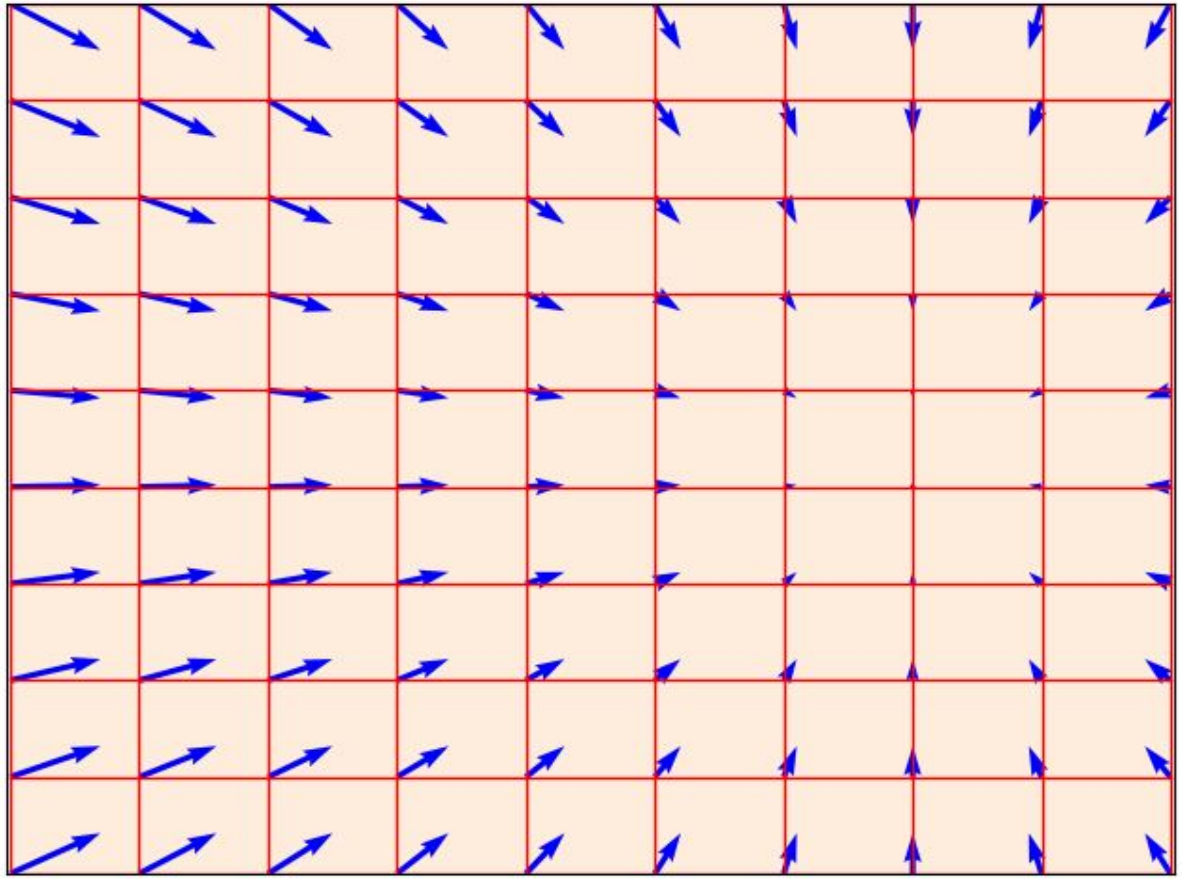
$x \sim p_{\text{init}}$ → Generative Model → $z \sim p_{\text{data}}$

Section 2:

Flow and Diffusion Models as Generative Models

Goal: Understand how to use ordinary and stochastic differential equations for generative models

Flow - Example



Numerical ODE simulation - Euler method

Algorithm 1 Simulating an ODE with the Euler method

Require: Vector field u_t, initial condition x_0, number of steps n

Set $t = 0$
Set step size $h = \frac{1}{n}$
Set $X_0 = x_0$
for $i = 1, \ldots, n-1$ **do**
    $X_{t+h} = X_t + hu_t(X_t)$ *Small step into direction of vector field*
    Update $t \leftarrow t + h$
end for
return $X_0, X_h, X_{2h}, \ldots, X_1$ *Return trajectory*

Toy example

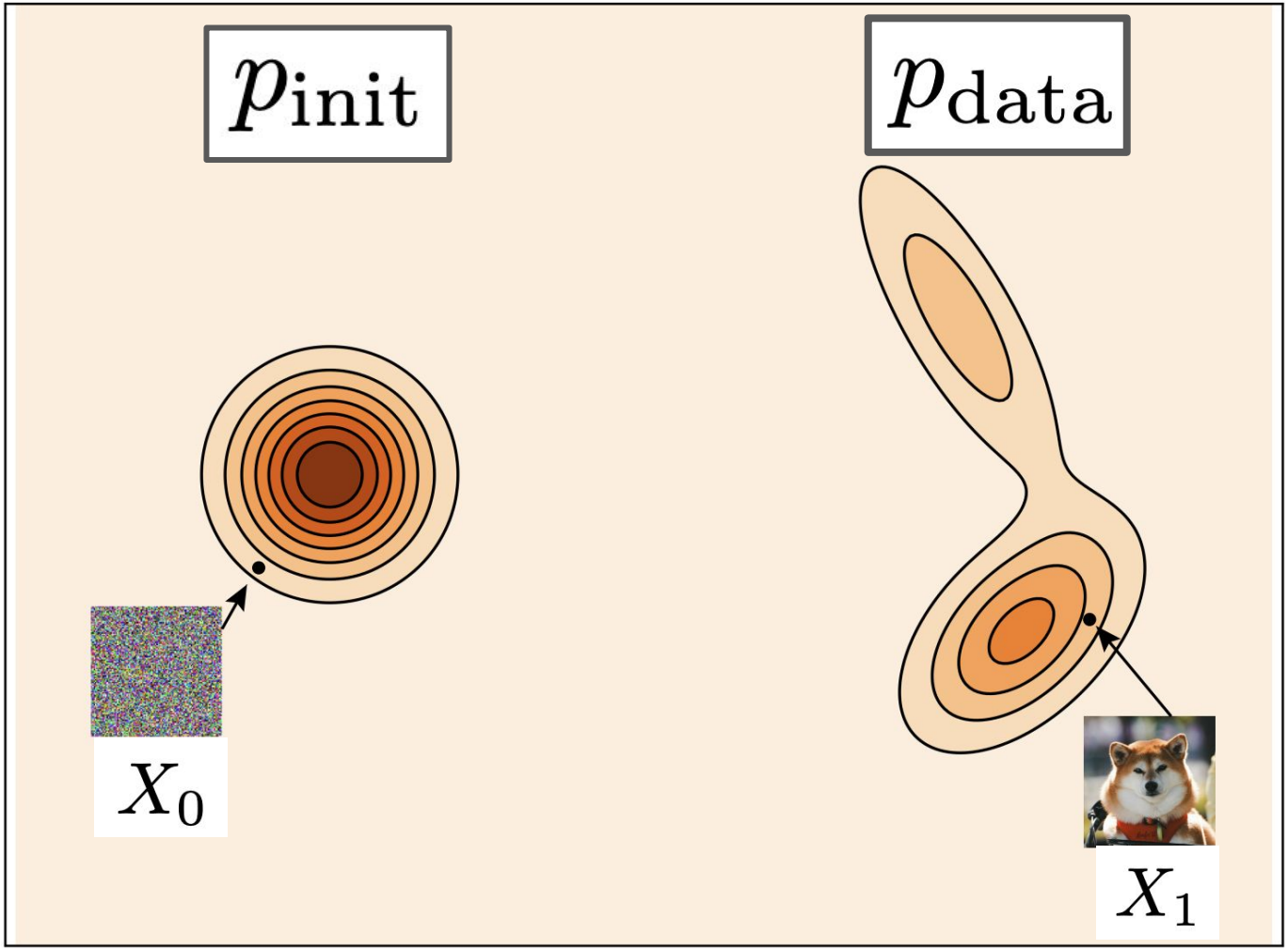


Figure credit: Yaron Lipman

Toy Flow Model

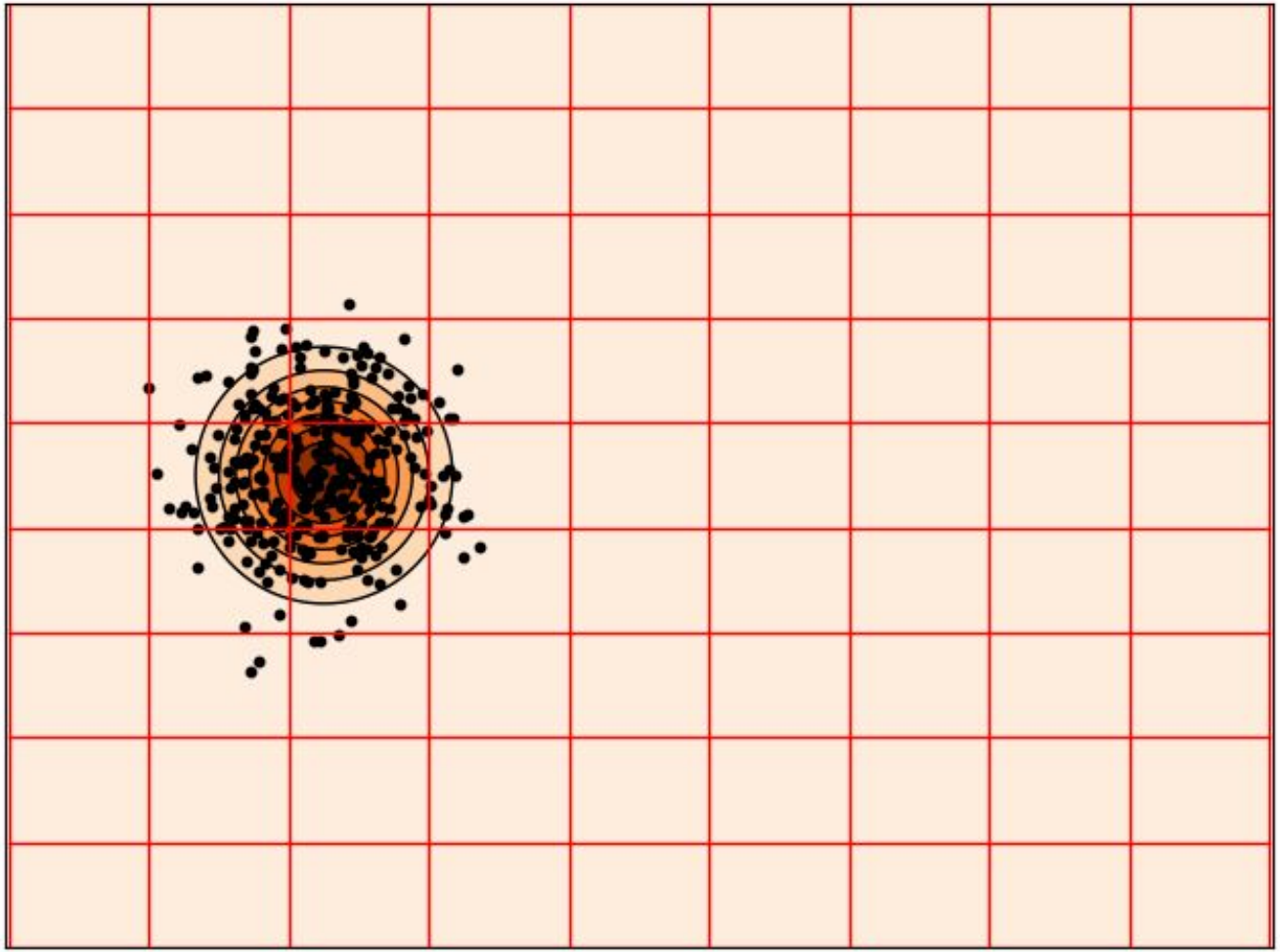


Figure credit: Yaron Lipman

How to generate objects with a Flow Model

Algorithm 1 Sampling from a Flow Model with Euler method

Require: Neural network vector field u_t^θ, number of steps n

1: Set $t = 0$
2: Set step size $h = \frac{1}{n}$
3: Draw a sample $X_0 \sim p_{\text{init}}$ *Random initialization!*
4: **for** $i = 1, \ldots, n-1$ **do**
5: $\quad X_{t+h} = X_t + h u_t^\theta(X_t)$
6: $\quad$ Update $t \leftarrow t + h$
7: **end for**
8: **return** X_1 *Return final point*

Examples generated with ODE simulation:

Image generation:

Stable Diffusion 3

Video generation:

Meta MovieGen

Algorithm 2 Sampling from a Diffusion Model (Euler-Maruyama method)

Require: Neural network u_t^θ, number of steps n, diffusion coefficient σ_t

1: Set $t = 0$
2: Set step size $h = \frac{1}{n}$
3: Draw a sample $X_0 \sim p_{\text{init}}$
4: **for** $i = 1, \ldots, n-1$ **do**
5: Draw a sample $\epsilon \sim \mathcal{N}(0, I_d)$
6: $X_{t+h} = X_t + h u_t^\theta(X_t) + \sigma_t \sqrt{h} \epsilon$
7: Update $t \leftarrow t + h$
8: **end for**
9: **return** X_1

Add additional noise scaled by diffusion coefficient

Summary: Flow and Diffusion Models

Flow Model

Initialize: $X_0 \sim p_{\text{init}},$

ODE: $\mathrm{d}X_t = u_t^\theta(X_t)\mathrm{d}t$

Gaussian

Neural network vector field

Diffusion coeff.

Diffusion Model

Initialize: $X_0 \sim p_{\text{init}},$

SDE: $\mathrm{d}X_t = u_t^\theta(X_t)\mathrm{d}t + \sigma_t \mathrm{d}W_t$

To get samples, simulate ODE/SDE from t=0 to t=1 and return X_1

Section 2:

Training Algorithms - 1: Flow matching

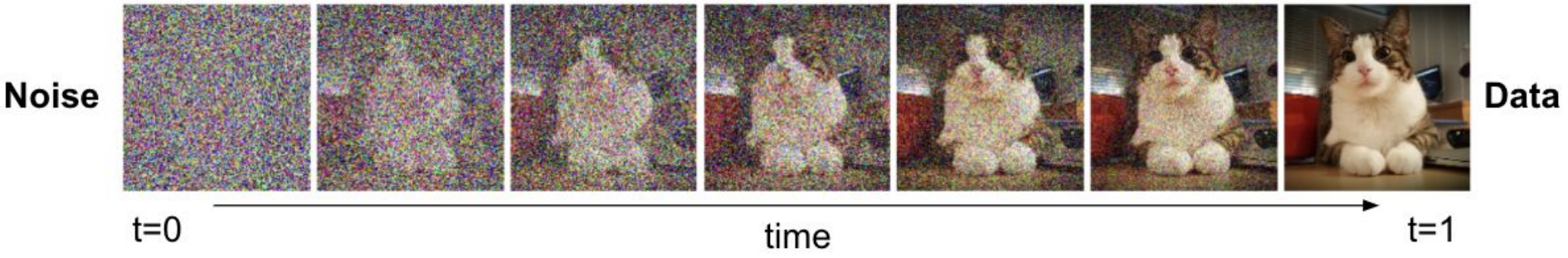
Goal: Train a flow model

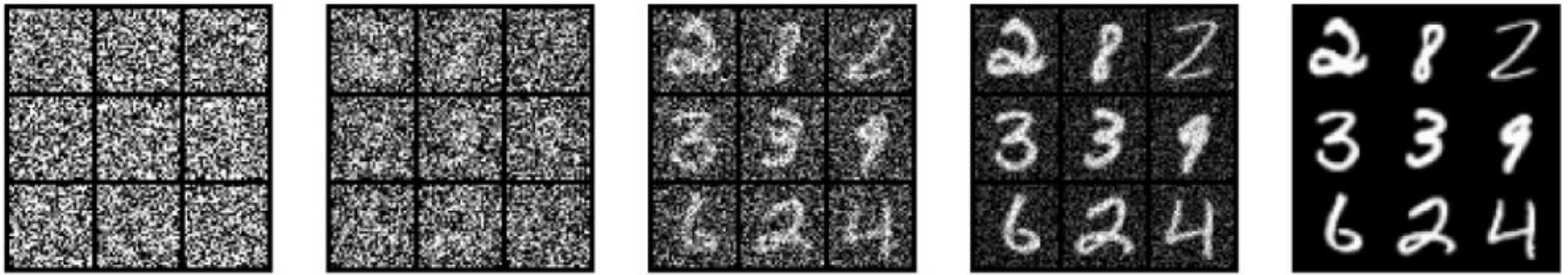
Key terminology:

“Conditional” = “Per single data point”

“Marginal” = “Across distribution of data points”

Probability Paths: The Path from Noise to Data

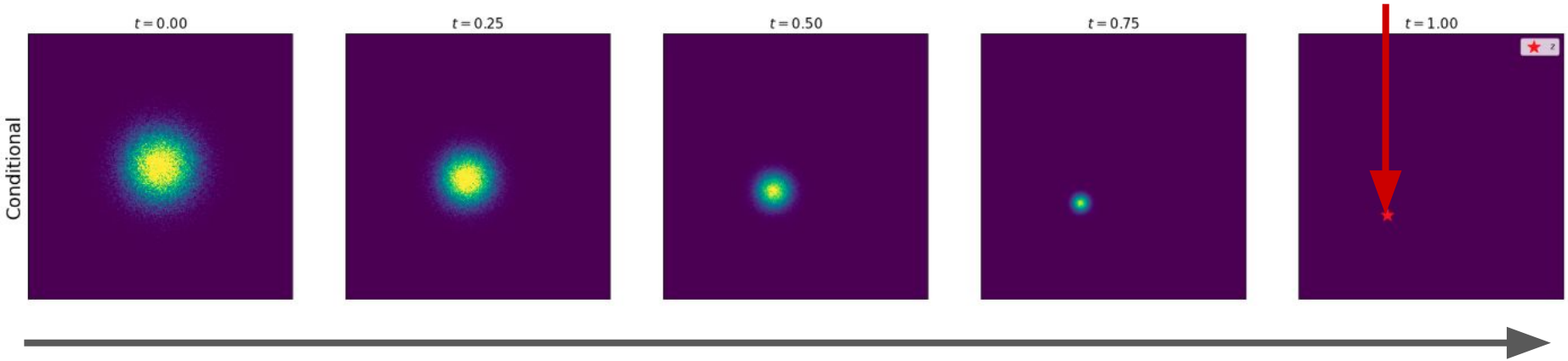




p_{init}

Conditional Probability Path $p_t(\cdot|z)$

z



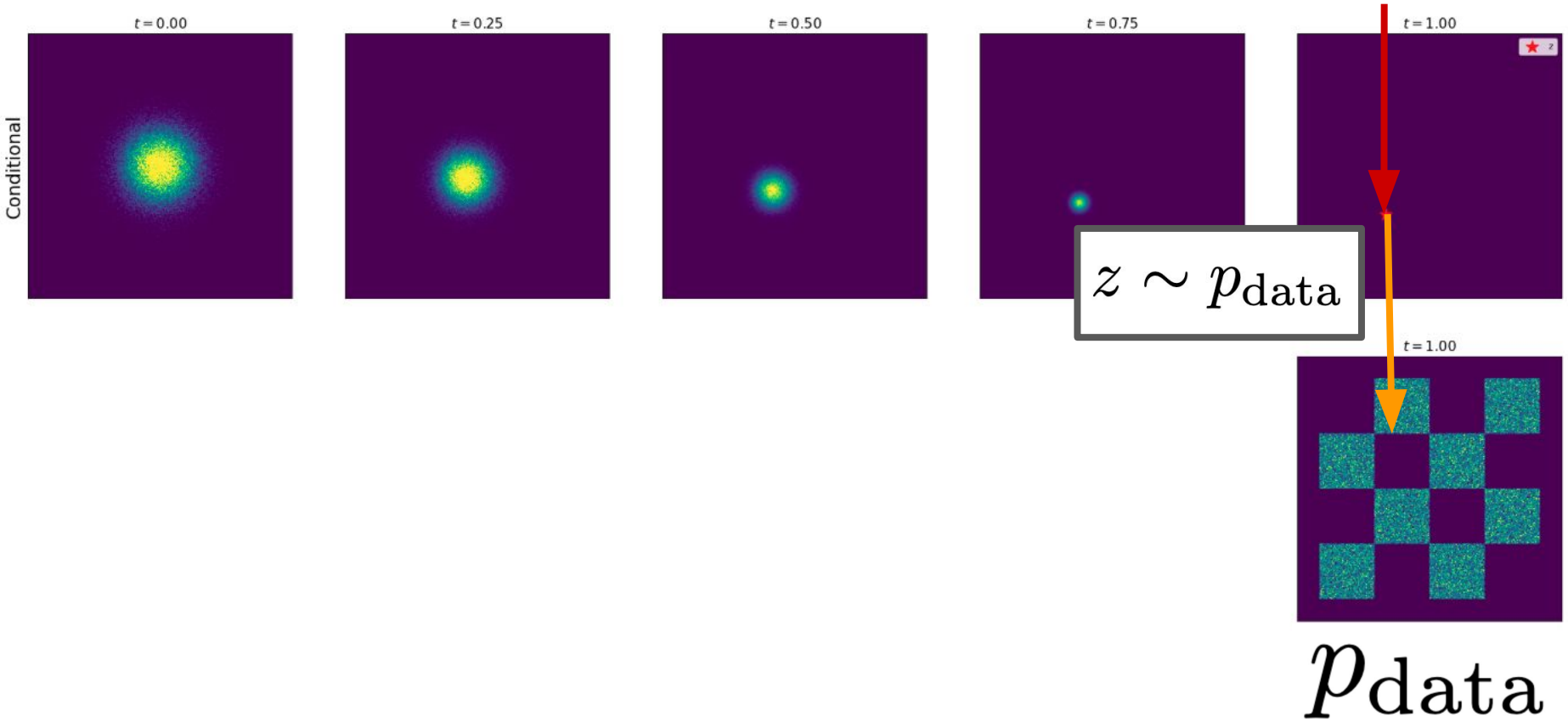
t=0

t=1

Conditional Probability Path $p_t(\cdot|z)$

p_{init}

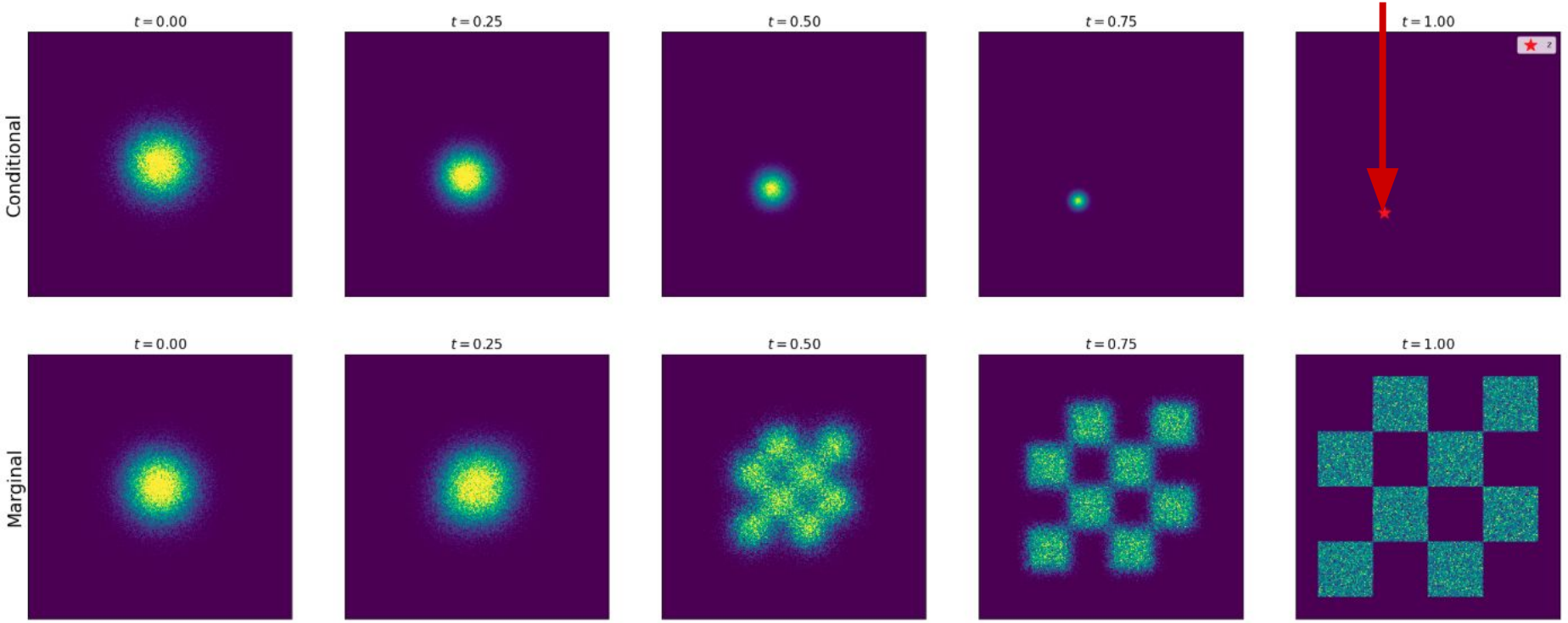
z



Conditional Probability Path $p_t(\cdot|z)$

p_{init}

z



p_{init}

Marginal Probability Path p_t

p_{data}

Simulating ODE with Conditional Vector Field for Conditional Probability Path

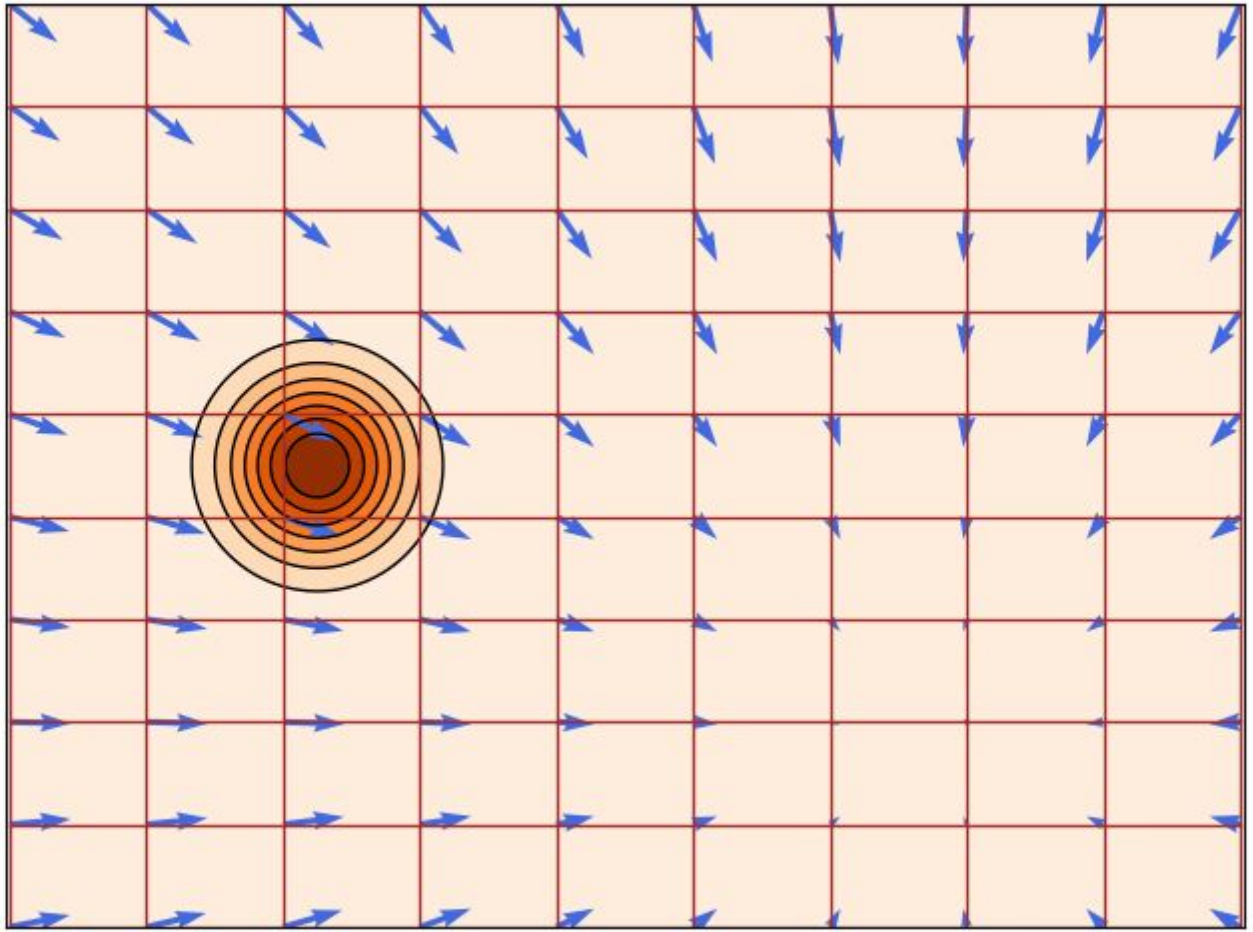


Figure credit: Yaron Lipman

Example marginal vector field - Meta MovieGen

These videos are generated by simulating the ODE with the (learnt) marginal vector field

Continuity Equation

Randomly initialized ODE

Given: $X_0 \sim p_{\text{init}}, \quad \frac{\mathrm{d}}{\mathrm{d}t} X_t = u_t(X_t)$

Follow probability path:

$$X_t \sim p_t \quad (0 \le t \le 1)$$

Marginals are p_t

equivalent

Continuity equation holds

$$\frac{\mathrm{d}}{\mathrm{d}t} p_t(x) = -\mathrm{div}(p_t u_t)(x)$$

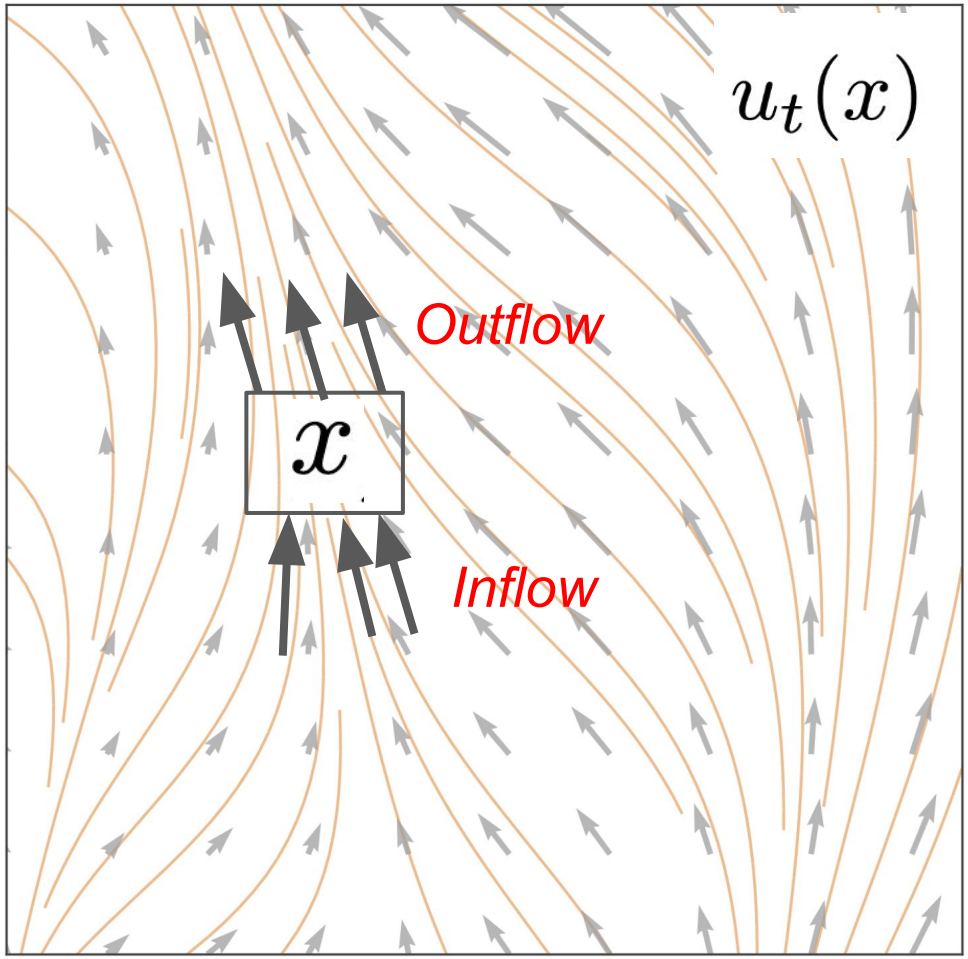
PDE holds

Continuity Equation

$$\frac{\mathrm{d}}{\mathrm{d}t} p_t(x) = -\mathrm{div}(p_t u_t)(x)$$

Change of probability mass at x

Outflow - inflow of probability mass from u



Algorithm 3 Flow Matching Training Procedure (General)

Require: A dataset of samples $z \sim p_{\text{data}}$, neural network u_t^θ

1: **for** each mini-batch of data **do**
2: Sample a data example z from the dataset.
3: Sample a random time $t \sim \text{Unif}_{[0,1]}$.
4: Sample $x \sim p_t(\cdot|z)$
5: Compute loss

$$\mathcal{L}(\theta) = \|u_t^\theta(x) - u_t^{\text{target}}(x|z)\|^2$$

6: Update the model parameters θ via gradient descent on $\mathcal{L}(\theta)$
7: **end for**

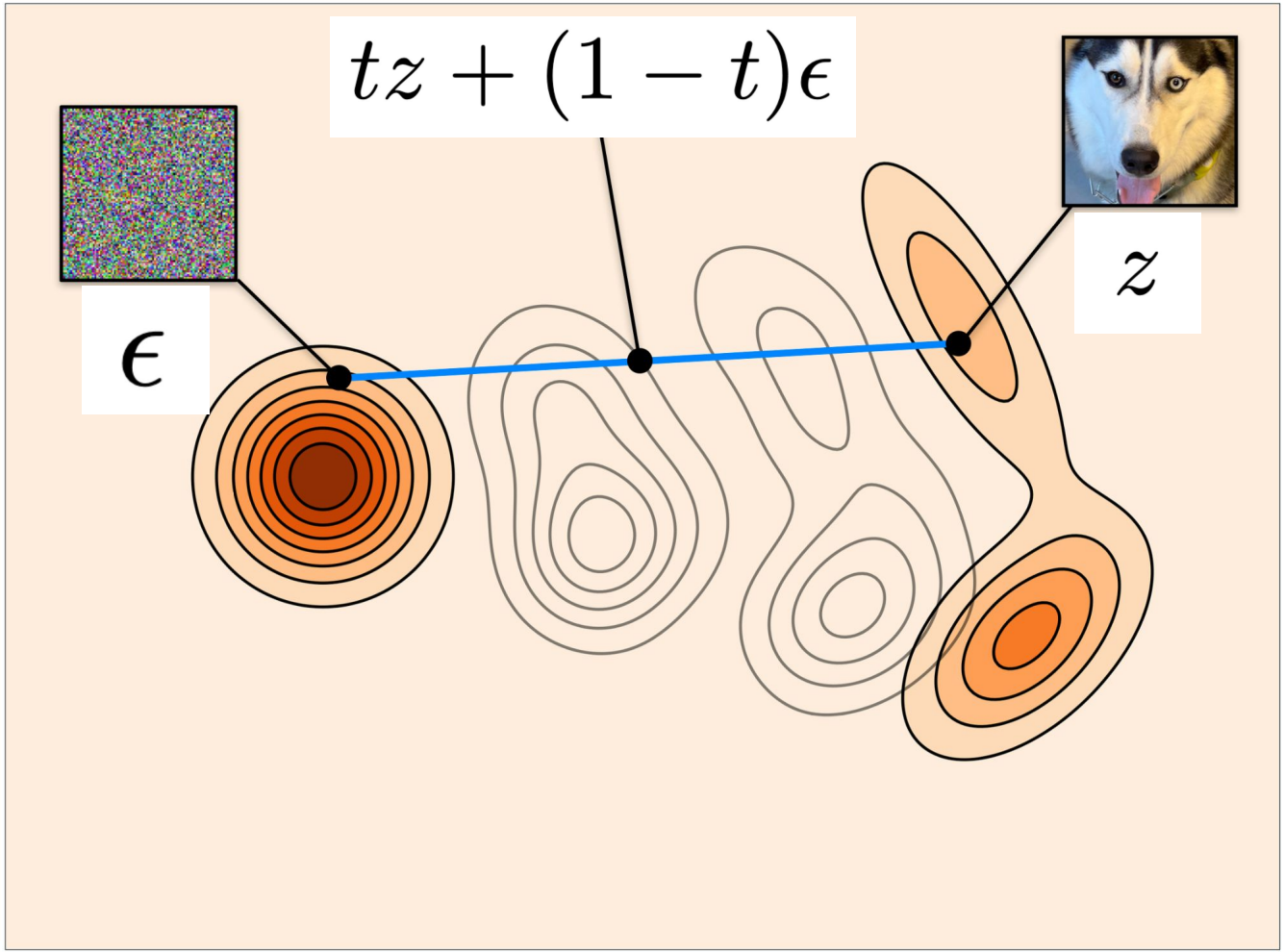


Figure credit: Yaron Lipman

Algorithm 4 Flow Matching Training for CondOT path

Require: A dataset of samples $z \sim p_{\text{data}}$, neural network u_t^θ

1: **for** each mini-batch of data **do**
2: Sample a data example z from the dataset.
3: Sample a random time $t \sim \text{Unif}_{[0,1]}$.
4: Sample noise $\epsilon \sim \mathcal{N}(0, I_d)$
5: Set $x = tz + (1-t)\epsilon$
6: Compute loss

$$\mathcal{L}(\theta) = \|u_t^\theta(x) - (z - \epsilon)\|^2$$

7: Update the model parameters θ via gradient descent on $\mathcal{L}(\theta)$.
8: **end for**

Example Flow Matching - Meta MovieGen

The neural network that generates these videos was trained with the algorithm in the previous slide

Example Flow Matching - Stable Diffusion 3

The neural network that generates these images was trained with the algorithm just shown

Outlook: Next class

1. **From Generation to Sampling**: Formalize “generating an image/etc.”
2. **Flow and diffusion models as generative models:**
   a. *Flows*: Sampling based on ODEs
   b. *Diffusion*: Sampling based on SDEs
3. **Training algorithms - 1**: Flow Matching — Today

---

4. **Training algorithms - 2**: Score Matching — Next class
5. **Guidance**: How to condition on a prompt
6. **Neural network architectures**
7. **Case studies of large-scale models**