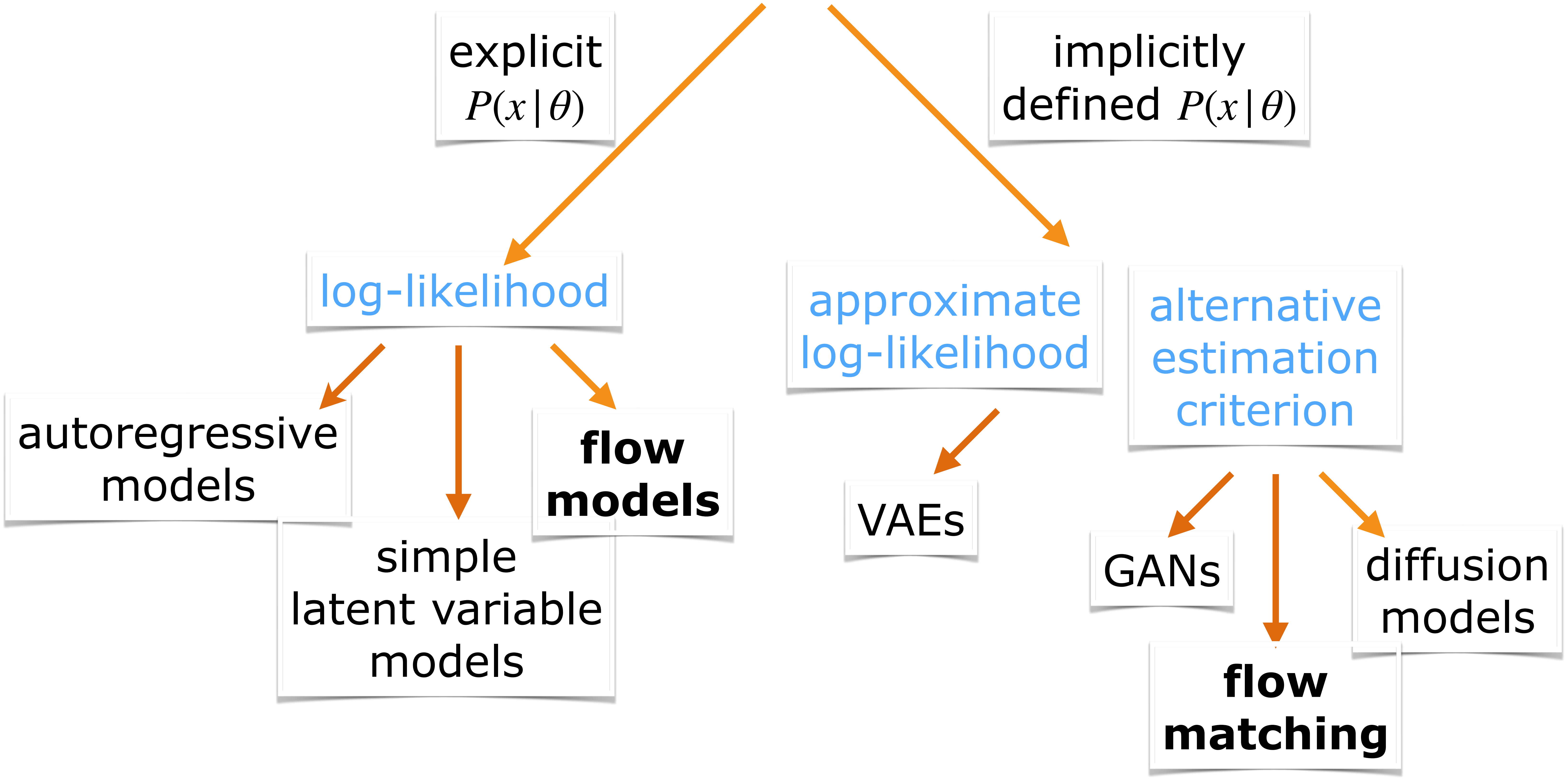
6.7900 Machine Learning (Fall 2024)

Lecture 23: generative models — flows

A slice of the generative “landscape”



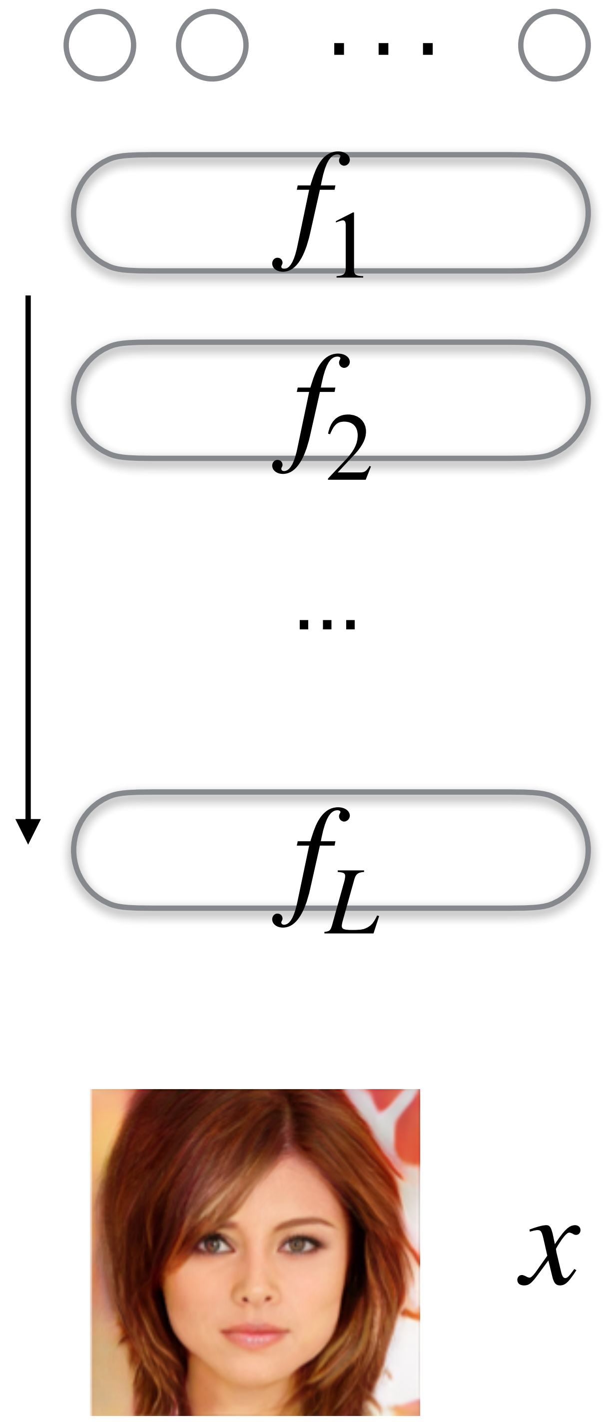
Warmup: normalizing flow

- We can transform simple latent randomization into a complex realization through a sequence of (always) invertible transformations

$$z \sim N(0, I)$$
$$x = f_L \circ f_{L-1} \circ \cdots \circ f_2 \circ f_1(z)$$

randomize

$z \sim N(0, I)$



$x = f(z; \theta)$

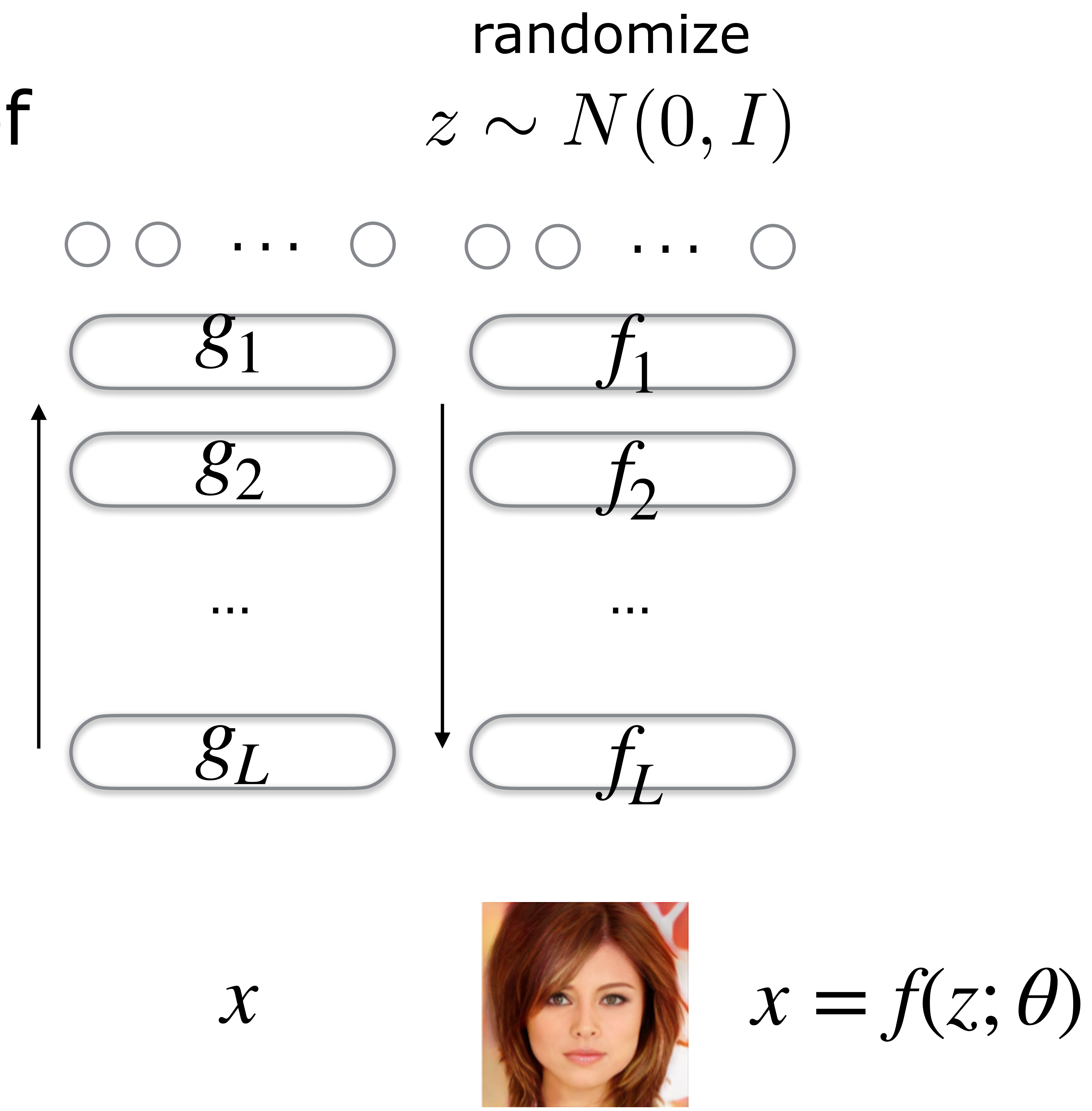
Warmup: normalizing flow

- We can transform simple latent randomization into a complex realization through a sequence of (always) invertible transformations

$$z \sim N(0, I)$$
$$x = f_L \circ f_{L-1} \circ \cdots \circ f_2 \circ f_1(z)$$

- This is advantageous since we can explicitly recover z and evaluate the log-likelihood of the observed x

$$z = g_1 \circ g_2 \circ \cdots \circ g_{L-1} \circ g_L(x), \;\; g_j = f_j^{-1}$$



Warmup: normalizing flow

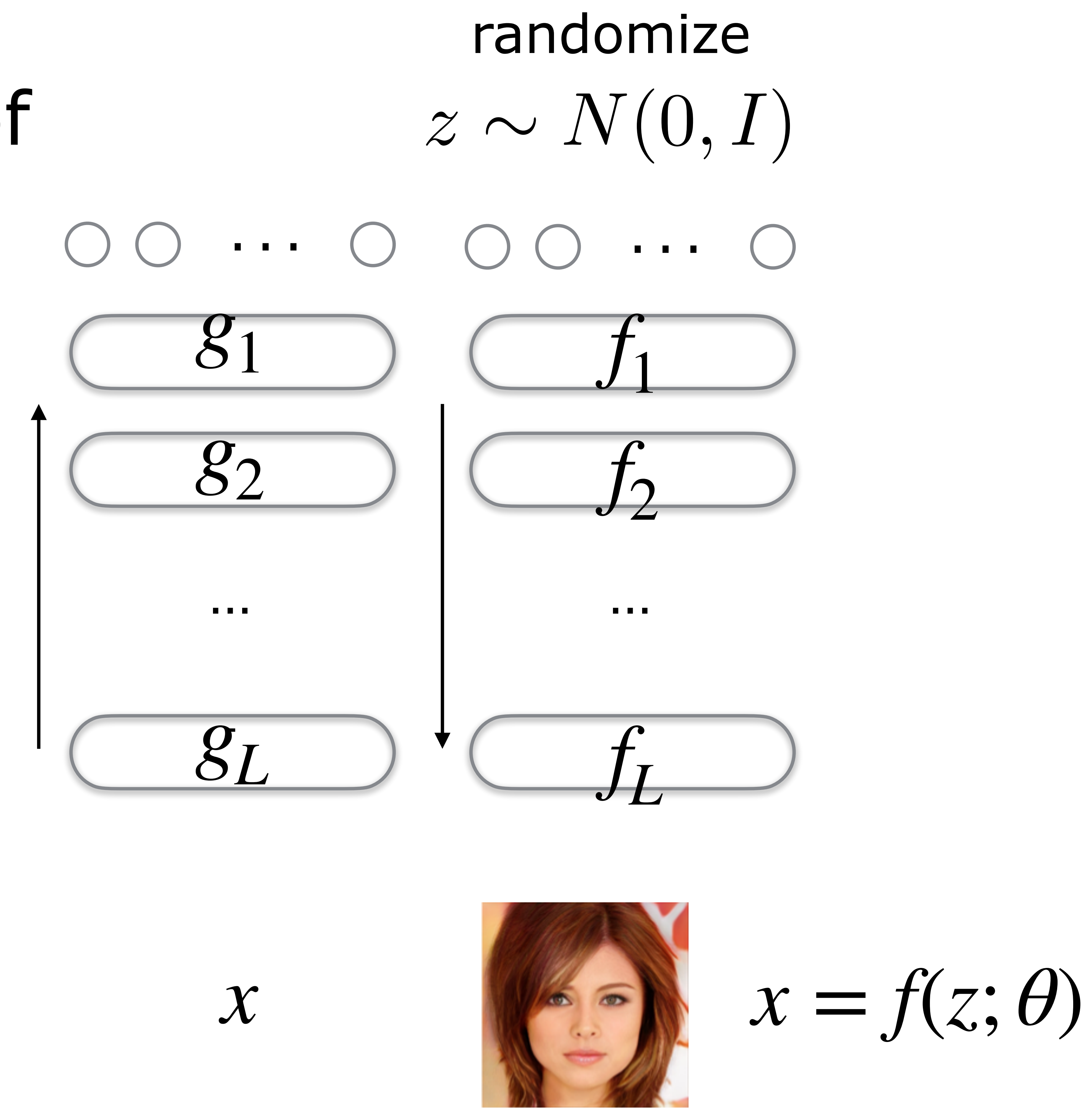
- We can transform simple latent randomization into a complex realization through a sequence of (always) invertible transformations

$$z \sim N(0, I)$$
$$x = f_L \circ f_{L-1} \circ \cdots \circ f_2 \circ f_1(z)$$

- This is advantageous since we can explicitly recover z and evaluate the log-likelihood of the observed x

$$z = g_1 \circ g_2 \circ \cdots \circ g_{L-1} \circ g_L(x), \quad g_j = f_j^{-1}$$

$$x = h_L \xrightarrow{g_L} h_{L-1} \xrightarrow{g_{L-1}} \ldots \xrightarrow{g_1} h_1 \rightarrow h_0 = z$$



Warmup: normalizing flow

- We can transform simple latent randomization into a complex realization through a sequence of (always) invertible transformations

$$z \sim N(0, I)$$
$$x = f_L \circ f_{L-1} \circ \cdots \circ f_2 \circ f_1(z)$$

- This is advantageous since we can explicitly recover z and evaluate the log-likelihood of the observed x

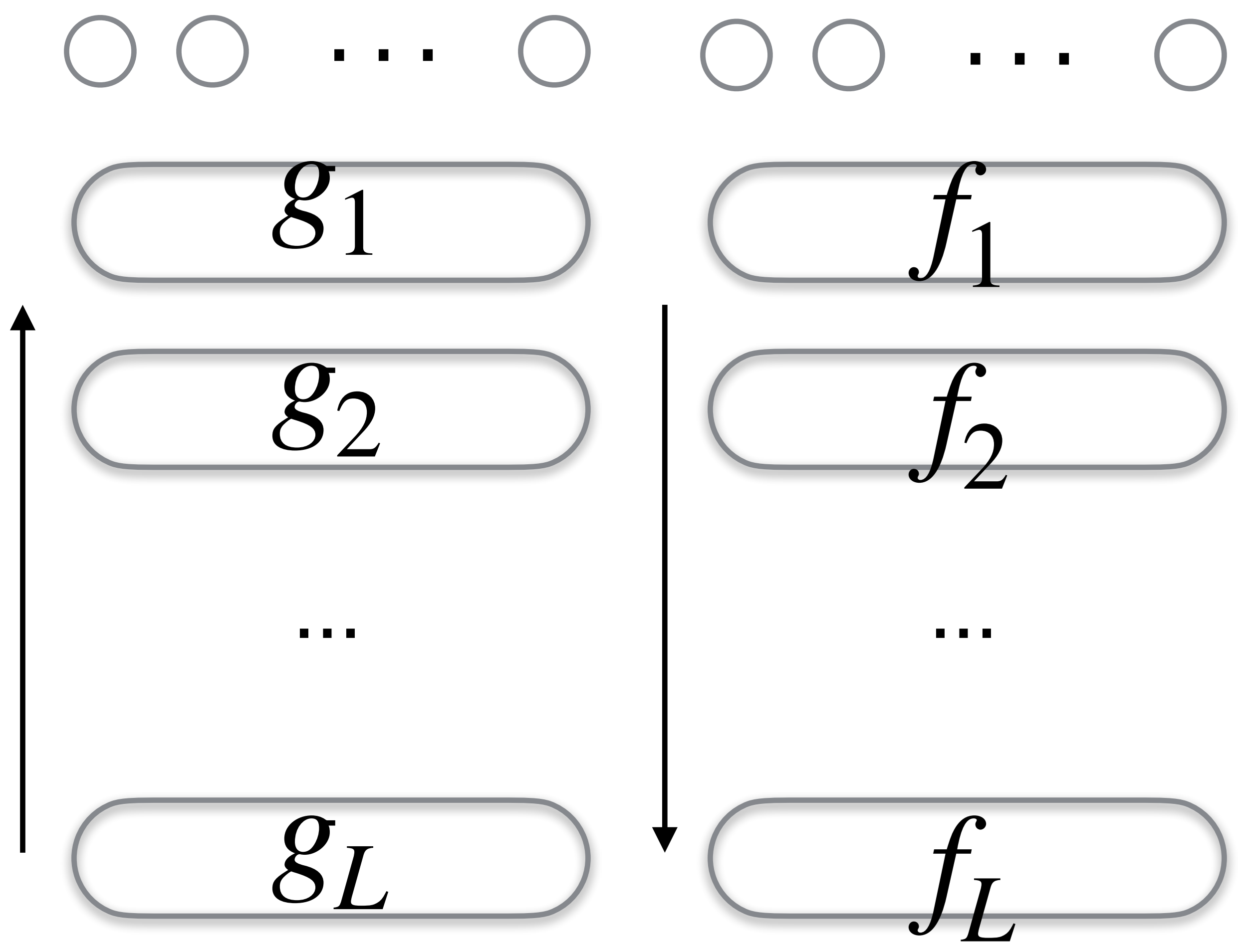
$$z = g_1 \circ g_2 \circ \cdots \circ g_{L-1} \circ g_L(x), \quad g_j = f_j^{-1}$$

$$\color{red}{x =} \; h_L \xrightarrow{g_L} h_{L-1} \xrightarrow{g_{L-1}} \ldots \xrightarrow{g_1} h_1 \rightarrow h_0 \; \color{red}{= z}$$

$$P(x;\theta) = N(z(x) \,|\, 0, I) \prod_{j=1}^{L} \left| \underbrace{\frac{\partial h_{j-1}}{\partial h_j}}_{\text{dxd}} \right|$$

randomize

$z \sim N(0, I)$



x

$x = f(z;\theta)$

Warmup: normalizing flow

- We can transform simple latent randomization into a complex realization through a sequence of (always) invertible transformations

$$z \sim N(0, I)$$
$$x = f_L \circ f_{L-1} \circ \cdots \circ f_2 \circ f_1(z)$$

- This is advantageous since we can explicitly recover z and evaluate the log-likelihood of the observed x

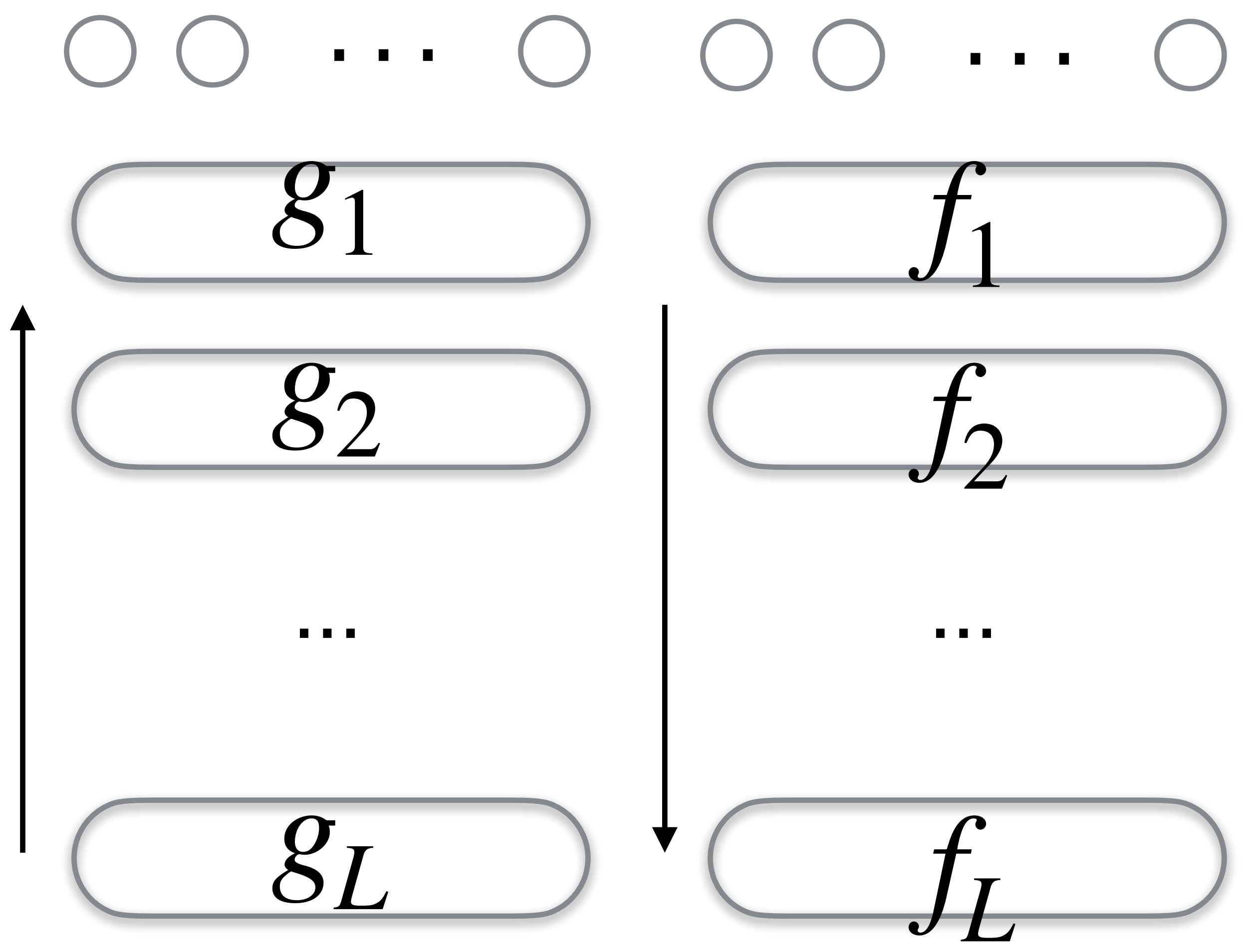
$$z = g_1 \circ g_2 \circ \cdots \circ g_{L-1} \circ g_L(x), \quad g_j = f_j^{-1}$$

$$\color{red}{x =} h_L \xrightarrow{g_L} h_{L-1} \xrightarrow{g_{L-1}} \ldots \xrightarrow{g_1} h_1 \to h_0 \color{red}{= z}$$

$$P(x;\theta) = N(z(x)\,|\,0,I) \prod_{j=1}^{L} \left| \underbrace{\frac{\partial h_{j-1}}{\partial h_j}}_{\text{dxd}} \right|$$

randomize

$z \sim N(0, I)$



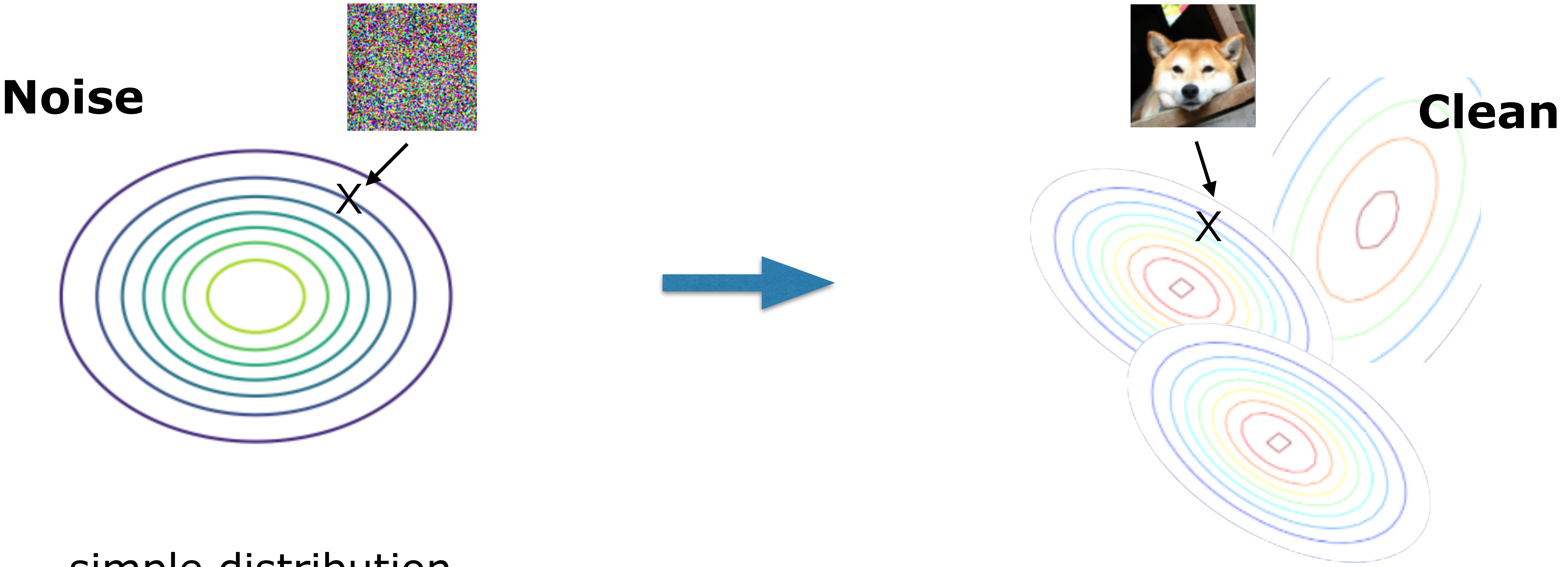
x

$x = f(z;\theta)$

But: challenging to realize complex models if each layer has to remain easily invertible!

Thinking about continuous flows

- We are interested in modeling how samples from a simple distribution $p_0(x)$ can be transported into samples from a complex distribution $p_1(x)$ (data distribution)

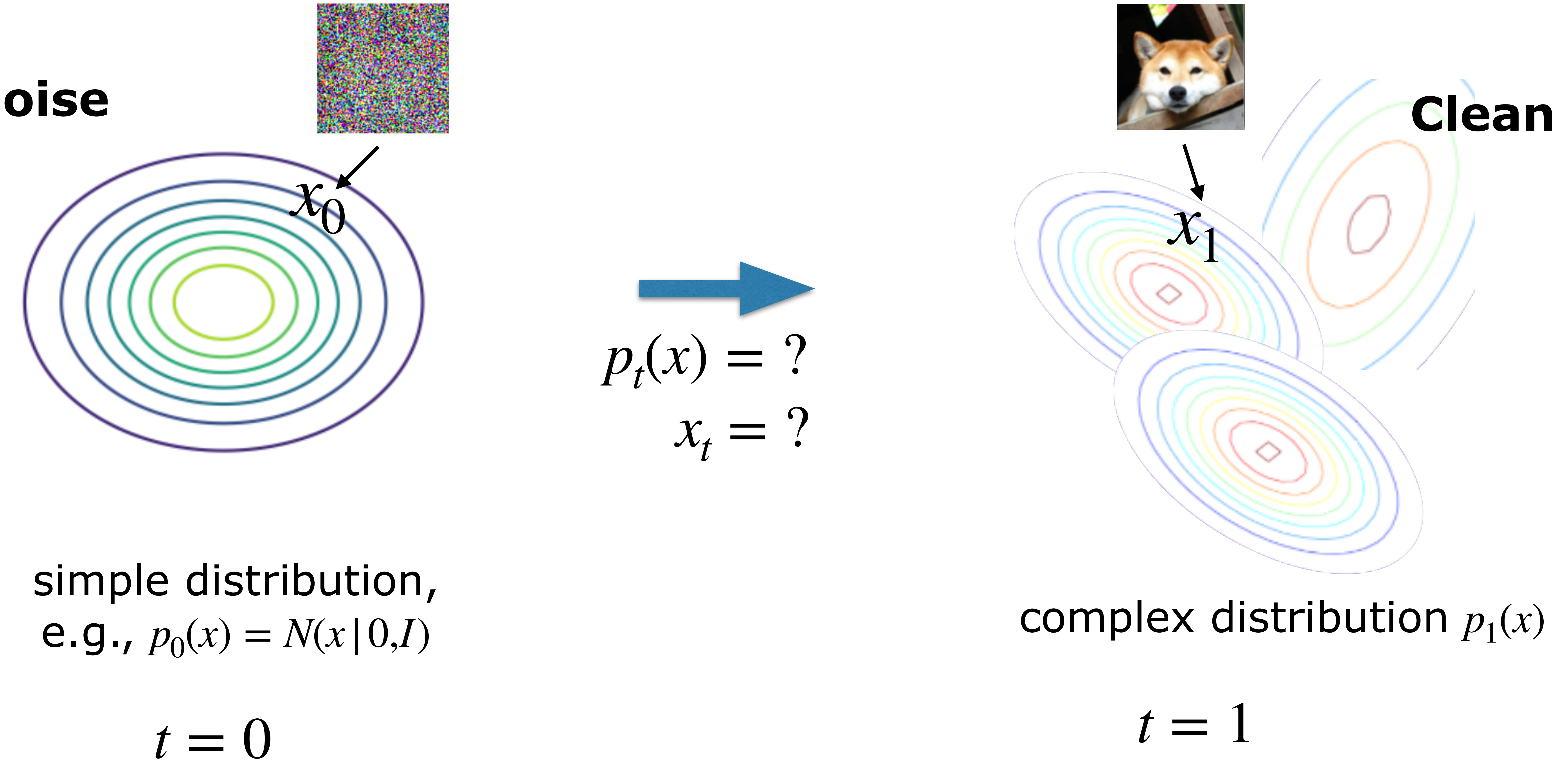


simple distribution,
e.g., $p_0(x) = N(x \mid 0, I)$

complex distribution $p_1(x)$

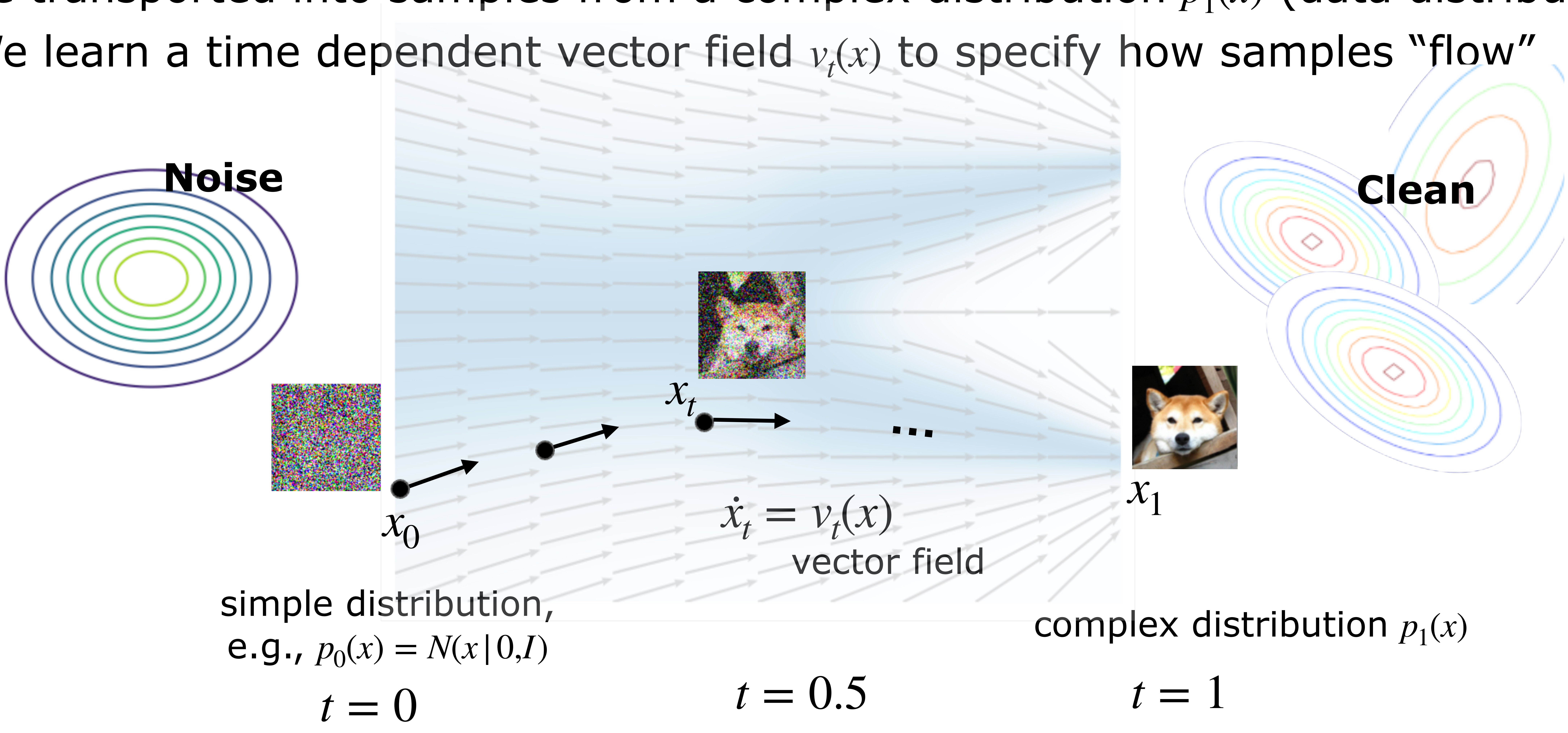
Thinking about continuous flows

- We are interested in modeling how samples from a simple distribution $p_0(x)$ can be transported into samples from a complex distribution $p_1(x)$ (data distribution)



Thinking about continuous flows

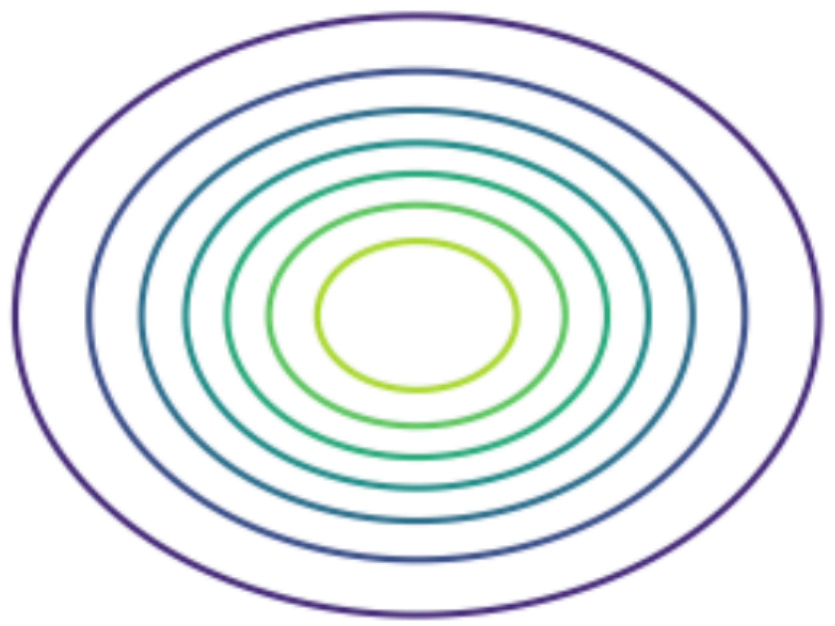
- We are interested in modeling how samples from a simple distribution $p_0(x)$ can be transported into samples from a complex distribution $p_1(x)$ (data distribution)
- We learn a time dependent vector field $v_t(x)$ to specify how samples "flow"



Example (vector fields)

- Simple noise distribution $N(0,I)$, clean data given by samples

Noise (t=0)

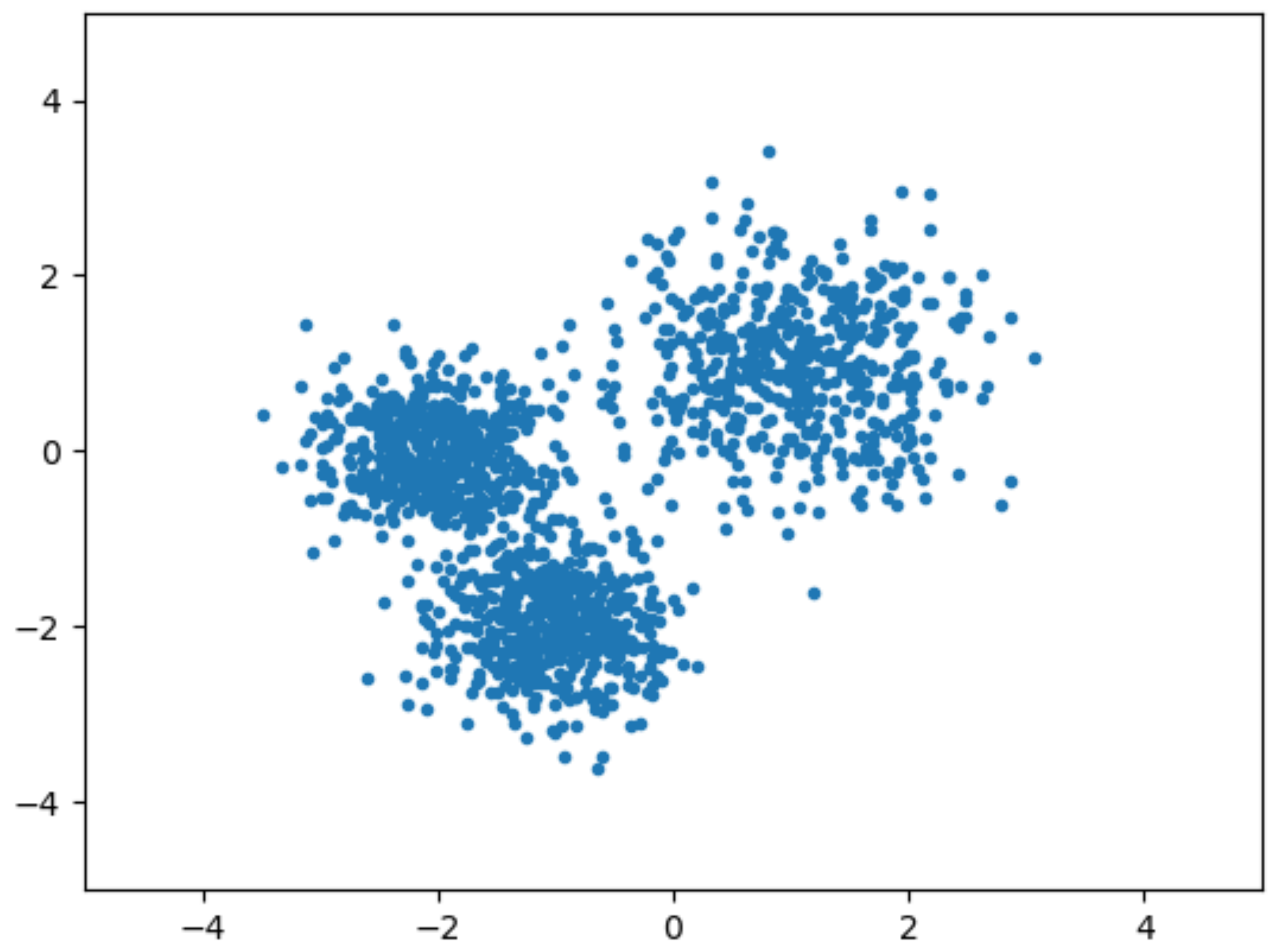


$$p_0(x) = N(x \mid 0, I)$$

vector field

$$v_t(x) = ?$$

Clean (t=1)



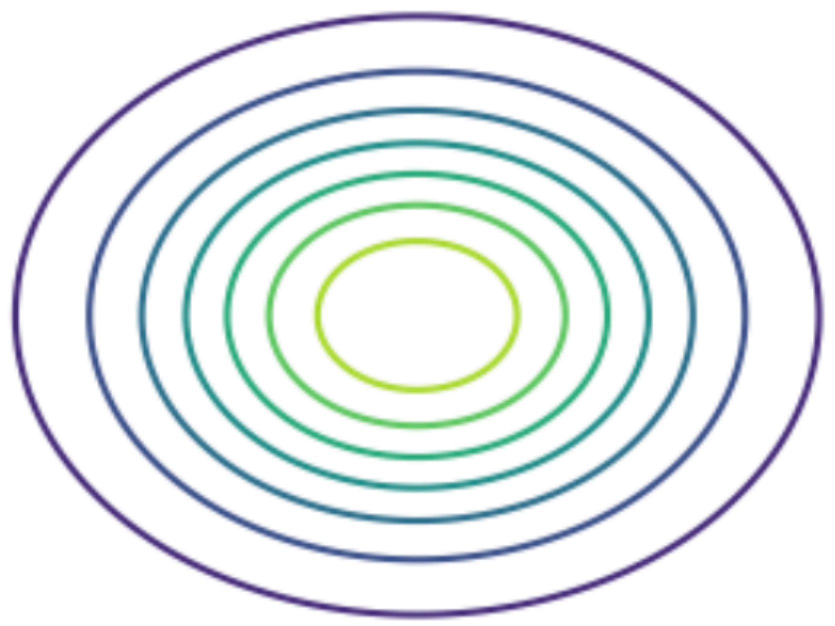
$$\hat{p}_1(x)$$

- Note: here $x \in \mathbb{R}^2$, hence $v_t(x) \in \mathbb{R}^2$ for all $x \in \mathbb{R}^2, t \in [0,1]$

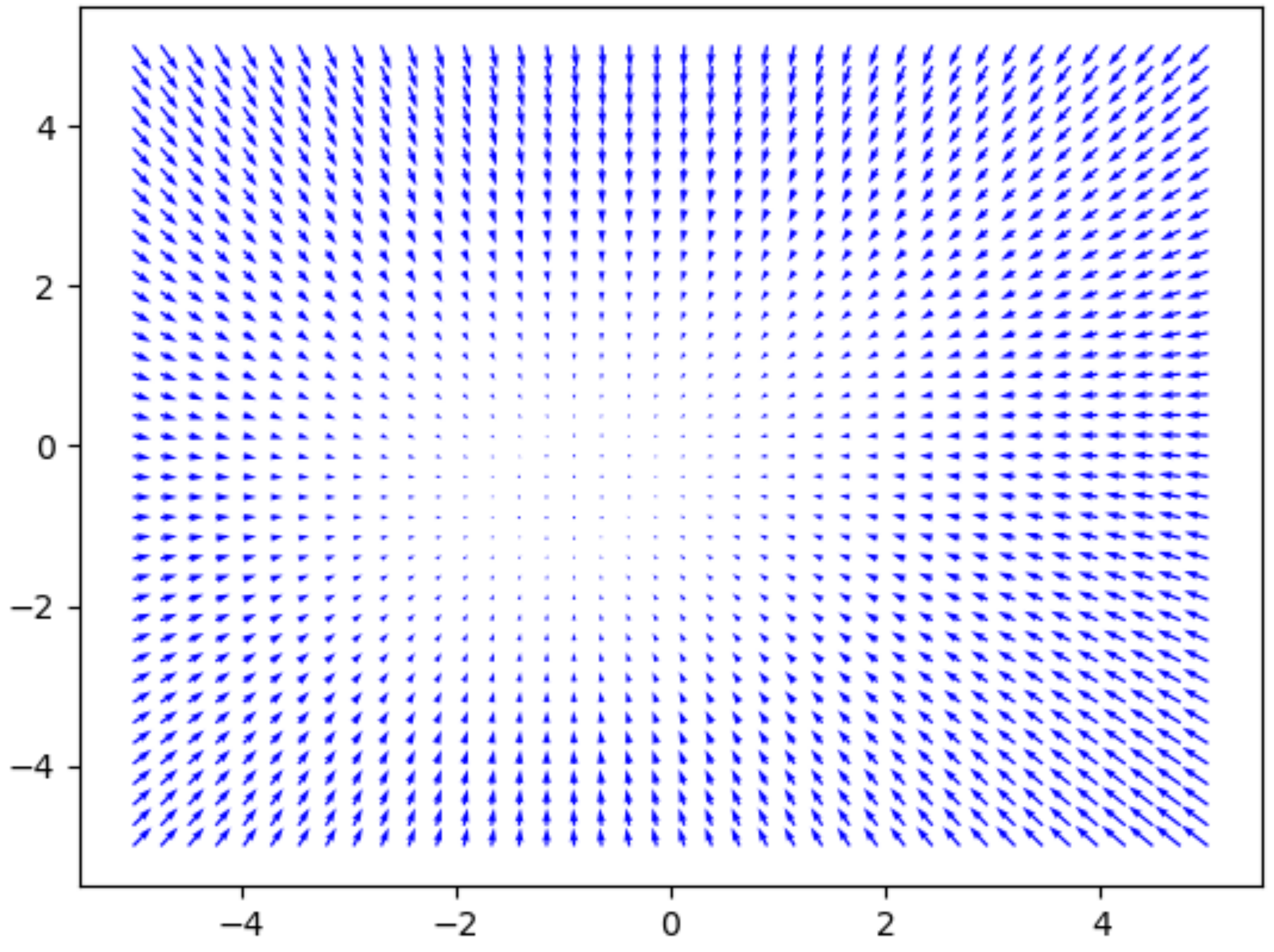
Example (vector fields)

- Simple noise distribution $N(0,I)$, clean data given by samples

Noise (t=0)

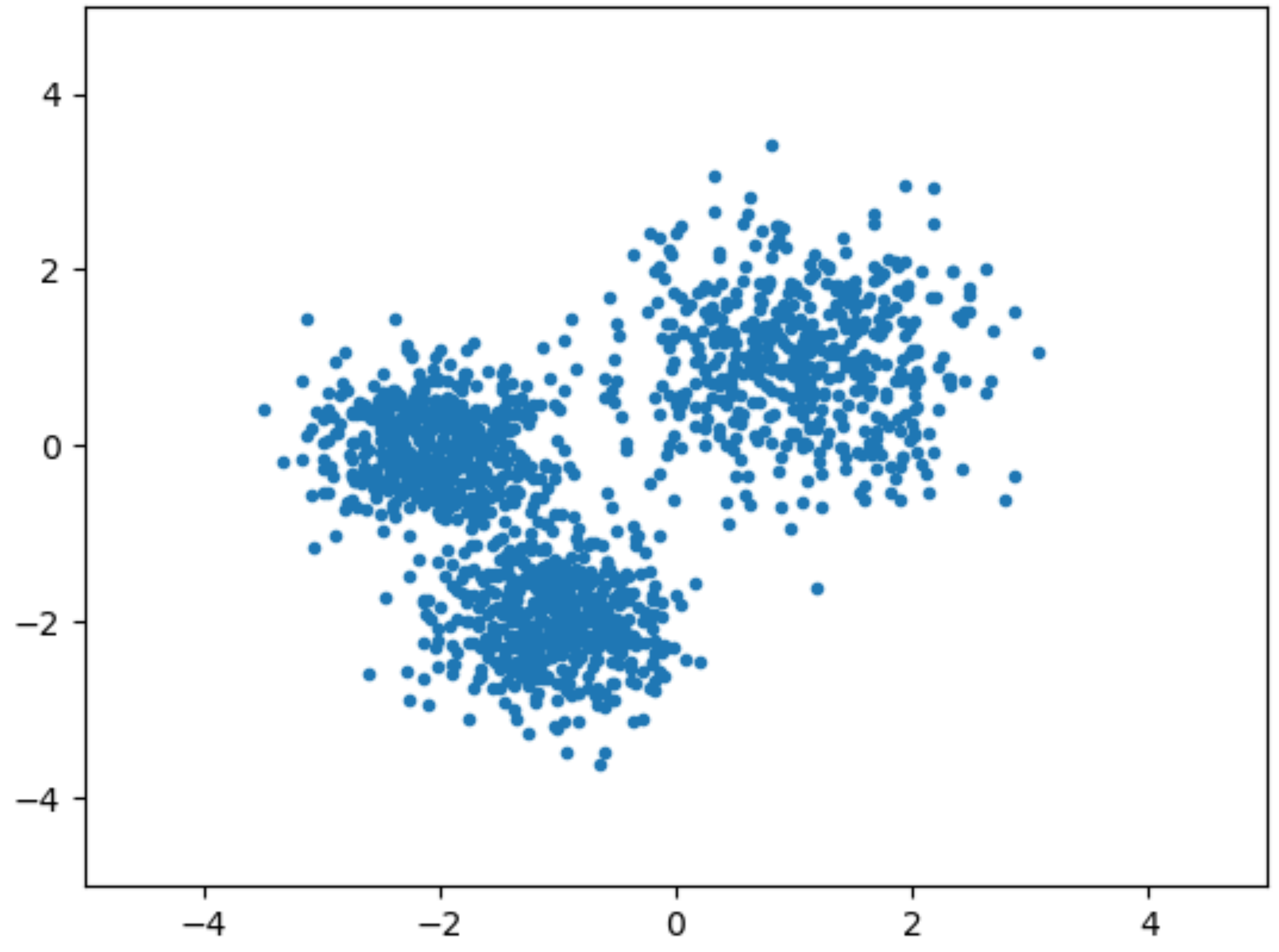


$p_0(x) = N(x \mid 0, I)$



$v_t(x),\ t = 0.1$

Clean (t=1)



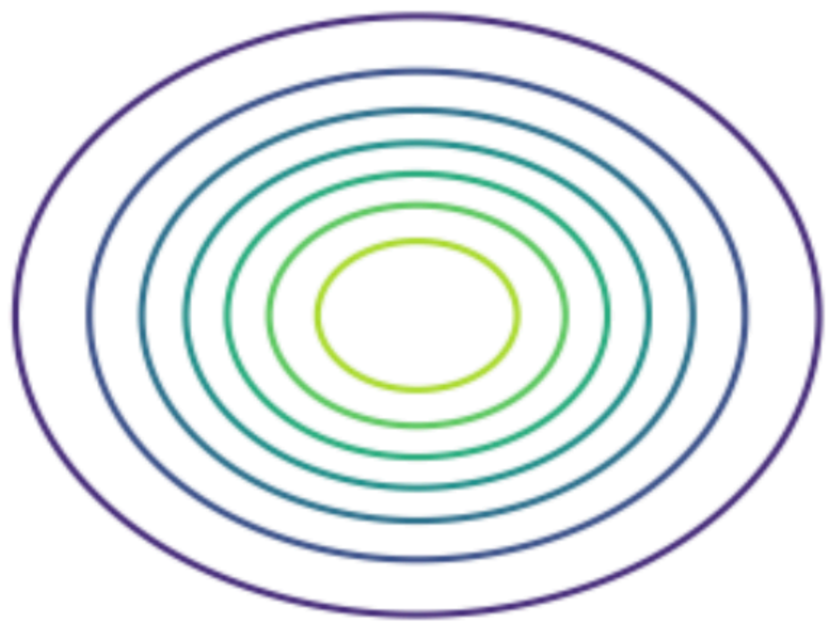
$\hat{p}_1(x)$

- Note: here $x \in \mathbb{R}^2$, hence $v_t(x) \in \mathbb{R}^2$ for all x, t

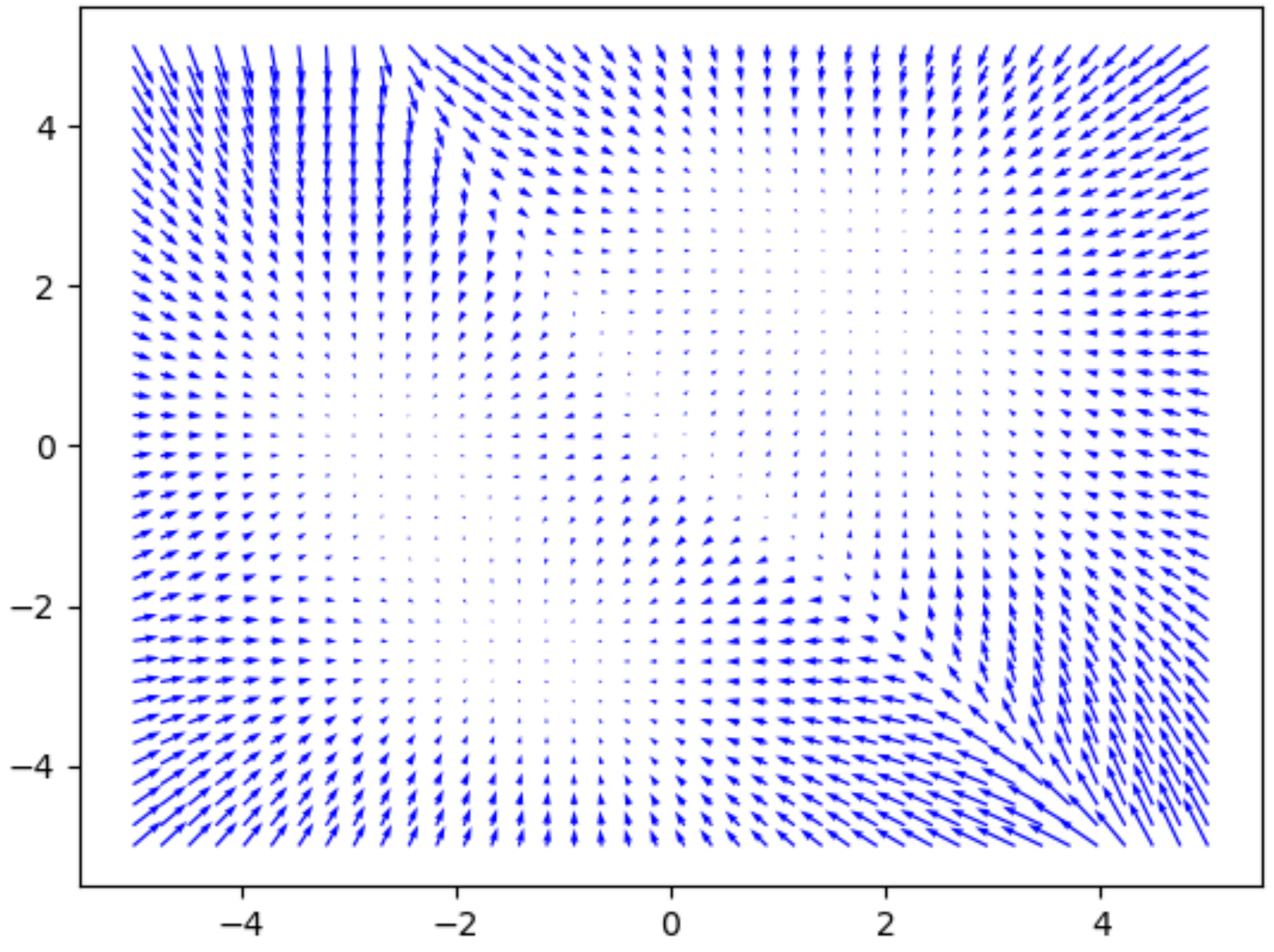
Example (vector fields)

- Simple noise distribution $N(0,I)$, clean data given by samples

Noise (t=0)

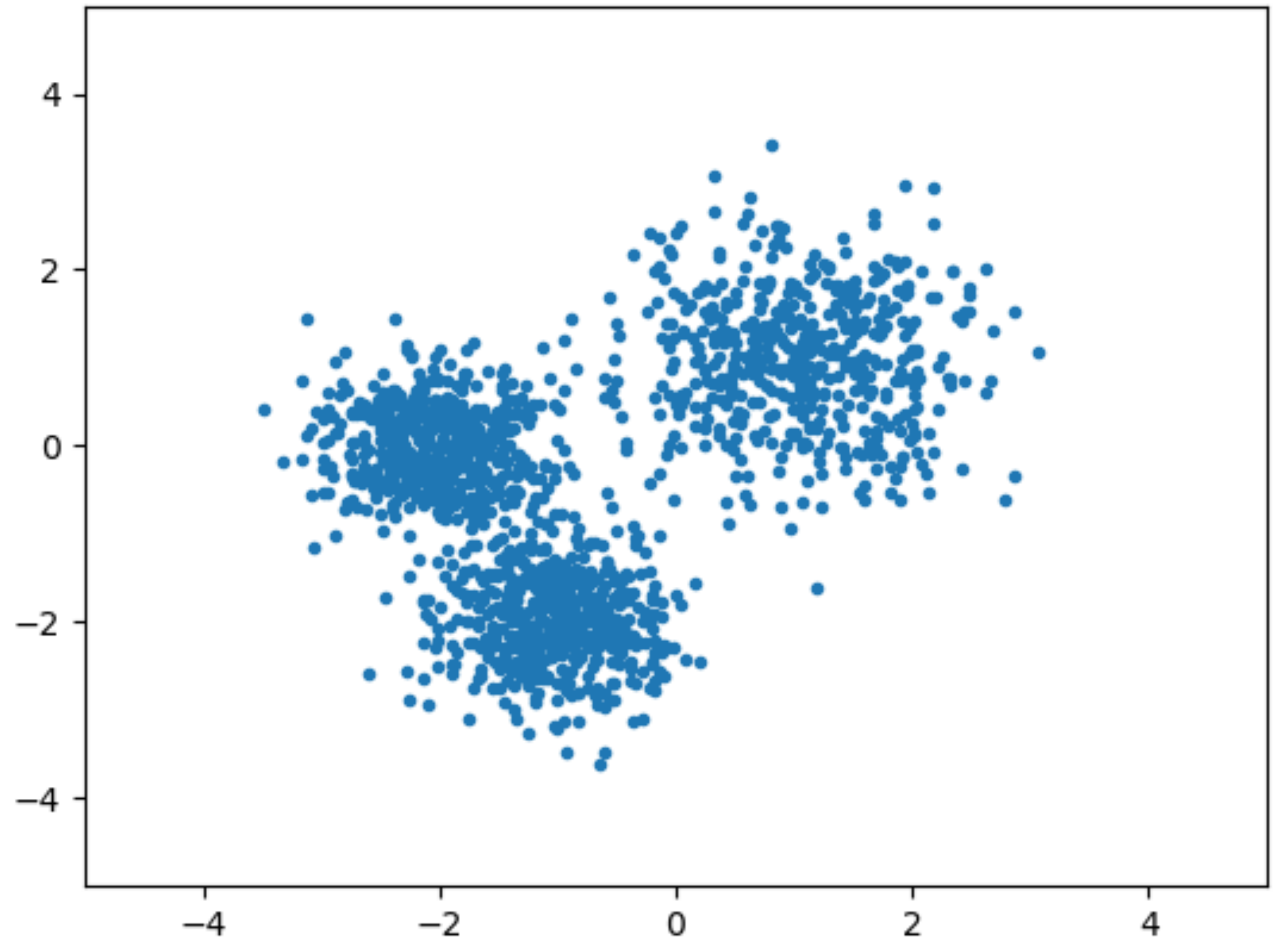


$p_0(x) = N(x \mid 0,I)$



$v_t(x),\ t = 0.5$

Clean (t=1)



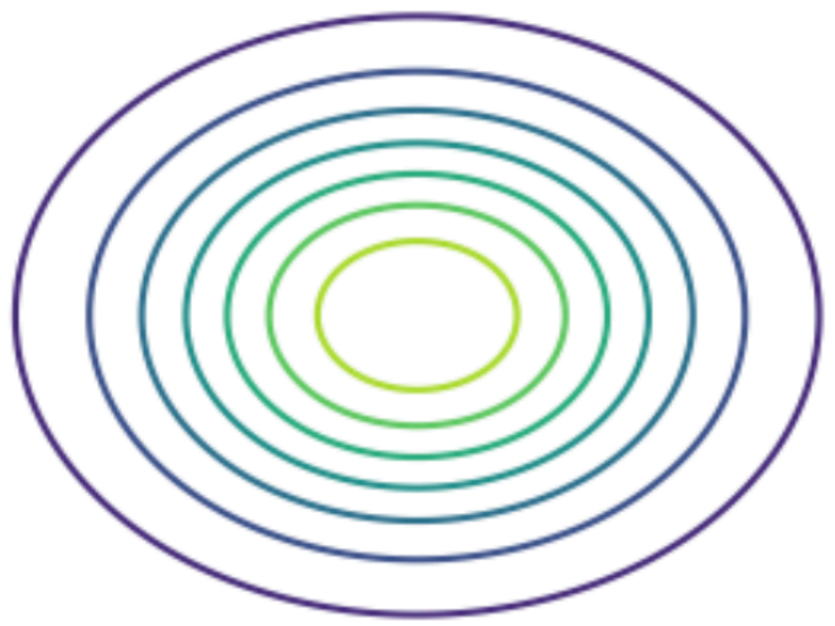
$\hat{p}_1(x)$

- Note: here $x \in \mathbb{R}^2$, hence $v_t(x) \in \mathbb{R}^2$ for all x, t

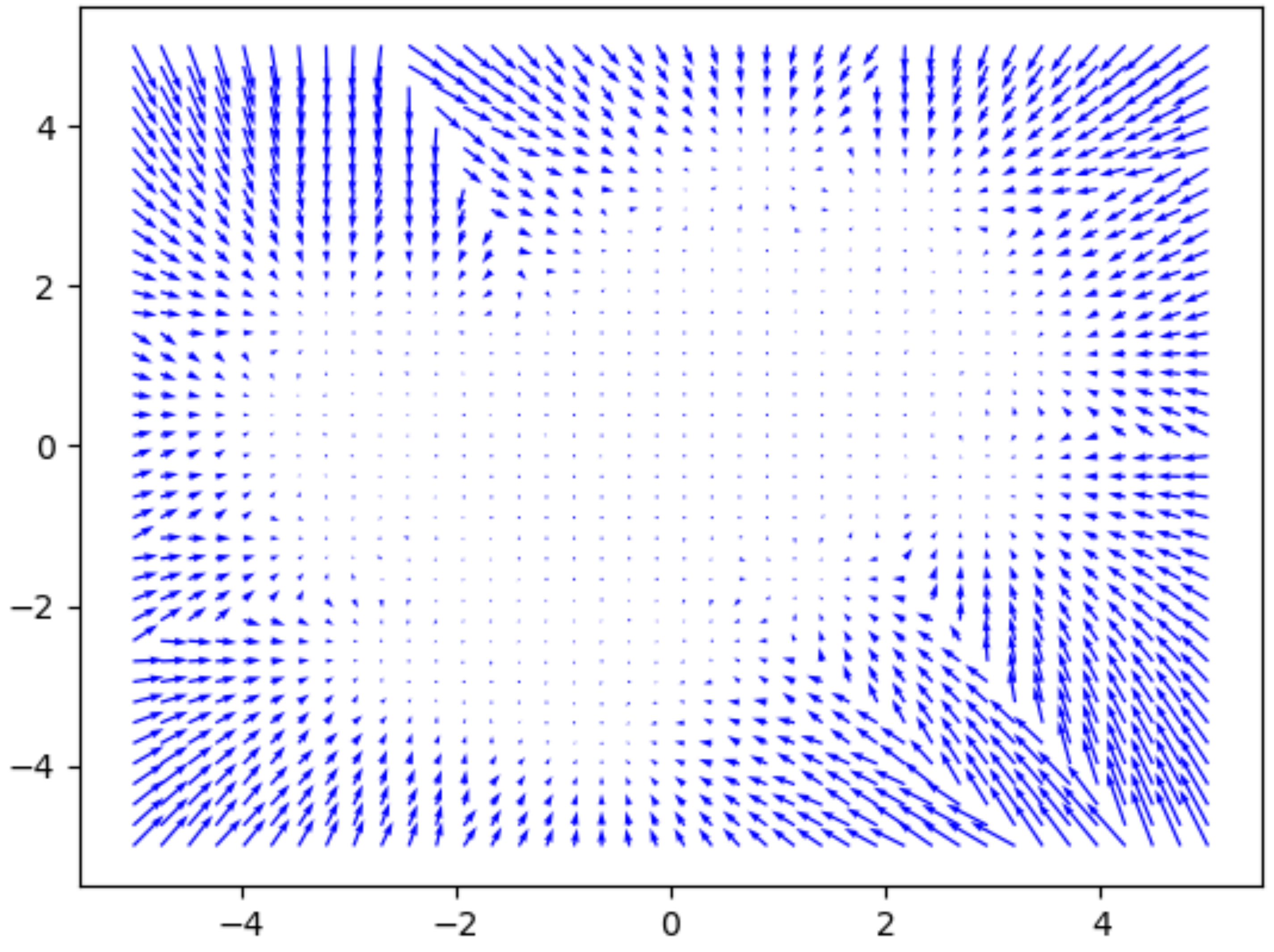
Example (vector fields)

- Simple noise distribution $N(0,I)$, clean data given by samples

Noise (t=0)

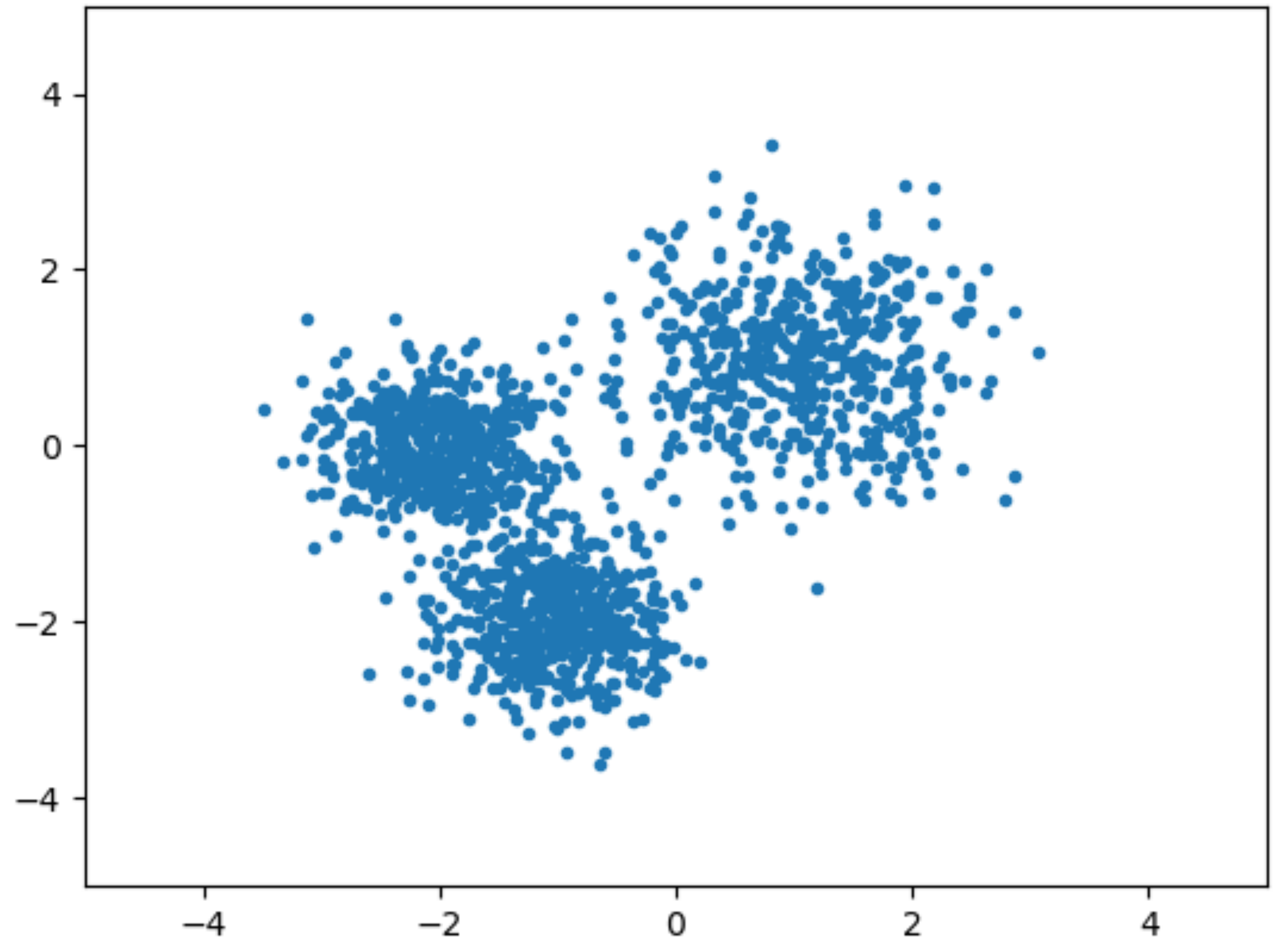


$p_0(x) = N(x\,|\,0,I)$



$v_t(x),\ t = 0.9$

Clean (t=1)



$\hat{p}_1(x)$

- Note: here $x \in \mathbb{R}^2$, hence $v_t(x) \in \mathbb{R}^2$ for all x, t

Background: continuity equation

- We are interested in how samples from a simple distribution $p_0(x)$ flow to samples from a complex distribution $p_1(x)$ as a function of time
- This is analogous to a problem, e.g., in fluid dynamics where the fluid density evolves over time depending on the fluid velocity (field)
- In our case, samples evolve according to a vector field and the distribution of samples at any intermediate time is governed by the continuity equation:

time dependent vector field $\frac{d}{dt}x_t = \dot{x}_t = v_t(x_t)$

$$\frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$$

" rate of change of density at x = rate coming in − rate going out "

Background: continuity equation

- We can think about modeling the flow of particles as initial samples from a simple distribution $p_0(x) = N(x|0,I)$ to samples from $p_1(x)$ in three different ways

(1)	$p_t(x)$	$\frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$	$x_0 \sim p_0(x), \quad \dot{x}_t = v_t(x_t), \quad t \in (0,t]$
	specify probability flow	solve/learn the vector field	sample using the vector field

- here we would specify how we wish the probability distribution to change from $p_0(x)$ to $p_1(x)$ as a function of time, e.g., $p_t(x) = (1-t)p_0(x) + tp_1(x)$

Background: continuity equation

- We can think about modeling the flow of particles as initial samples from a simple distribution $p_0(x) = N(x \mid 0, I)$ to samples from $p_1(x)$ in three different ways

(1) $p_t(x)$ $\quad$ $\frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$ $\quad$ $x_0 \sim p_0(x), \quad \dot{x}_t = v_t(x_t), \quad t \in (0,t]$

specify probability flow $\quad$ solve/learn the vector field $\quad$ sample using the vector field

- here we would specify how we wish the probability distribution to change from $p_0(x)$ to $p_1(x)$ as a function of time, e.g., $p_t(x) = (1-t)p_0(x) + tp_1(x)$
- finding the vector field that would support this density evolution is not easy!! (nor unique)

Background: continuity equation

- We can think about modeling the flow of particles as initial samples from a simple distribution $p_0(x) = N(x|0,I)$ to samples from $p_1(x)$ in three different ways

(1) $p_t(x)$ $\quad \frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$ $\quad x_0 \sim p_0(x), \quad \dot{x}_t = v_t(x_t), \quad t \in (0,t]$

specify probability flow — solve/learn the vector field — sample using the vector field

(2) $v_t(x)$ $\quad x_0 \sim p_0(x), \quad \dot{x}_t = v_t(x_t), \quad t \in (0,t]$ $\quad \frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$

sample using the vector field — calculate probability flow

- we could instead start by specifying the vector field itself, then everything else is easy

Background: continuity equation

- We can think about modeling the flow of particles as initial samples from a simple distribution $p_0(x) = N(x|0,I)$ to samples from $p_1(x)$ in three different ways

(1) $p_t(x)$ $\quad \frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$ $\quad x_0 \sim p_0(x), \quad \dot{x}_t = v_t(x_t), \quad t \in (0,t]$

specify probability flow — solve/learn the vector field — sample using the vector field

(2) $v_t(x)$ $\quad x_0 \sim p_0(x), \quad \dot{x}_t = v_t(x_t), \quad t \in (0,t]$ $\quad \frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$

sample using the vector field — calculate probability flow

- we could instead start by specifying the vector field itself, then everything else is easy
- but finding the vector field that gives us $p_1(x)$ at the other end (t=1) is challenging!!

Background: continuity equation

- We can think about modeling the flow of particles as initial samples from a simple distribution $p_0(x) = N(x \mid 0, I)$ to samples from $p_1(x)$ in three different ways

(1) $p_t(x)$ — specify probability flow

$$\frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$$

solve/learn the vector field

$$x_0 \sim p_0(x), \quad \dot{x}_t = v_t(x_t), \quad t \in (0,t]$$

sample using the vector field

(2) $v_t(x)$

$$x_0 \sim p_0(x), \quad \dot{x}_t = v_t(x_t), \quad t \in (0,t]$$

sample using the vector field

$$\frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$$

calculate probability flow

(3)

$$x_t = x_0 + t(x_1 - x_0)$$

$$x_0 \sim p_0(x), \quad x_1 \sim p_1(x)$$

specify simple interpolating trajectories between source and target samples

$v_t(x)$

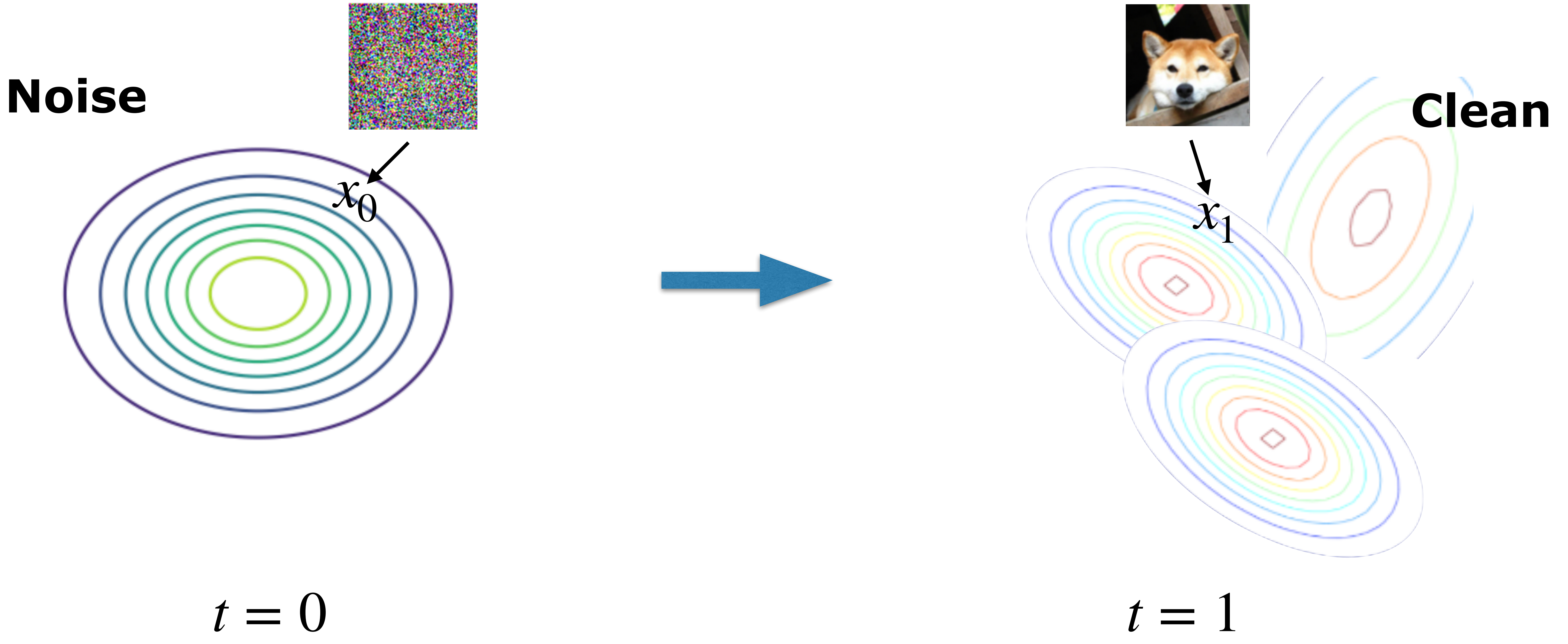
learn the vector field from such guidance

$$x_0 \sim p_0(x), \quad \dot{x}_t = v_t(x_t), \quad t \in (0,t]$$

sample using the vector field

Flow matching

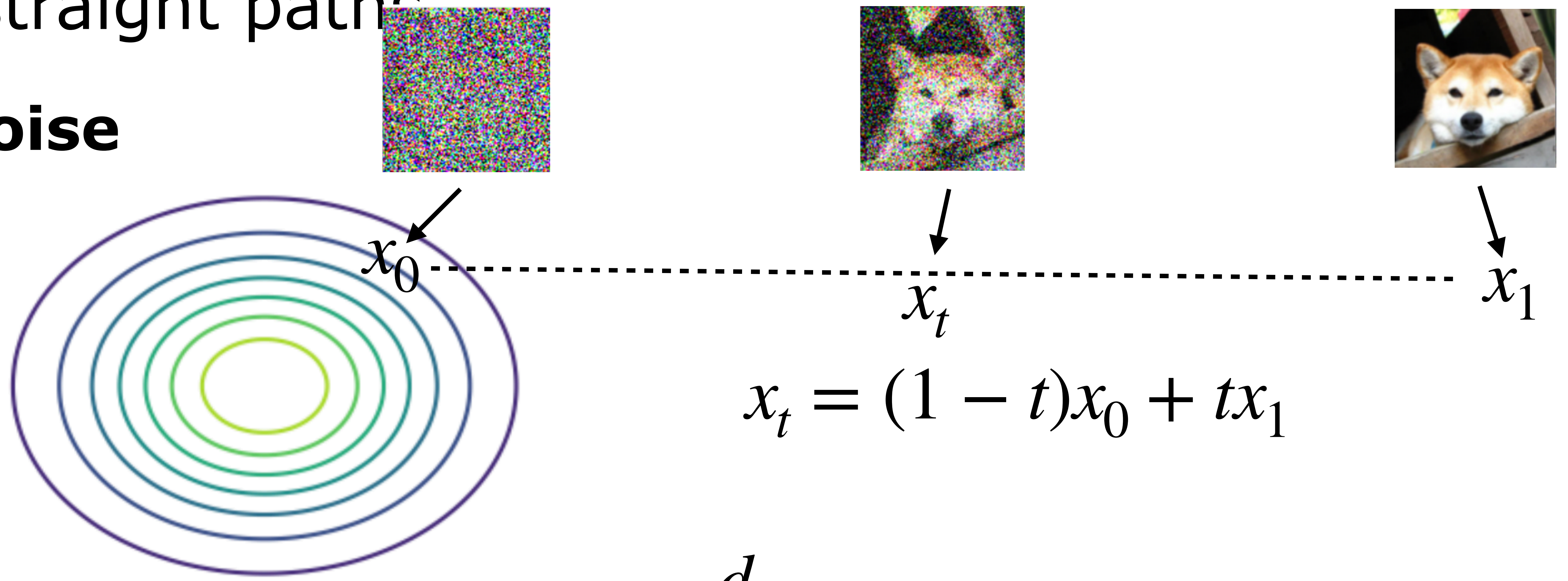
- We can think about turning noise into clean samples along simple (linear) interpolating trajectories and learn a model to do so
- This is more straightforward than diffusion (also appears to work better)



Flow matching: a simple setting

- We can think about turning noise into clean samples along simple (linear) interpolating trajectories and learn a model to do so
- E.g., for a single clean image we can easily map noise samples back to the image via straight paths

Noise



Here the target is just a single example

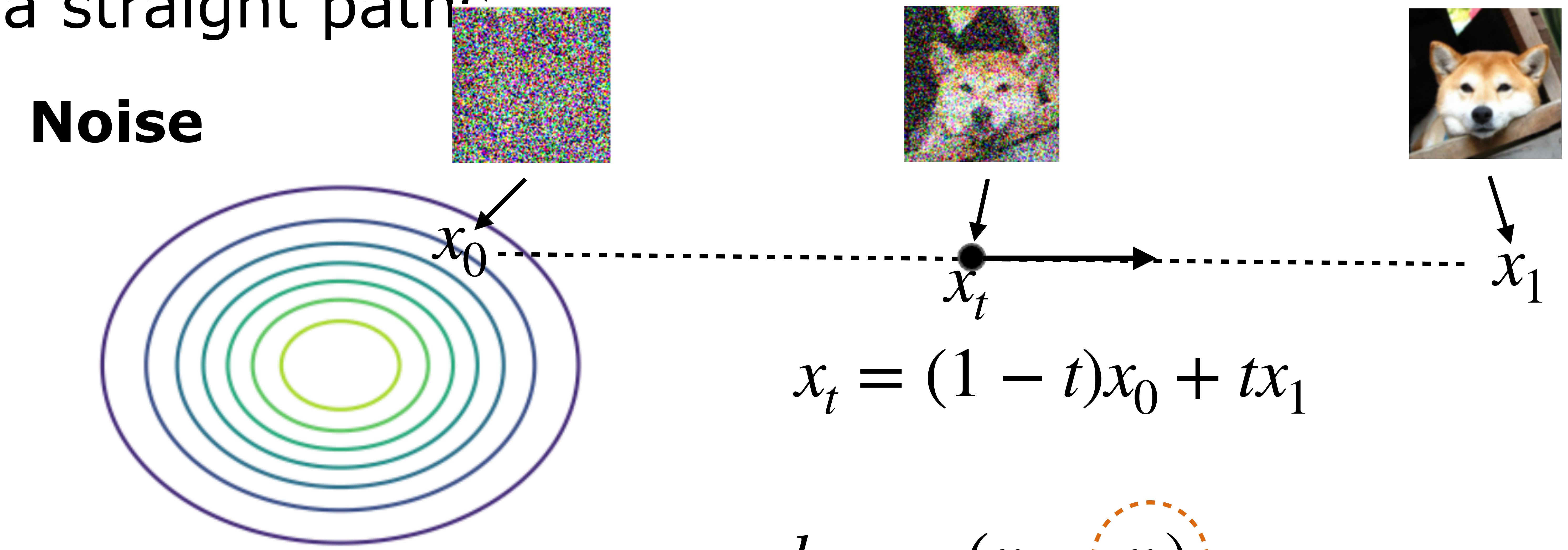
$$x_t = (1 - t)x_0 + tx_1$$

$$\frac{d}{dt}x_t = (x_1 - x_0)$$

$t = 0$ $\qquad\qquad$ $t = 1$

Flow matching: a simple setting

- We can think about turning noise into clean samples along simple (linear) interpolating trajectories and learn a model to do so
- E.g., for a single clean image we can easily map noise samples back to the image via straight paths



Here the target is just a single example

$$x_t = (1 - t)x_0 + tx_1$$

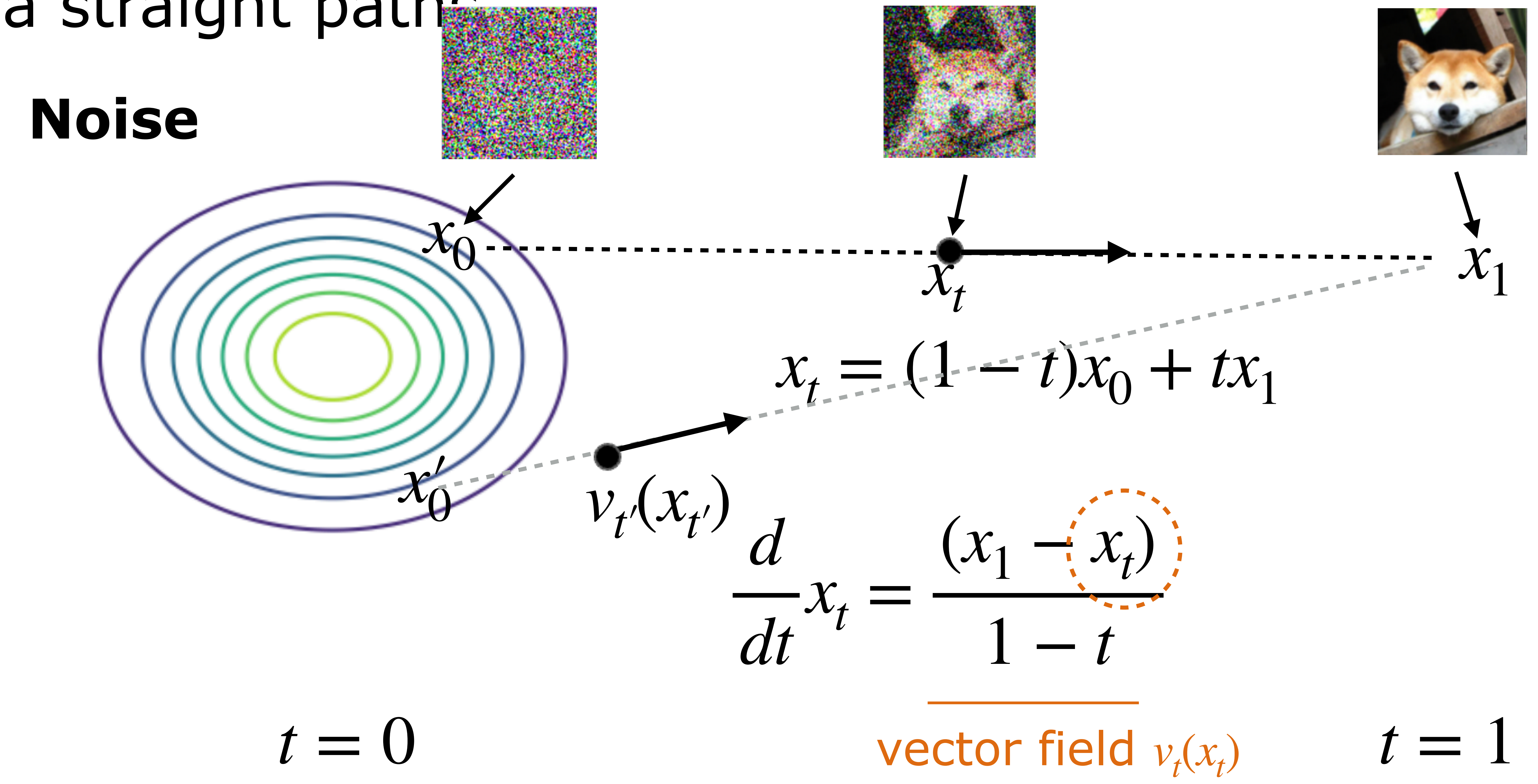
$$\frac{d}{dt}x_t = \underbrace{\frac{(x_1 - x_t)}{1 - t}}_{\text{vector field } v_t(x_t)}$$

$t = 0$

$t = 1$

Flow matching: a simple setting

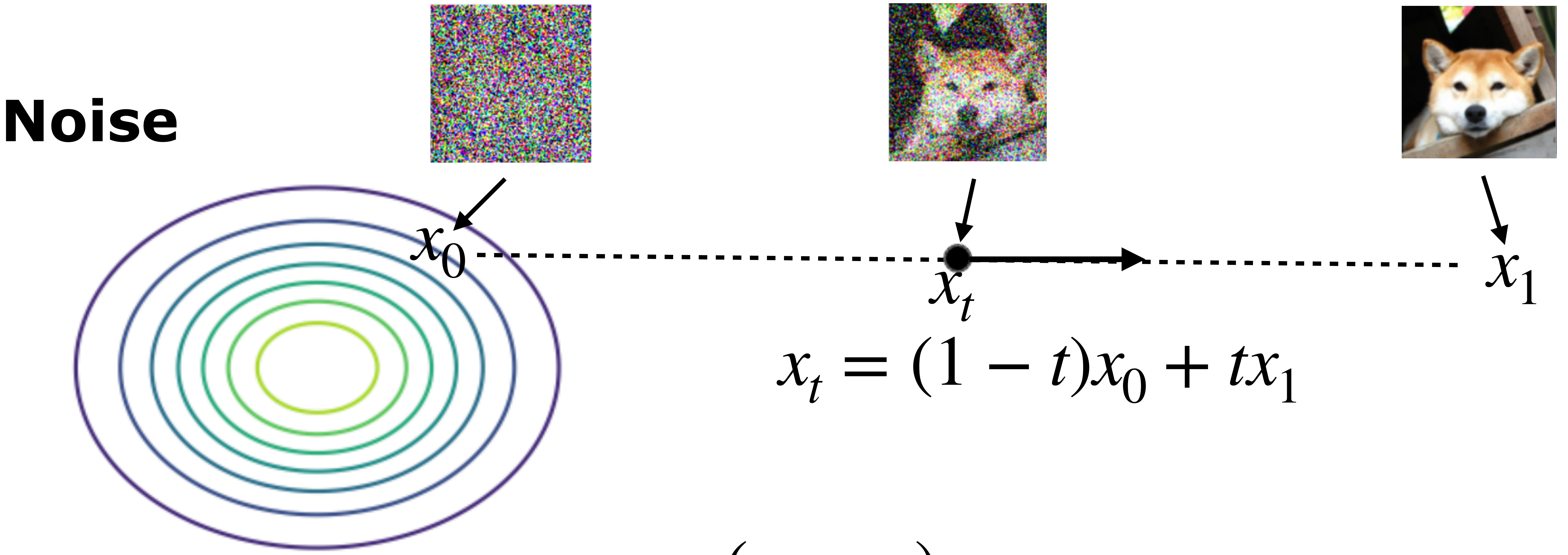
- We can think about turning noise into clean samples along simple (linear) interpolating trajectories and learn a model to do so
- E.g., for a single clean image we can easily map noise samples back to the image via straight paths



Here the target is just a single example

Probability flow in a simple setting

- If we sample $x_0 \sim N(0,I)$ and set $x_t = (1-t)x_0 + tx_1$ then $p_t(x_t)$ is also Gaussian with mean tx_1 and variance $(1-t)^2$… so we know the probability flow!



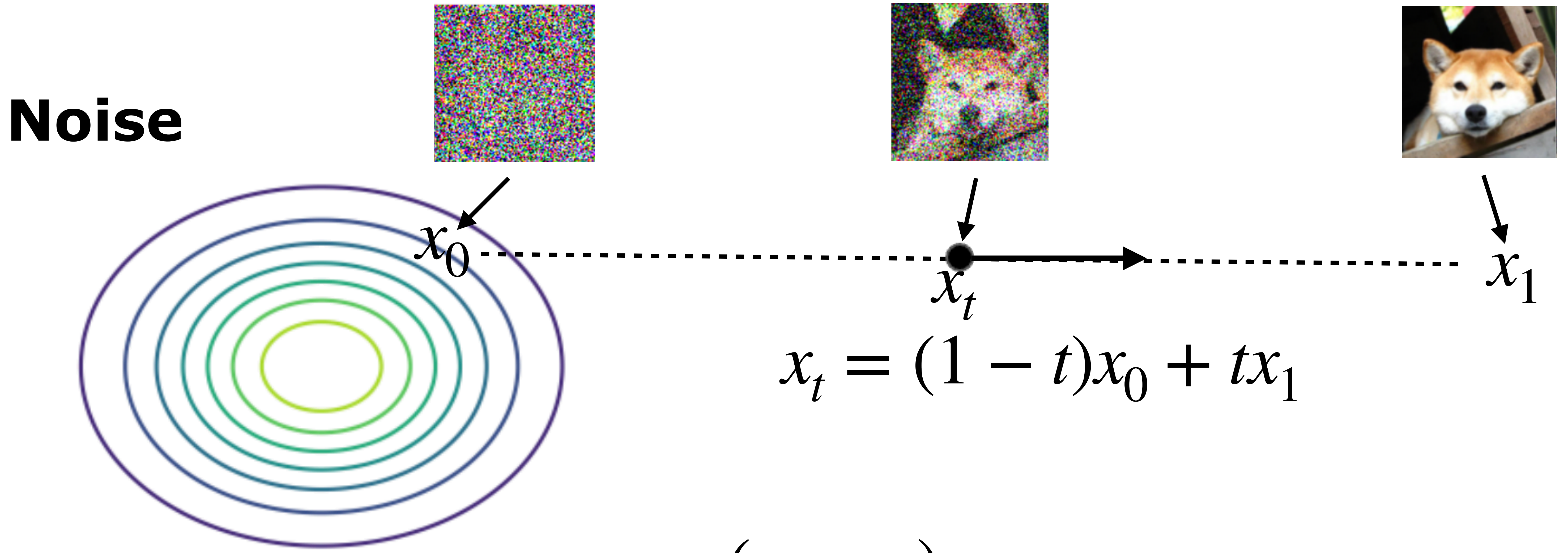
Here the target is just a single example

$$v_t(x) = \frac{(x_1 - x)}{1-t} \qquad p_t(x) = N(x \mid tx_1, (1-t)^2 I)$$

- **Exercise**: show that with these choices (in 1d): $\frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$

Probability flow in a simple setting

- If we sample $x_0 \sim N(0,I)$ and set $x_t = (1-t)x_0 + tx_1$ then $p_t(x_t)$ is also Gaussian with mean tx_1 and variance $(1-t)^2$… so we know the probability flow!



Here the target is just a single example

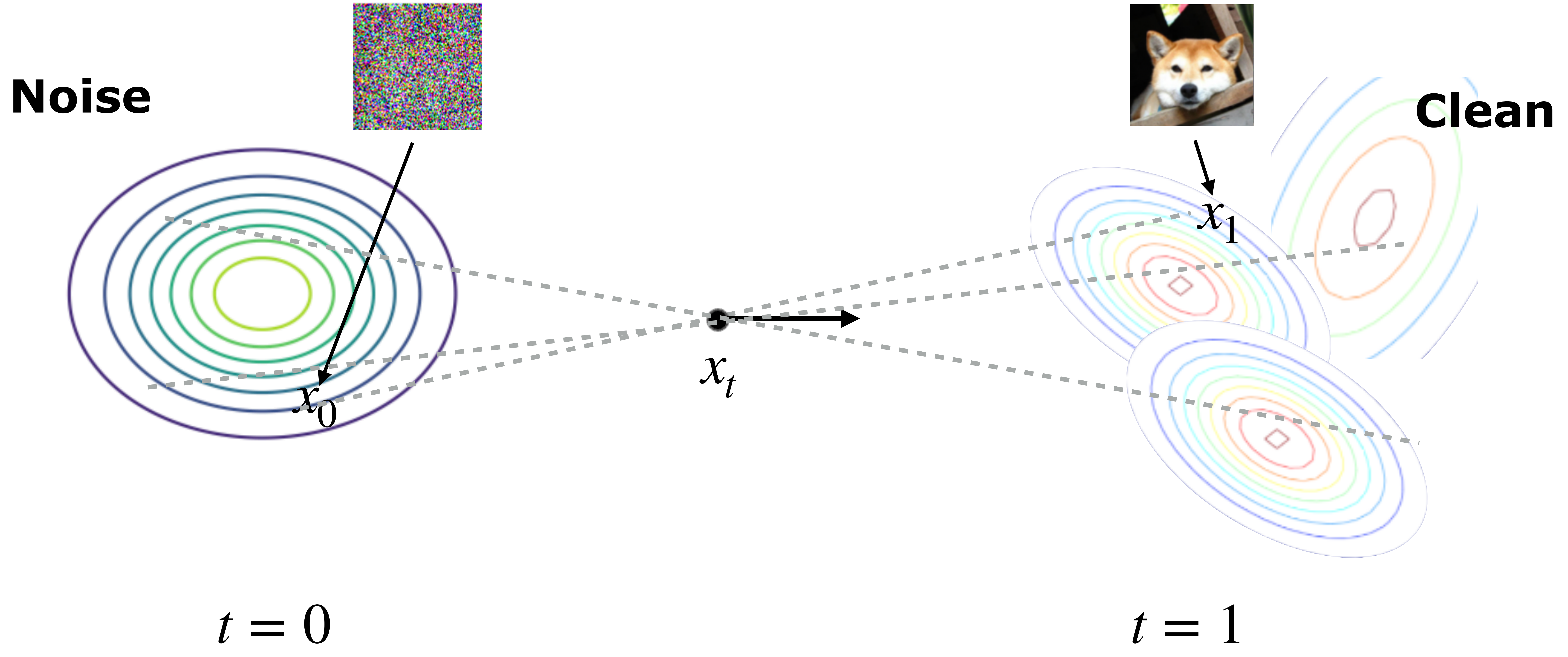
$$v_t(x) = \frac{(x_1 - x)}{1-t} \qquad p_t(x) = N(x \mid tx_1, (1-t)^2 I) \rightarrow \delta(x - x_1)$$

as $t \rightarrow 1$

- **Exercise**: show that with these choices (in 1d): $\frac{d}{dt}p_t(x) = -\nabla_x \cdot (p_t(x)v_t(x))$

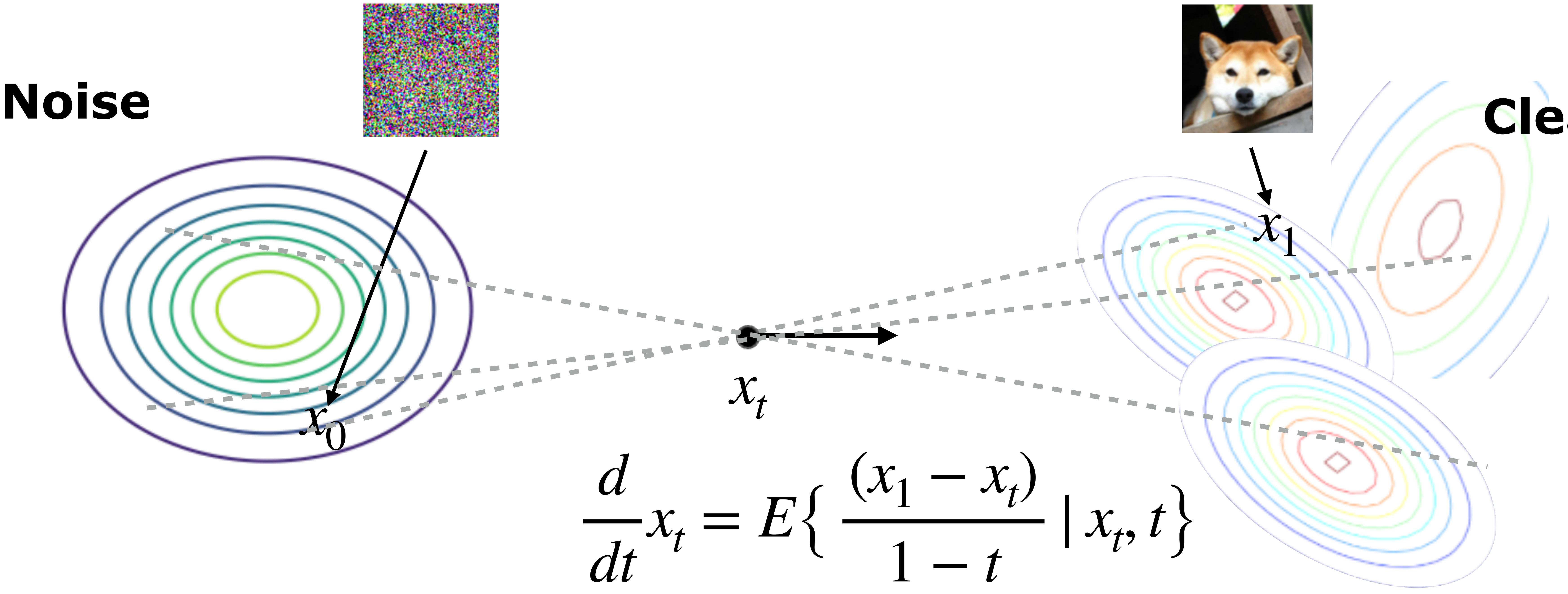
Flow matching

- Given t and x_t, there are multiple pairs of x_0 and x_1 whose linear interpolation at time t would result in x_t; each of them suggest going in a different direction
- The vector field we want is a (conditional) expectation of these suggestions



Flow matching

- Given t and x_t, there are multiple pairs of x_0 and x_1 whose linear interpolation at time t would result in x_t; each of them suggest going in a different direction
- The vector field we want is a (conditional) expectation of these suggestions



$$\frac{d}{dt}x_t = \underbrace{E\left\{ \frac{(x_1 - x_t)}{1 - t} \,|\, x_t, t \right\}}_{\text{vector field}}$$

$t = 0$ $\qquad$ $t = 1$

Flow matching: algorithms

- **Training algorithm:**

sample $x_0 \sim N(0,I)$

sample $x_1 \sim q(x_1)$ (data distribution)

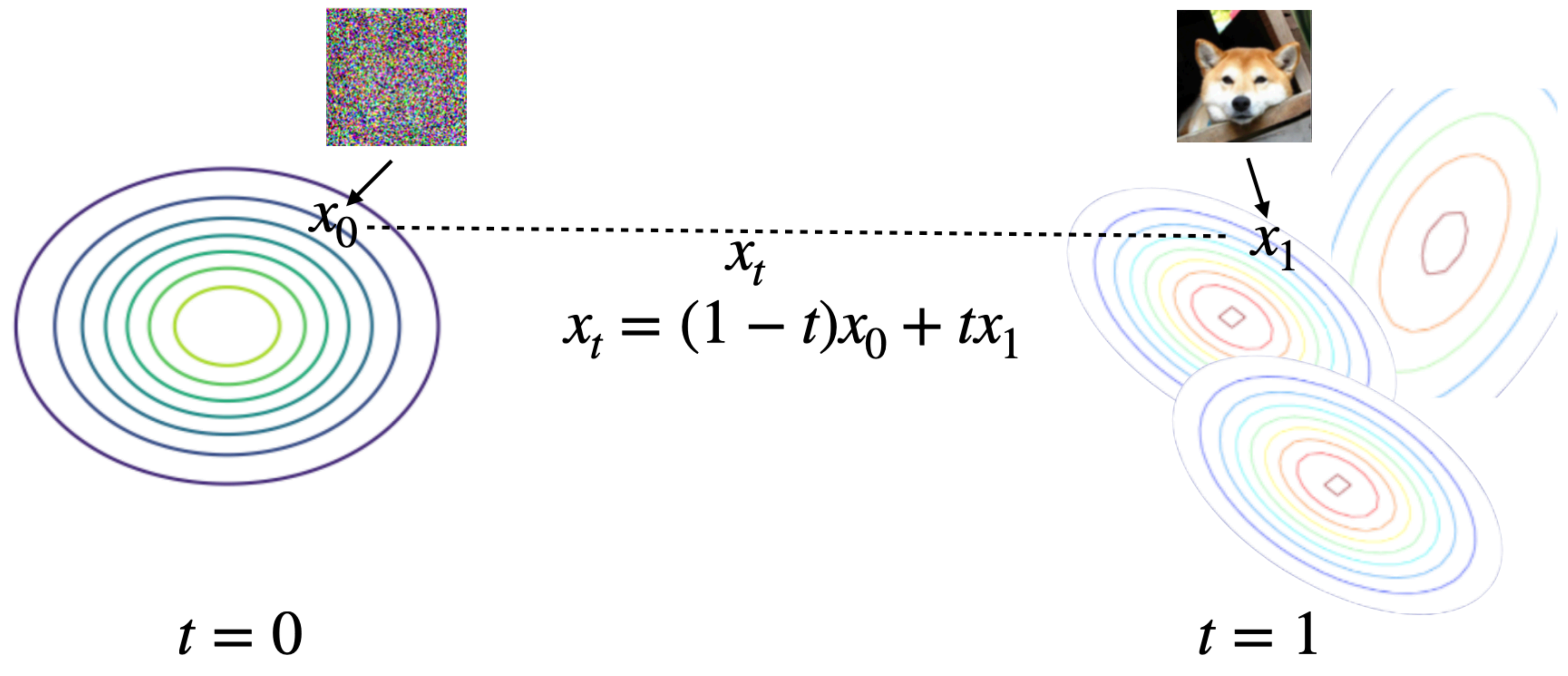
sample $t \sim U(0,1)$

$x_t = (1-t)x_0 + tx_1$

take a gradient step to min

$$\|\frac{(x_1 - x_t)}{1-t} - v_\theta(x_t, t)\|^2$$

vector field



Flow matching: algorithms

- **Training algorithm:**

sample $x_0 \sim N(0,I)$

sample $x_1 \sim q(x_1)$ (data distribution)

sample $t \sim U(0,1)$

$x_t = (1-t)x_0 + tx_1$

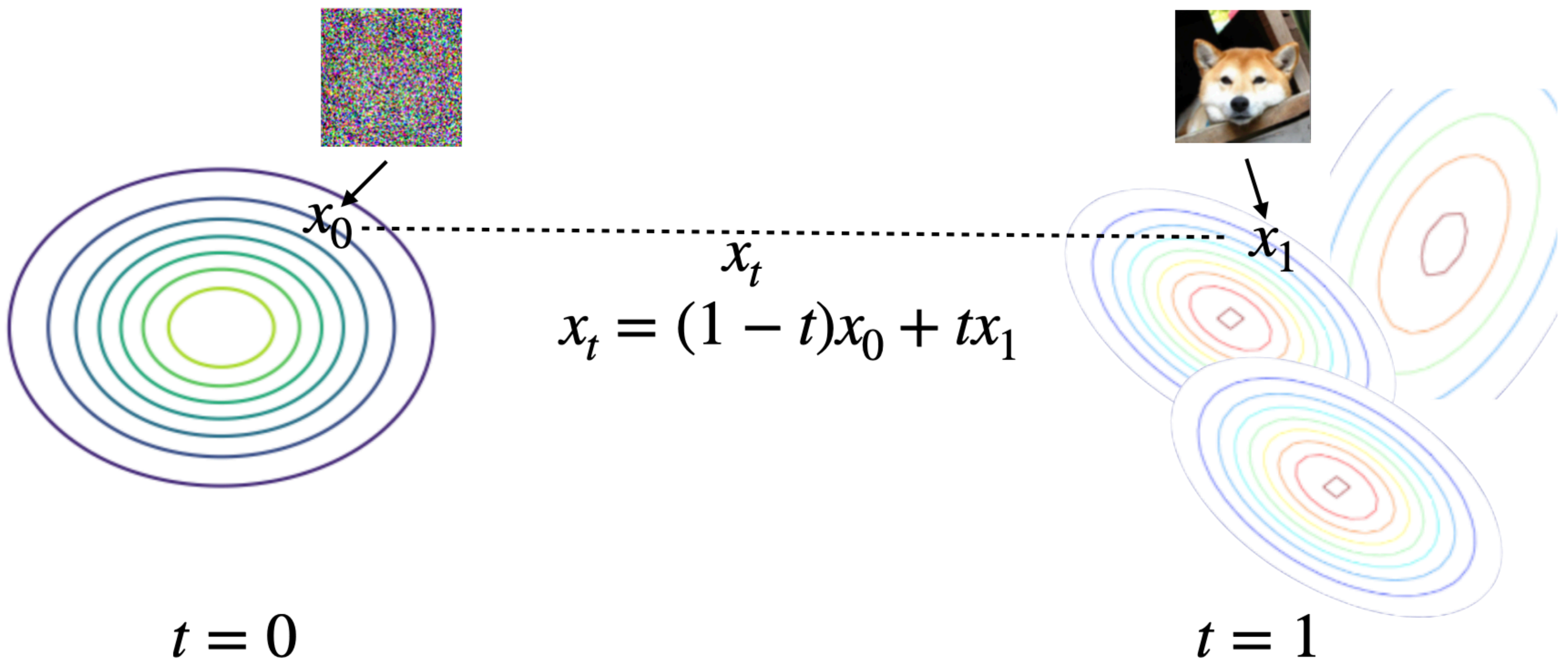
take a gradient step to min

$$\left\|\frac{(x_1 - x_t)}{1-t} - v_\theta(x_t, t)\right\|^2$$

vector field

$$\Rightarrow v_{\hat{\theta}}(x_t, t) \approx E\left\{\frac{(x_1 - x_t)}{1-t} \,\middle|\, x_t, t\right\}$$

optimal MSE estimate is
conditional expectation



Flow matching: algorithms

- **Training algorithm:**

sample $x_0 \sim N(0,I)$

sample $x_1 \sim q(x_1)$ (data distribution)

sample $t \sim U(0,1)$

$x_t = (1-t)x_0 + tx_1$

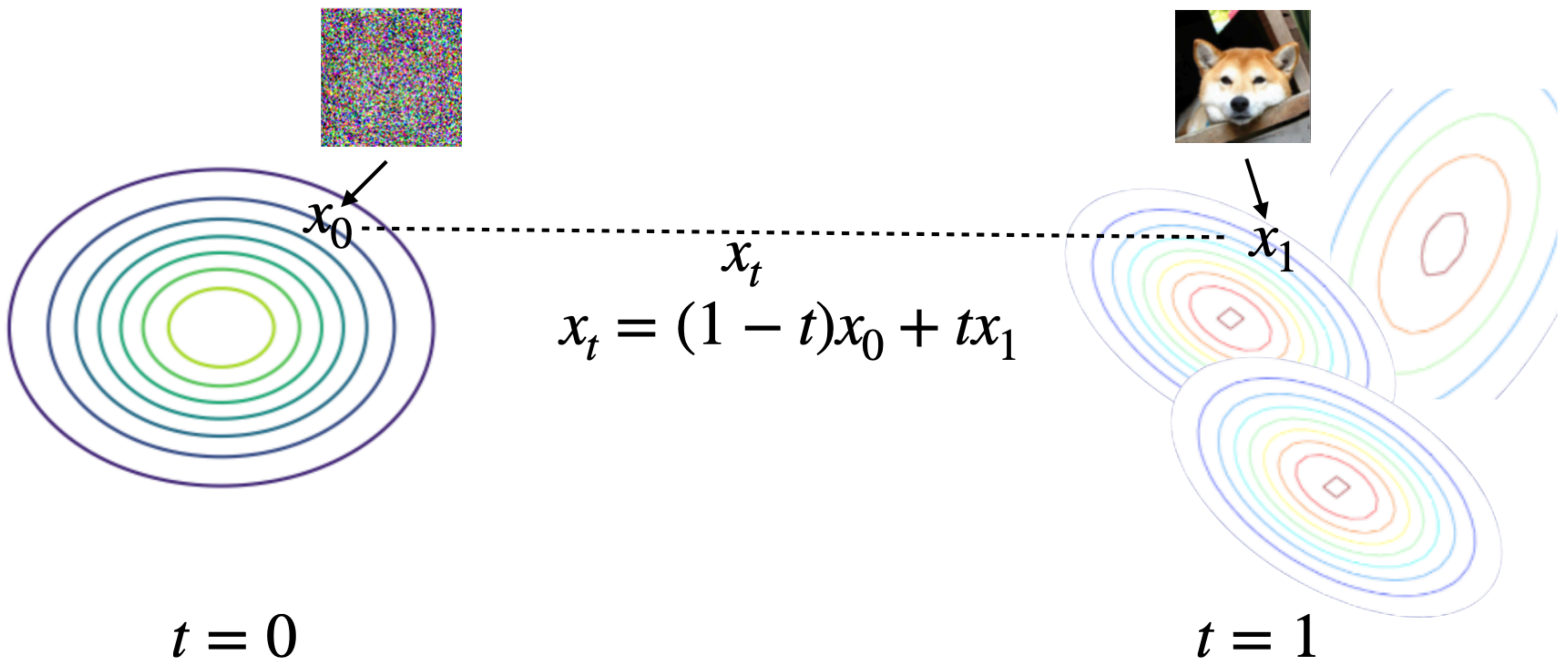
take a gradient step to min

$$\left\|\frac{(x_1 - x_t)}{1-t} - v_\theta(x_t, t)\right\|^2$$

vector field

$$\Rightarrow v_{\hat{\theta}}(x_t, t) \approx E\left\{\frac{(x_1 - x_t)}{1-t} \,\middle|\, x_t, t\right\}$$

optimal MSE estimate is conditional expectation



Sampling algorithm

sample $x_0 \sim N(0,I)$

$$\frac{d}{dt}x_t = v_\theta(x_t, t) \quad \text{from } t = 0 \text{ to } t = 1$$

Justification

- We wish to show that the conditional expectation gives us a vector field that transports $p_0(x)$ to $q(x)$ (data).
- In a single target example case, we can easily obtain the vector field that transports all noise samples x_0 to that single $\hat{x}_1$

$$x_0 \sim p_0(x)$$
$$x_1 = \hat{x}_1$$

$$v(x_t, t \,|\, \hat{x}_1) = \frac{\hat{x}_1 - x_t}{1 - t}$$

$$\frac{d}{dt} p_t(x \,|\, \hat{x}_1) = -\nabla_x \cdot (p_t(x \,|\, \hat{x}_1) v(x \,|\, t, \hat{x}_1))$$

$$p_t(x \,|\, \hat{x}_1) = N(x \,|\, t\hat{x}_1, (1-t)^2 I)$$

Justification

- We wish to show that the conditional expectation gives us a vector field that transports $p_0(x)$ to $q(x)$ (data).
- In a single target example case, we can easily obtain the vector field that transports all noise samples x_0 to that single $\hat{x}_1$

$$x_0 \sim p_0(x)$$
$$x_1 = \hat{x}_1$$

$$v(x_t, t \,|\, \hat{x}_1) = \frac{\hat{x}_1 - x_t}{1 - t}$$

$$\frac{d}{dt} p_t(x \,|\, \hat{x}_1) = -\nabla_x \cdot (p_t(x \,|\, \hat{x}_1) v(x \,|\, t, \hat{x}_1))$$

$$p_t(x \,|\, \hat{x}_1) = N(x \,|\, t\hat{x}_1, (1-t)^2 I)$$

- A vector field corresponding to a probability flow $p_t(x) = \int p_t(x \,|\, x_1) q(x_1) dx_1$ would give us the right target distribution since at t=1 $p_1(x \,|\, x_1) = \delta(x - x_1)$ and

$$p_1(x) = \int p_1(x \,|\, x_1) q(x_1) dx_1 = \int \delta(x - x_1) q(x_1) dx_1 = q(x)$$

Justification

- Let's take the single point continuity equation and integrate both sides over x_1 with respect to the data distribution $q(x)$

$$\int q(x_1)\frac{d}{dt}p_t(x \mid x_1)dx_1 = -\int q(x_1)\,\nabla_x \cdot (p_t(x \mid x_1)v(x \mid t, x_1))dx_1$$

Justification

- Let's take the single point continuity equation and integrate both sides over x_1 with respect to the data distribution $q(x)$

$$\int q(x_1)\frac{d}{dt}p_t(x|x_1)dx_1 = -\int q(x_1)\,\nabla_x \cdot (p_t(x|x_1)v(x|t,x_1))dx_1$$

$$\frac{d}{dt}\int q(x_1)p_t(x|x_1)dx_1 = -\nabla_x \cdot \left(\int q(x_1)p_t(x|x_1)v(x|t,x_1)dx_1\right)$$

Justification

- Let's take the single point continuity equation and integrate both sides over x_1 with respect to the data distribution $q(x)$

$$\int q(x_1)\frac{d}{dt}p_t(x|x_1)dx_1 = -\int q(x_1)\,\nabla_x \cdot (p_t(x|x_1)v(x|t,x_1))dx_1$$

$$\frac{d}{dt}\int q(x_1)p_t(x|x_1)dx_1 = -\nabla_x \cdot \left(\int q(x_1)p_t(x|x_1)v(x|t,x_1)dx_1\right)$$

$$\frac{d}{dt}p_t(x) = -\nabla_x \cdot \left(p_t(x)\int \frac{q(x_1)p_t(x|x_1)}{p_t(x)}v(x|t,x_1)dx_1\right)$$

Justification

- Let's take the single point continuity equation and integrate both sides over x_1 with respect to the data distribution $q(x)$

$$\int q(x_1) \frac{d}{dt} p_t(x \mid x_1) dx_1 = -\int q(x_1) \nabla_x \cdot (p_t(x \mid x_1) v(x \mid t, x_1)) dx_1$$

$$\frac{d}{dt} \int q(x_1) p_t(x \mid x_1) dx_1 = -\nabla_x \cdot \left(\int q(x_1) p_t(x \mid x_1) v(x \mid t, x_1) dx_1 \right)$$

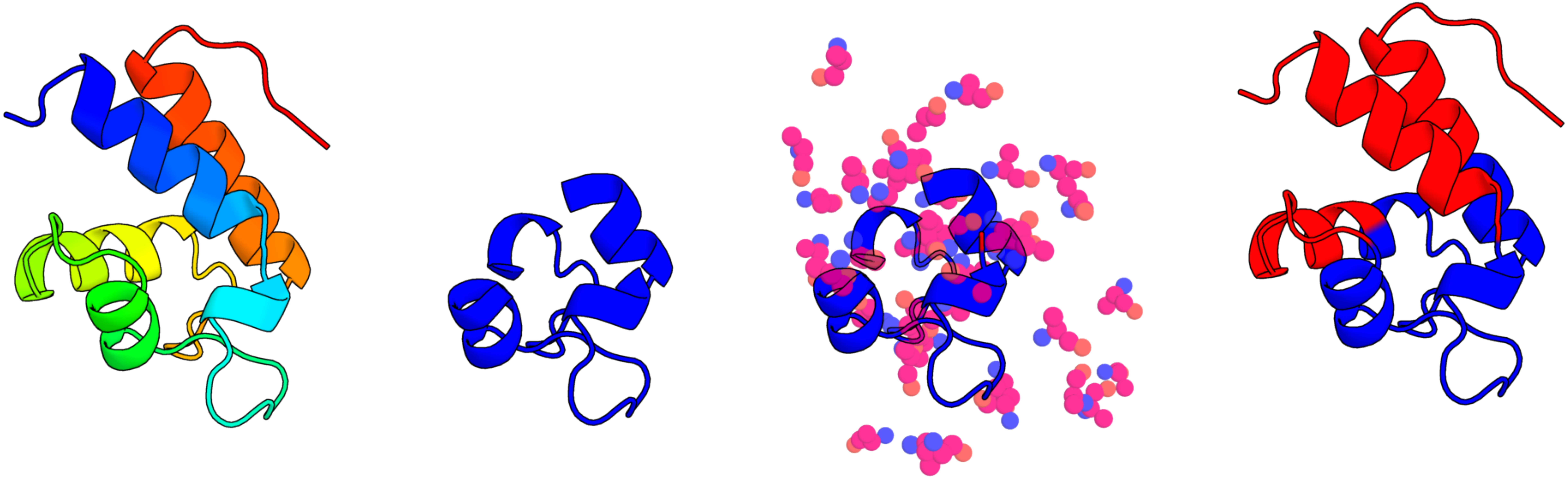
$$\frac{d}{dt} p_t(x) = -\nabla_x \cdot \left(p_t(x) \int \frac{q(x_1) p_t(x \mid x_1)}{p_t(x)} v(x \mid t, x_1) dx_1 \right)$$

- The vector field that gives the right probability flow is the conditional expectation:

$$v(x, t) = \int p_t(x_1 \mid x) v(x \mid t, x_1) dx_1 = E\left\{ v(x \mid x_1, t) \mid x, t \right\} = E\left\{ \frac{(x_1 - x)}{1 - t} \mid x, t \right\}$$

Example: Motif-scaffolding training

1. Take PDB structure.

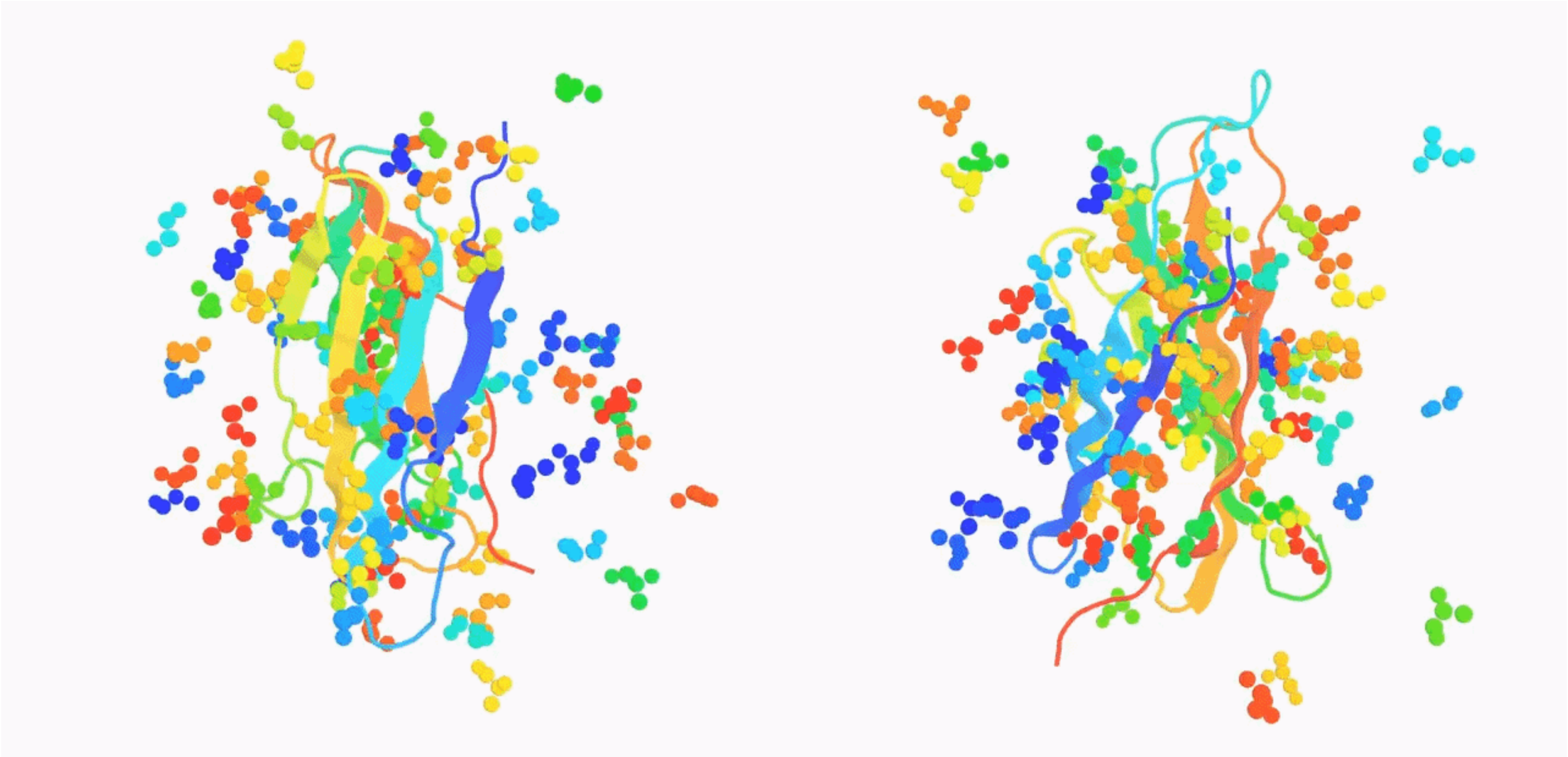


2. Select motif with cropping strategy
3. Noise scaffold.
4. Train FrameFlow to denoise

[Yim et al. 2024]

Diffusion vs Flow

- Example: molecular motif scaffolding



diffusion SDE

flow

[Jason Yim 2024]

Additional (optional) reading

- Bishop et al. “Deep Learning”, chapter 18
- Lipmann et al., “Flow Matching for Generative Modeling”, https://arxiv.org/pdf/2210.02747
- Albergo et al., “Stochastic Interpolants: A Unifying Framework for Flows and Diffusions”, https://arxiv.org/abs/2303.08797
-