Probabilistic Modeling with Flows

Deep Probabilistic Modeling with Flows

Amirabbas Asadi, Sajjad Heydari Nejad

Course instructors : Dr. Kasra Alishahi & Dr. Ehsan Mousavi

Sharif University of Technology
Department of Mathematical Sciences

February 2024

Density Estimation

You've probably seen lots of generative models

Density Estimation

You've probably seen lots of generative models

GAN, Diffusion, Autoregressive(e.g. LLMs), VAE, Flows ...

Density Estimation

What do they have in common?

Density Estimation

What do they have in common?

What is the difficulty of generative modeling?

Density Estimation

What do they have in common?

What is the difficulty of generative modeling?

More importantly...

How to design a new generative model?

Density Estimation

Today we try to touch the nature of generative modeling

by going through a fundamental and old problem

Density Estimation

Today we try to touch the nature of generative modeling

by going through a fundamental and old problem

Density Estimation

Density Estimation

We will also discuss an interesting generative model

Density Estimation

We will also discuss an interesting generative model

Normalizing Flow

Density Estimation

We will also discuss an interesting generative model

Normalizing Flow

That offers:

Density Estimation

We will also discuss an interesting generative model

Normalizing Flow

That offers:

- Density estimation for continuous data

Density Estimation

We will also discuss an interesting generative model

Normalizing Flow

That offers:

- Density estimation for continuous data
- A normalized distribution and exact likelihood computation

Density Estimation

We will also discuss an interesting generative model

Normalizing Flow

That offers:

- Density estimation for continuous data
- A normalized distribution and exact likelihood computation
- Efficient and straightforward way to sample i.i.d data

Density Estimation

Density Estimation is a difficult problem!

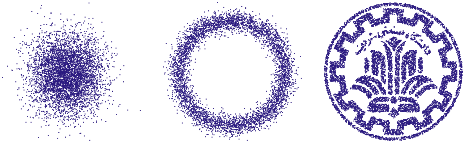
Density Estimation is a difficult problem!

Density Estimation

Density Estimation is a difficult problem!

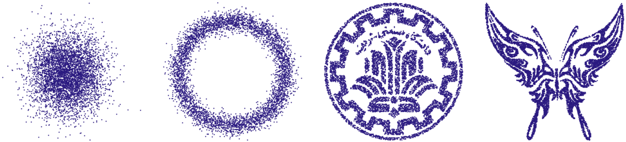
Density Estimation

Density Estimation is a difficult problem!



Density Estimation

Density Estimation is a difficult problem!



Normalizing Flow and Transport

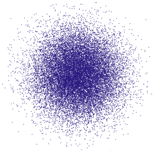
Basic idea:

Transforming a simple density p to a complicated target p^*

Normalizing Flow and Transport

Basic idea:

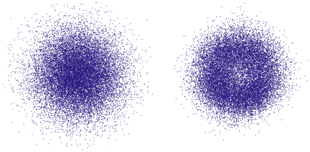
Transforming a simple density p to a complicated target p^*



Normalizing Flow and Transport

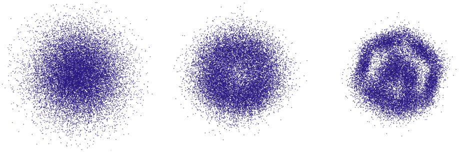
Basic idea:

Transforming a simple density p to a complicated target p^*



Basic idea:

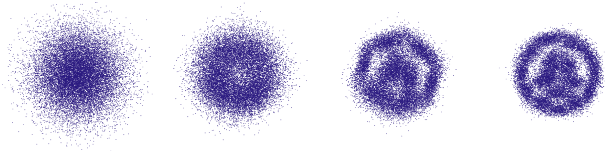
Transforming a simple density p to a complicated target p^*



Normalizing Flow and Transport

Basic idea:

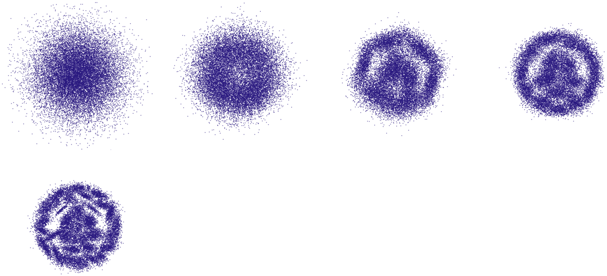
Transforming a simple density p to a complicated target p^*



Normalizing Flow and Transport

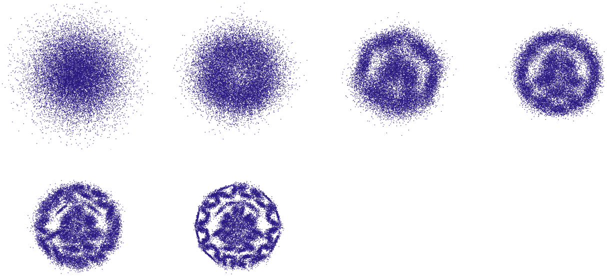
Basic idea:

Transforming a simple density p to a complicated target p^*



Basic idea:

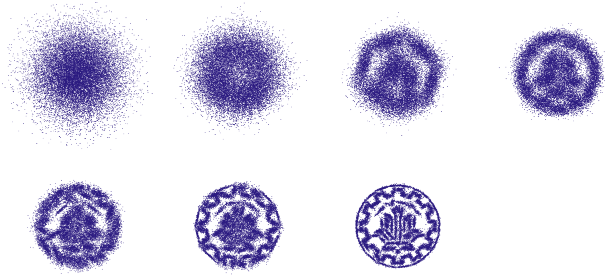
Transforming a simple density p to a complicated target p^*



Normalizing Flow and Transport

Basic idea:

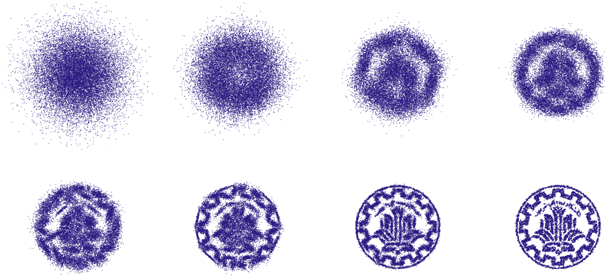
Transforming a simple density p to a complicated target p^*



Normalizing Flow and Transport

Basic idea:

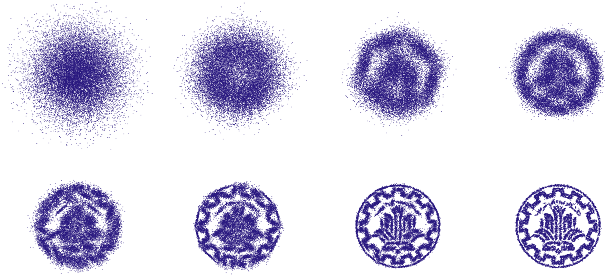
Transforming a simple density p to a complicated target p^*



Normalizing Flow and Transport

Basic idea:

Transforming a simple density p to a complicated target p^*



Is it possible!?

Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

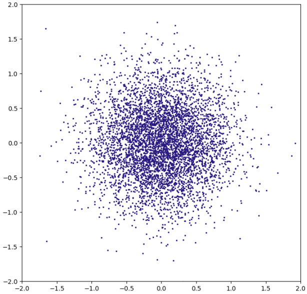
$$(y_1, y_2) = T(x_1, x_2) = (0.75x_1 + 0.4, 0.5x_2 - 0.2)$$

Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

$$(y_1, y_2) = T(x_1, x_2) = (0.75x_1 + 0.4, 0.5x_2 - 0.2)$$

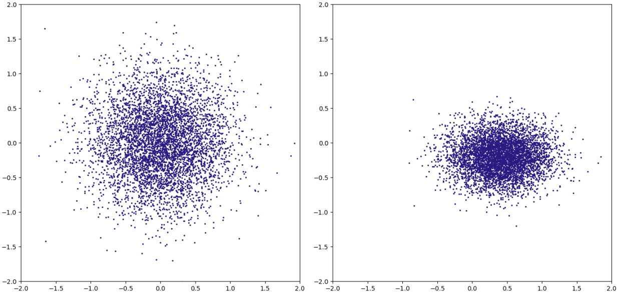


Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

$$(y_1, y_2) = T(x_1, x_2) = (0.75x_1 + 0.4, 0.5x_2 - 0.2)$$



Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

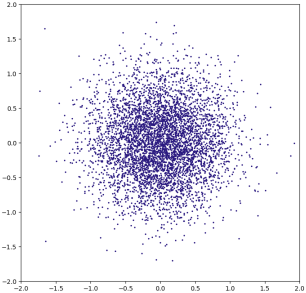
$$(y_1, y_2) = T(x_1, x_2) = (\tanh(x_1), \tanh(x_2))$$

Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

$$(y_1, y_2) = T(x_1, x_2) = (\tanh(x_1), \tanh(x_2))$$

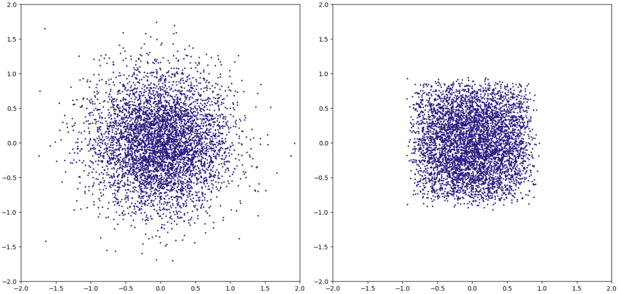


Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

$$(y_1, y_2) = T(x_1, x_2) = (\tanh(x_1), \tanh(x_2))$$



Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

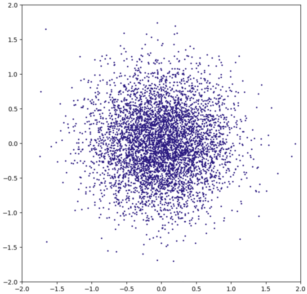
$$(y_1, y_2) = T(x_1, x_2) = (x_1^{\frac{1}{3}}, x_2^{\frac{1}{3}})$$

Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

$$(y_1, y_2) = T(x_1, x_2) = (x_1^{\frac{1}{3}}, x_2^{\frac{1}{3}})$$

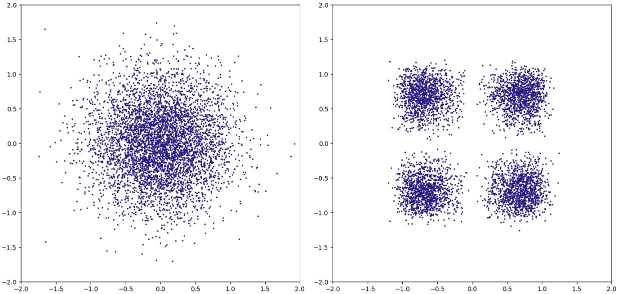


Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

$$(y_1, y_2) = T(x_1, x_2) = (x_1^{\frac{1}{3}}, x_2^{\frac{1}{3}})$$



Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

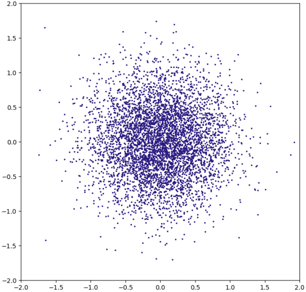
$$(y_1, y_2) = T(x_1, x_2) = (x_1(\frac{1}{|x|} + 0.4), x_2(\frac{1}{|x|} + 0.4))$$

Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

$$(y_1, y_2) = T(x_1, x_2) = (x_1(\frac{1}{|x|} + 0.4), x_2(\frac{1}{|x|} + 0.4))$$

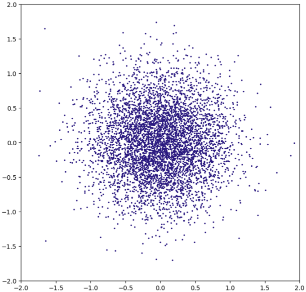


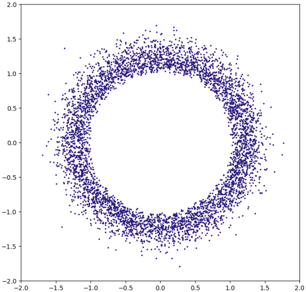
Example

$$T : X \to Y$$

$$X \sim \mathcal{N}(\mu, \Sigma)$$

$$(y_1, y_2) = T(x_1, x_2) = (x_1(\frac{1}{|x|} + 0.4), x_2(\frac{1}{|x|} + 0.4))$$





Transport

How far can we go in this way?

Transport

How far can we go in this way?

Can we obtain any density by transforming a simple one?

Transport

How far can we go in this way?

Can we obtain any density by transforming a simple one?

Under reasonable assumptions, such a transformation always exists!

Change of Variable (Discrete Case)

Okay but how to find the transformed density?

$$p_y(y) = p_x(T^{-1}(y))|\det J_{T^{-1}}(y)|$$

Change of Variable

We can find the optimal parameters of the transformation by solving a standard likelihood maximization problem.

Change of Variable

We can find the optimal parameters of the transformation by solving a standard likelihood maximization problem.

$$\log p_y(y) = \log p_x(T^{-1}(y)) + \log |\det J_{T^{-1}}(y)|$$

Change of Variable

We can find the optimal parameters of the transformation by solving a standard likelihood maximization problem.

$$\log p_y(y) = \log p_x(T^{-1}(y)) + \log |\det J_{T^{-1}}(y)|$$

But how to choose T?

Change of Variable

We can find the optimal parameters of the transformation by solving a standard likelihood maximization problem.

$$\log p_y(y) = \log p_x(T^{-1}(y)) + \log |\det J_{T^{-1}}(y)|$$

But how to choose T?

Can we just choose some neural network?

Change of Variable

We can find the optimal parameters of the transformation by solving a standard likelihood maximization problem.

$$\log p_y(y) = \log p_x(T^{-1}(y)) + \log |\det J_{T^{-1}}(y)|$$

But how to choose T?

Can we just choose some neural network?

Unfortunately NO!

Change of Variable

In practice, We have to find a T such that:

Change of Variable

In practice, We have to find a T such that:

- T is invertible

Change of Variable

In practice, We have to find a T such that:

- T is invertible
- $|\det J_{T^{-1}}(\cdot)|$ is efficient to compute

Change of Variable

In practice, We have to find a T such that:

- T is invertible
- $|\det J_{T^{-1}}(\cdot)|$ is efficient to compute

Only simple transformations satisfy the above conditions.

Example: Coupling Flow

Consider a simple invertible map like

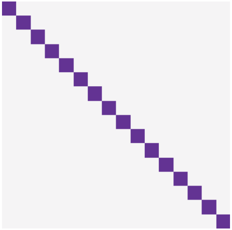
$$h((x_1, x_2, ..., x_d)) = (a_1 x_1 + b_1, ..., a_d x_d + b_d)$$

Example: Coupling Flow

Consider a simple invertible map like

$$h((x_1, x_2, ..., x_d)) = (a_1 x_1 + b_1, ..., a_d x_d + b_d)$$

In this case the Jacobian looks like

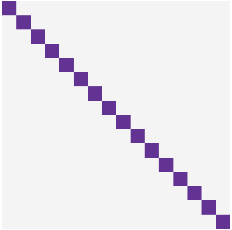


Example: Coupling Flow

Consider a simple invertible map like

$$h((x_1, x_2, ..., x_d)) = (a_1 x_1 + b_1, ..., a_d x_d + b_d)$$

In this case the Jacobian looks like



Why such a map is not so useful?

Example: Coupling Flow

Now define h as before:

$$h((x_1, x_2, ..., x_d); \theta) = (\theta_{11} x_1 + \theta_{12}, ..., \theta_{d1} x_d + \theta_{d2})$$

Take some partition of dimensions, like:

$$A : \{x_1, x_2, ..., x_m\}, B : \{x_{m+1}, ..., x_d\}$$

Define a map parametrized by θ in the following way:

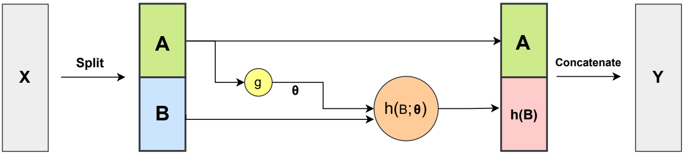
$\theta = g(x_1, ..., x_m)$

$T(x_1, x_2, ..., x_m) = x_1, x_2, ..., x_m$

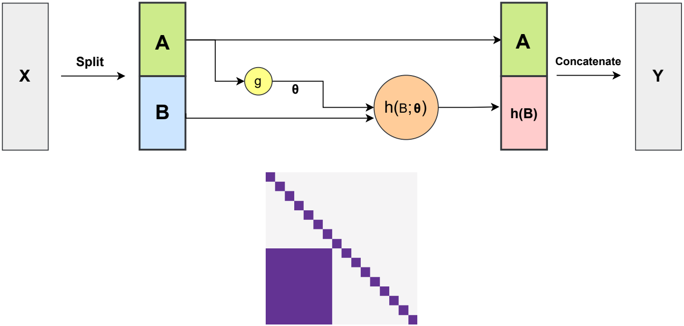
$T(x_{m+1}, ..., x_d) = h(x_{m+1}, ..., x_d; \theta)$

Is T invertible?

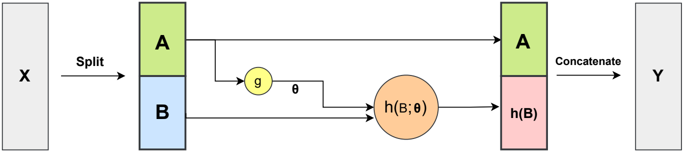
Example: Coupling Flow

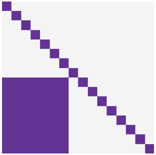


Example: Coupling Flow



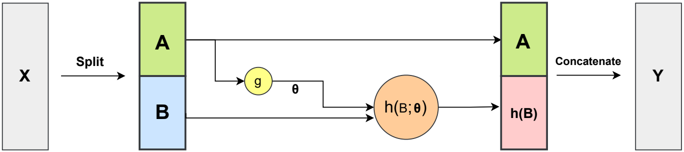
Example: Coupling Flow

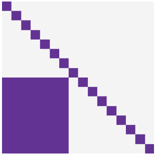




g can be any function! like any neural network!

Example: Coupling Flow



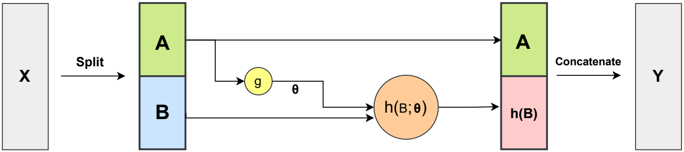


g can be any function! like any neural network!

Coupling provides an elegant framework to design invertible maps

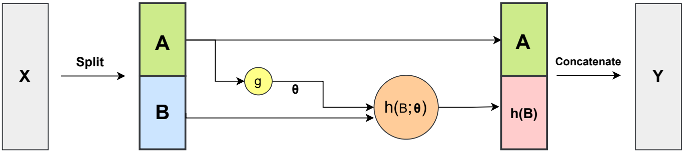
Example: Coupling Flow

What's the problem with coupling?



Example: Coupling Flow

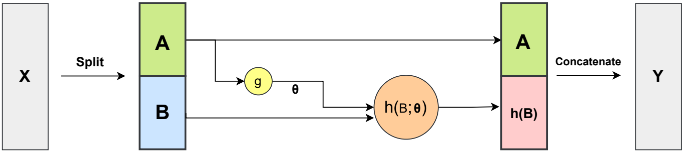
What's the problem with coupling?



It doesn't touch half of the variables!!

Example: Coupling Flow

What's the problem with coupling?



It doesn't touch half of the variables!!
So we can do it many times on different partitions!

$$T = T_1 \circ T_2 \circ ... \circ T_n$$

Summary

It's hard to find a expressive map T such that:

Summary

It's hard to find a expressive map T such that:

- T is invertible

Summary

It's hard to find a expressive map T such that:

- T is invertible
- $|\det J_{T^{-1}}(\cdot)|$ is efficient to compute

Summary

It's hard to find a expressive map T such that:

- T is invertible
- $|\det J_{T^{-1}}(\cdot)|$ is efficient to compute

Only simple maps can satisfy such hard contraints.

Summary

It's hard to find a expressive map T such that:

- T is invertible
- $|\det J_{T^{-1}}(\cdot)|$ is efficient to compute

Only simple maps can satisfy such hard contraints.

So we combine lots of them to construct a rich transformation.

Summary

It's hard to find a expressive map T such that:

- T is invertible
- $|\det J_{T^{-1}}(\cdot)|$ is efficient to compute

Only simple maps can satisfy such hard contraints.

So we combine lots of them to construct a rich transformation.

Is there any other way?

Deterministic Continuous Transformations

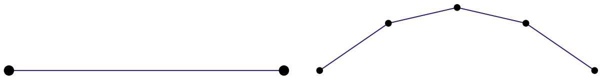
Let's increase the number of transformations while making them simpler...

Deterministic Continuous Transformations

Let's increase the number of transformations while making them simpler...

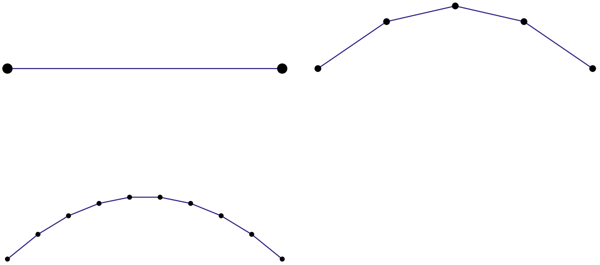
Deterministic Continuous Transformations

Let's increase the number of transformations while making them simpler...



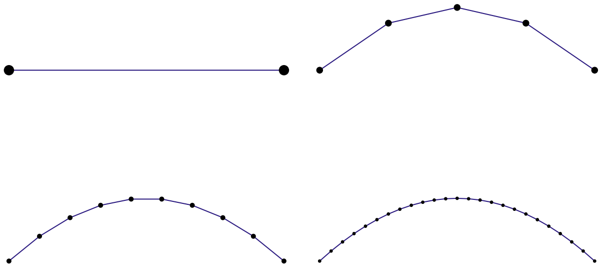
Deterministic Continuous Transformations

Let's increase the number of transformations while making them simpler...



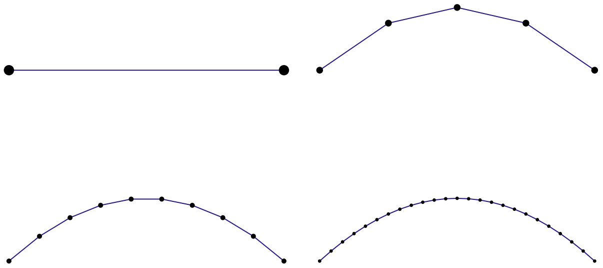
Deterministic Continuous Transformations

Let's increase the number of transformations while making them simpler...



Deterministic Continuous Transformations

Let's increase the number of transformations while making them simpler...



$$N \to \infty \ !??$$

Deterministic Continuous Transformations

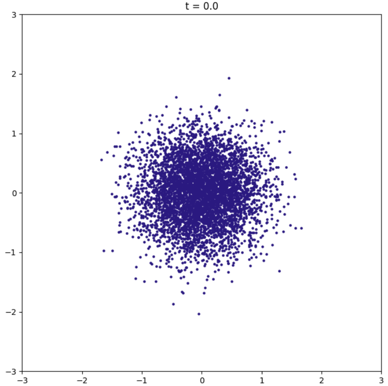
Now we define the transformed particles with a differential equation

$$dX_t = f(X_t, t)\, dt$$

Example

$$dX_t = -X_t \, dt$$

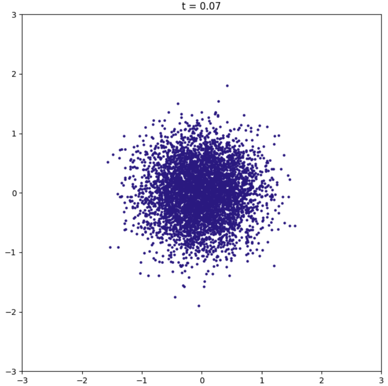
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t \, dt$$

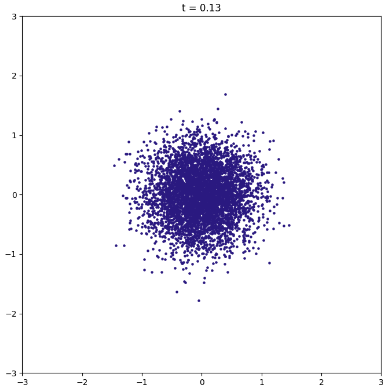
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t \, dt$$

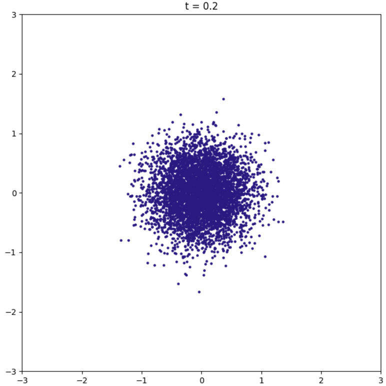
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t\, dt$$

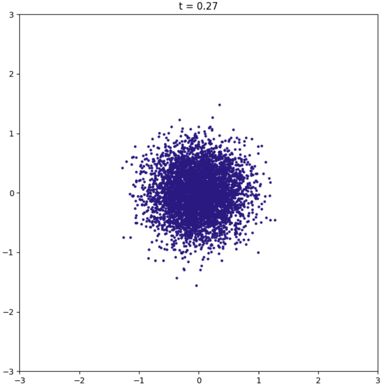
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t \, dt$$

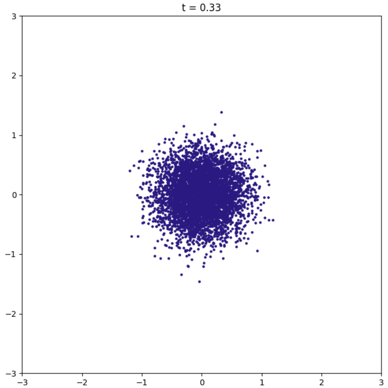
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t \, dt$$

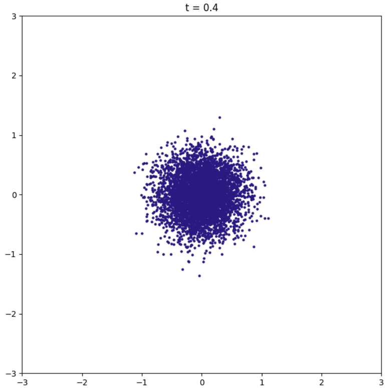
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t \, dt$$

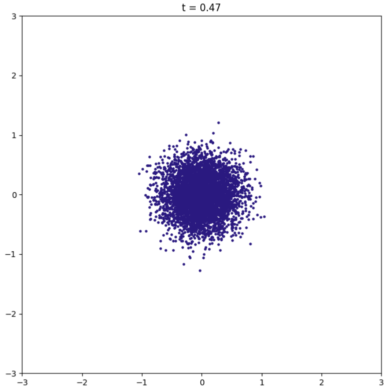
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t \, dt$$

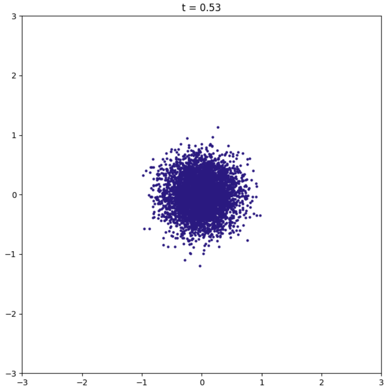
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t\,dt$$

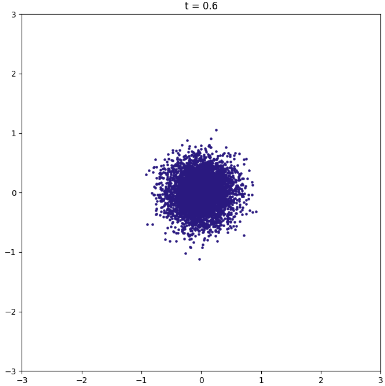
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t\, dt$$

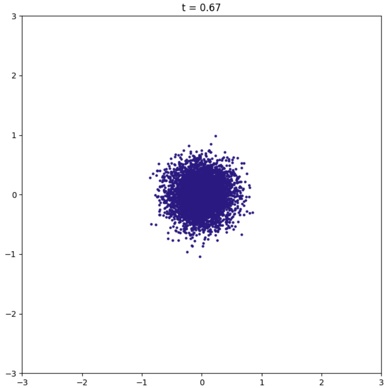
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t\, dt$$

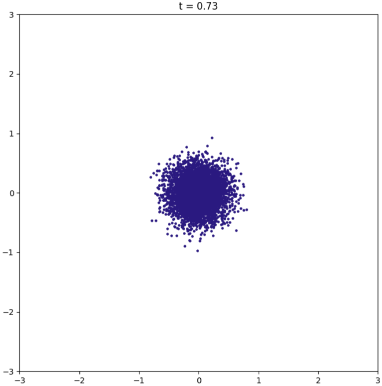
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t\,dt$$

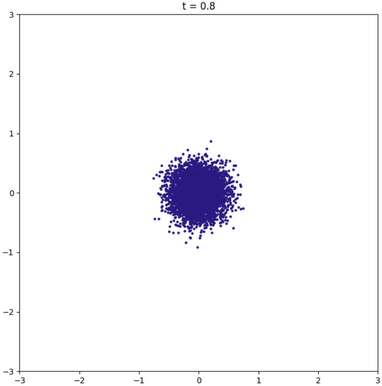
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t\,dt$$

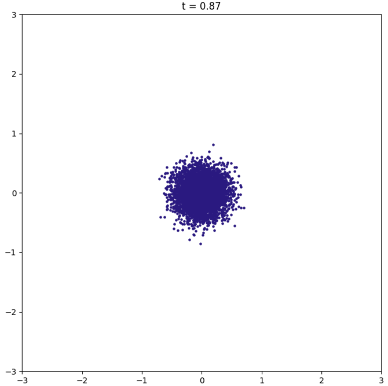
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t \, dt$$

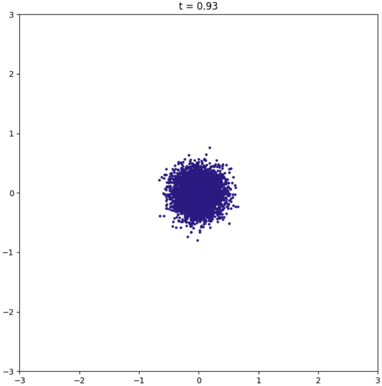
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t\,dt$$

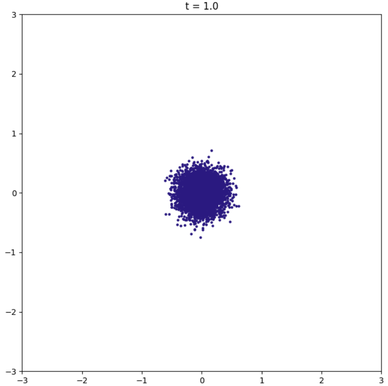
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -X_t\,dt$$

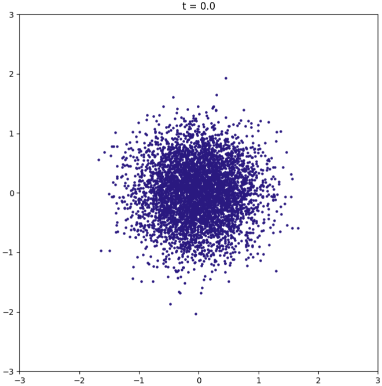
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\, dt$$

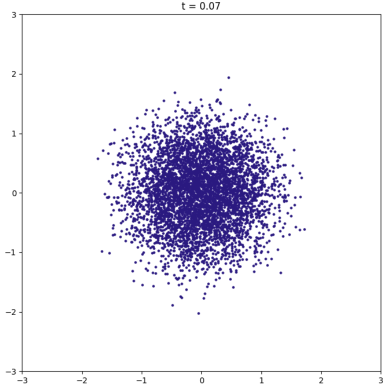
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

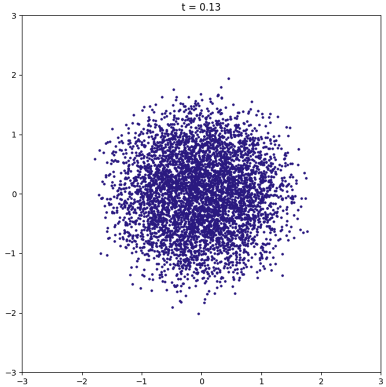
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\, dt$$

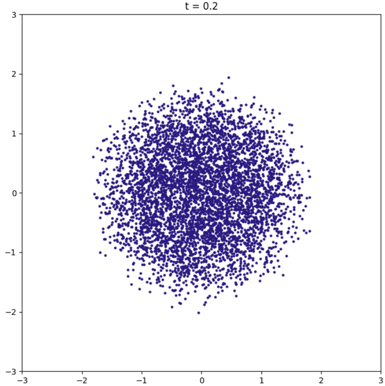
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

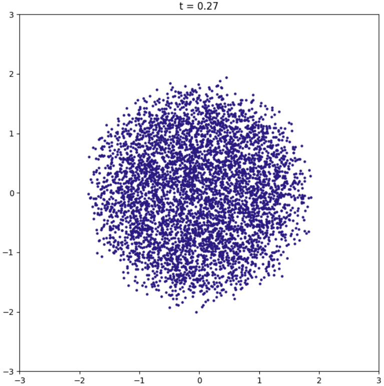
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\, dt$$

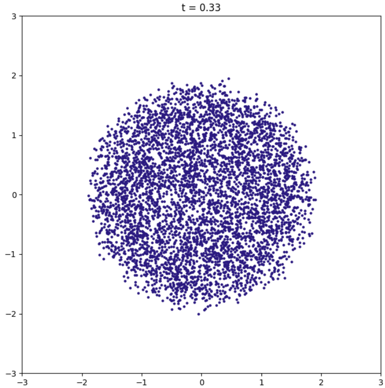
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

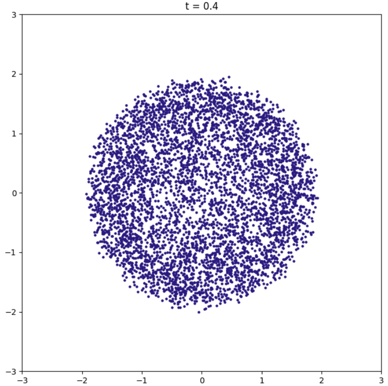
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

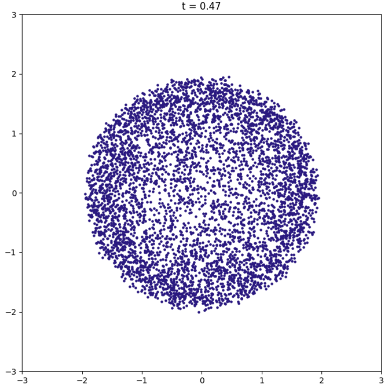
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

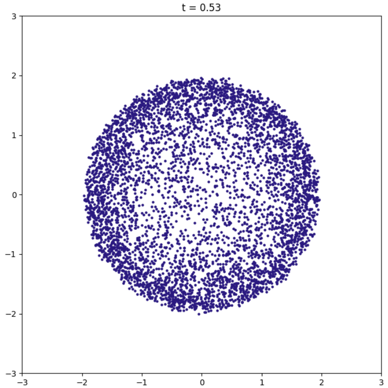
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

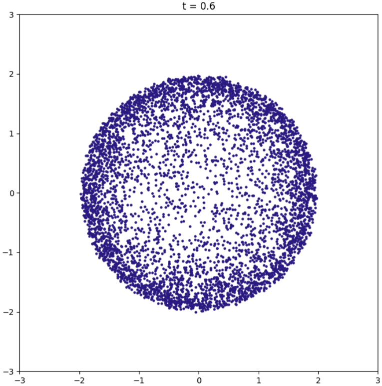
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\, dt$$

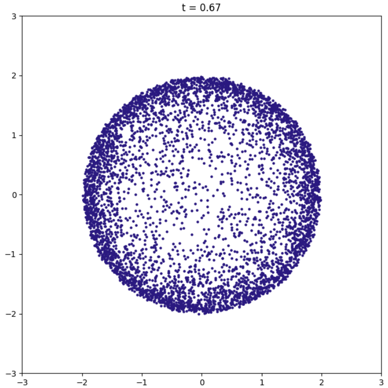
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

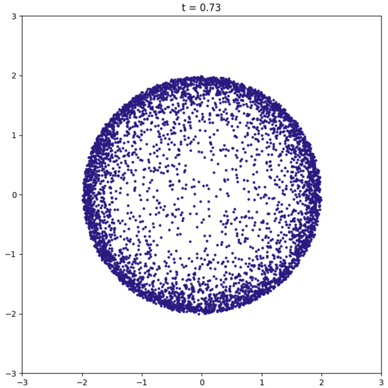
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\, dt$$

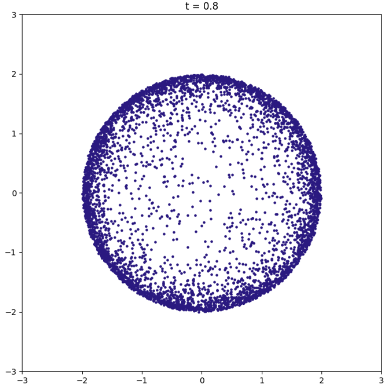
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$

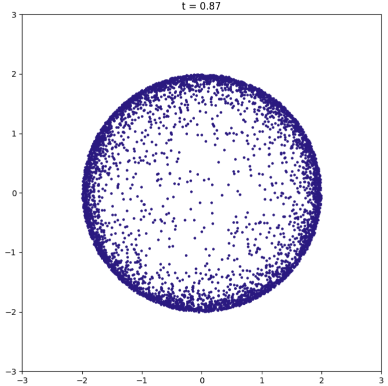


Example

Probabilistic Modeling with Flows

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

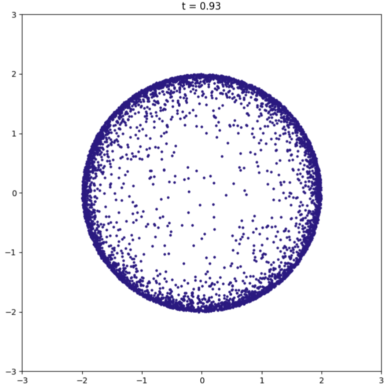
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

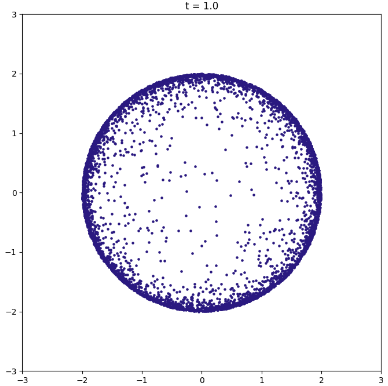
$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Example

$$dX_t = -0.7(||X_t||_2^2 - 4.0)\,dt$$

$$X_0 \sim \mathcal{N}(\mu, \Sigma)$$



Instantaneous Change of Variables Theorem

$$dX_t = f(X_t, t)\,dt$$

Neural Ordinary Differential Equations, arXiv:1806.07366 [cs.LG]

Instantaneous Change of Variables Theorem

$$dX_t = f(X_t, t)\, dt$$

$$T_\epsilon = X_{t+\epsilon}$$

Neural Ordinary Differential Equations, arXiv:1806.07366 [cs.LG]

Instantaneous Change of Variables Theorem

$$dX_t = f(X_t, t)\, dt$$

$$T_\epsilon = X_{t+\epsilon}$$

$$\frac{\partial \log p(x)}{\partial t} = \lim_{\epsilon \to 0^+} \frac{\log p(x) - \log \left|\det(\frac{\partial T_\epsilon}{\partial x})\right| - \log p(x)}{\epsilon}$$

Neural Ordinary Differential Equations, arXiv:1806.07366 [cs.LG]

Instantaneous Change of Variables Theorem

$$dX_t = f(X_t, t)\, dt$$

$$T_\epsilon = X_{t+\epsilon}$$

$$\frac{\partial \log p(x)}{\partial t} = \lim_{\epsilon \to 0^+} \frac{\log p(x) - \log \left|\det(\frac{\partial T_\epsilon}{\partial x})\right| - \log p(x)}{\epsilon}$$

$$\frac{\partial \log p(x)}{\partial t} = -\mathrm{Tr}(\frac{df}{dx})$$

Neural Ordinary Differential Equations, arXiv:1806.07366 [cs.LG]

Continuity equation

$$dX_t = f(X_t, t)\, dt$$

Then the density of the transformed rvs is the solution of:

$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial}{\partial x}[f(x,t)p(x,t)]$$

Neural ODE

$$dX_t = f(X_t, t)\, dt$$

Neural ODE

$$dX_t = f(X_t, t)\, dt$$

f can be a custom parameterized function e.g. a Neural Network

Neural ODE

$$dX_t = f(X_t, t)\, dt$$

f can be a custom parameterized function e.g. a Neural Network

A Neural Ordinary Differential Equation!

Neural ODE

$$dX_t = f(X_t, t)\, dt$$

f can be a custom parameterized function e.g. a Neural Network

A Neural Ordinary Differential Equation!

A Neural Network with continuous depth!

Neural ODE

$$dX_t = f(X_t, t)\, dt$$

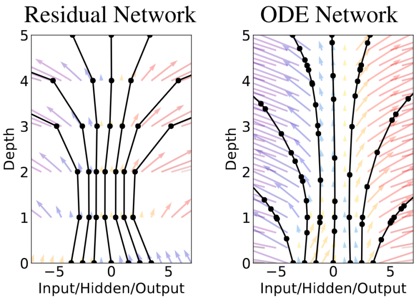
f can be a custom parameterized function e.g. a Neural Network

A Neural Ordinary Differential Equation!

A Neural Network with continuous depth!

But How can we train such a model?

Neural ODE



Continuous Depth needs Continuous Backpropagation!

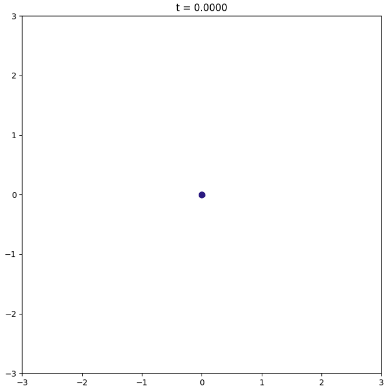
Neural Ordinary Differential Equations, arXiv:1806.07366 [cs.LG]

Deterministic Continuous Transformations

Let's see our last example of deterministic transports

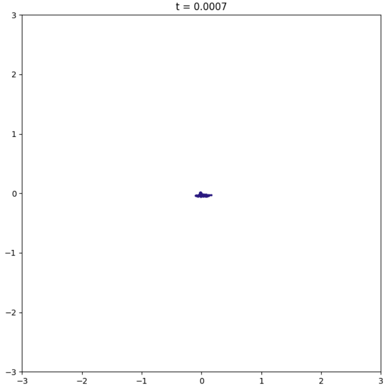
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t)dt$$



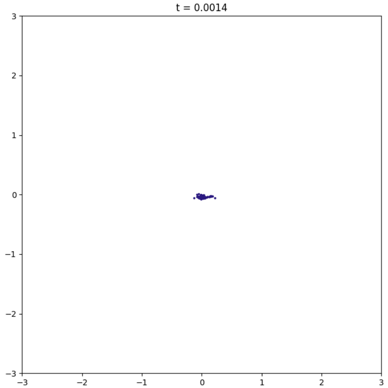
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



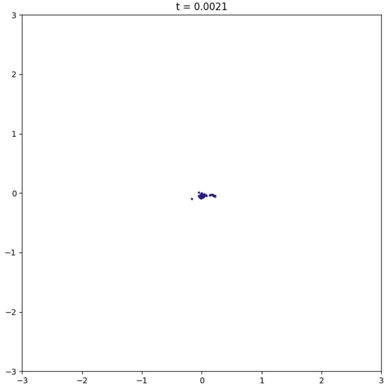
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



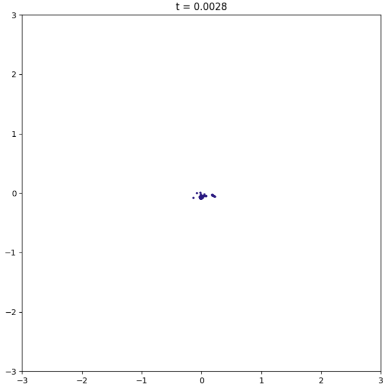
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t)dt$$



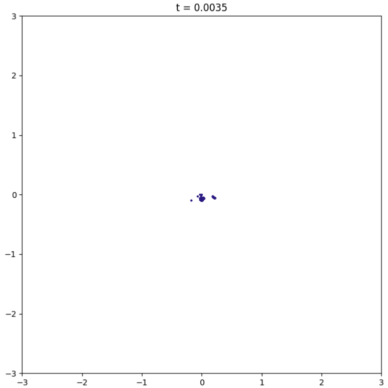
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



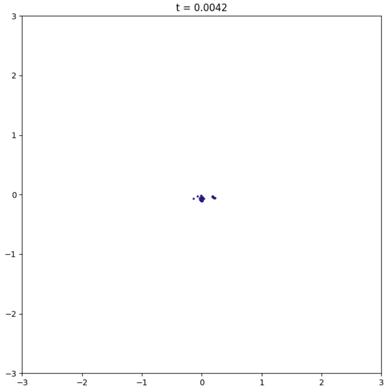
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



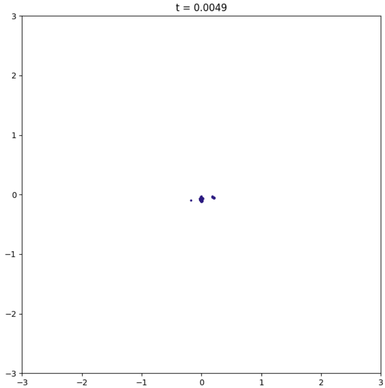
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



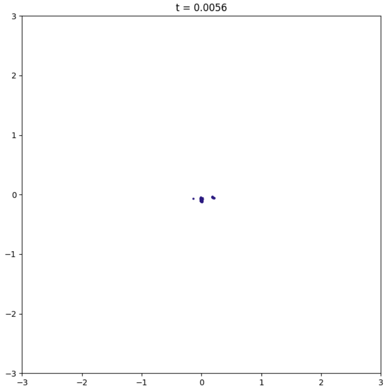
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



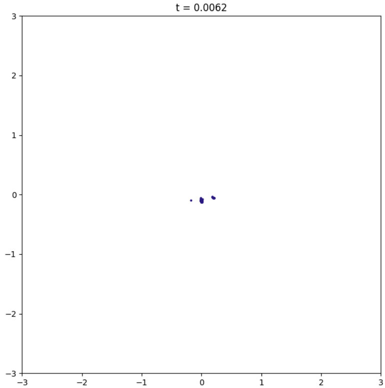
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



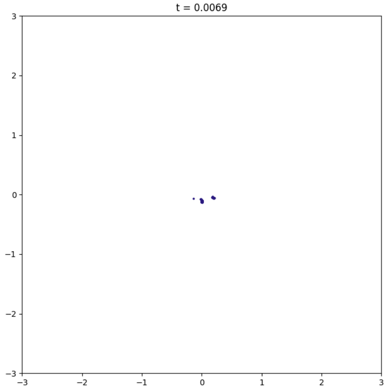
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t)dt$$



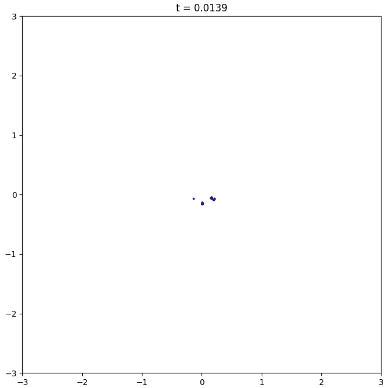
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



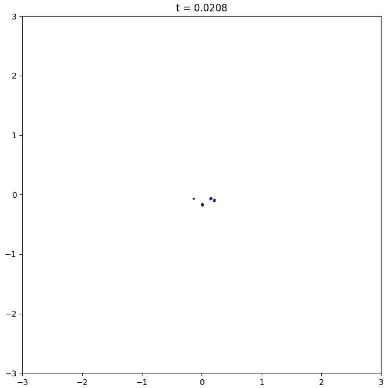
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



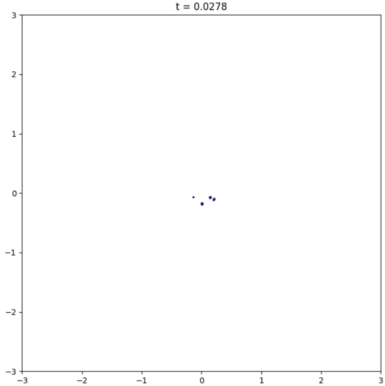
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



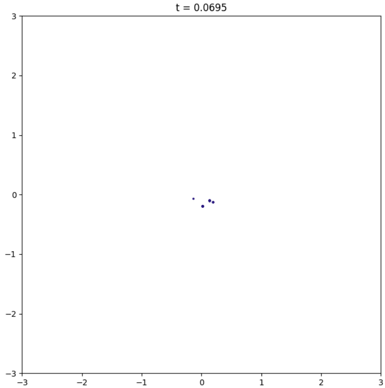
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



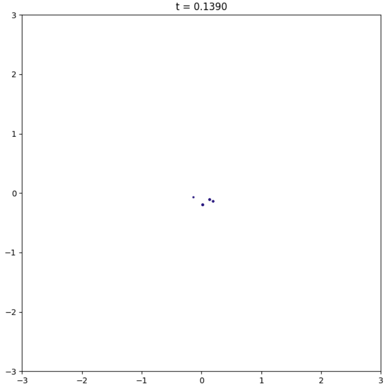
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t)dt$$



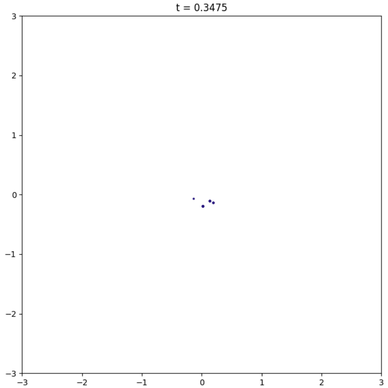
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



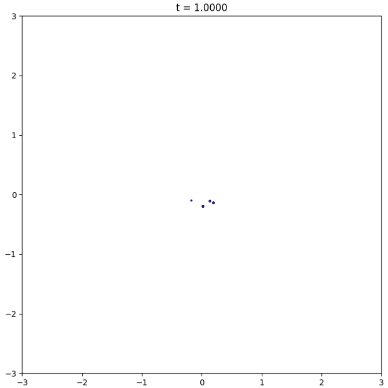
Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t)dt$$



Example: Gradient Flow

$$dX_t = \tau \nabla_x U(X_t) dt$$



Deterministic Continuous Transformations

What just happened!!?

Deterministic Continuous Transformations

What just happened!!?

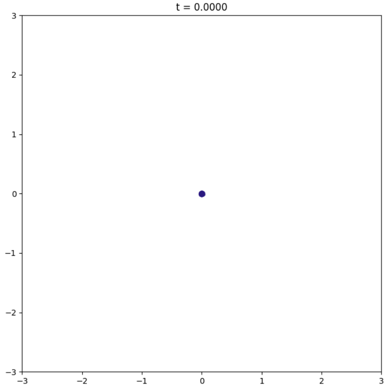
How to fix it?

Stochastic Continuous Transformations

Transport maps do not have to be deterministic!

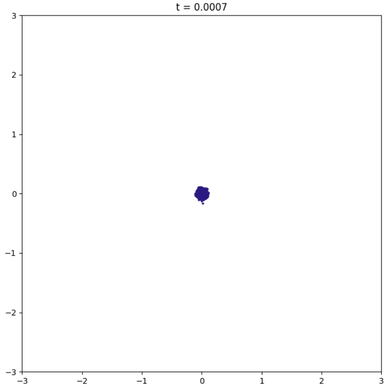
Example: Brownian Motion

$$dX_t = dW_t$$



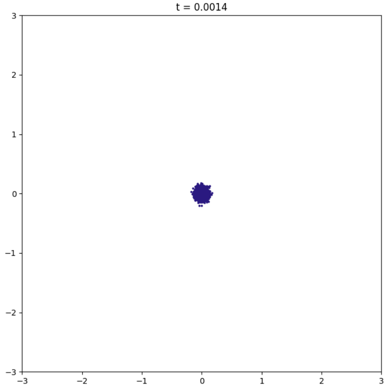
Example: Brownian Motion

$$dX_t = dW_t$$



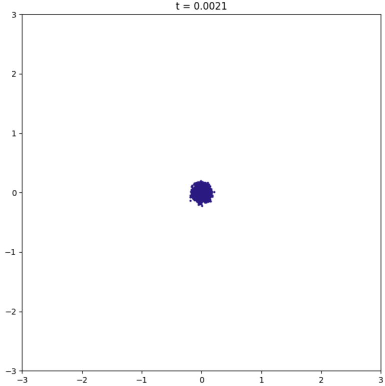
Example: Brownian Motion

$$dX_t = dW_t$$



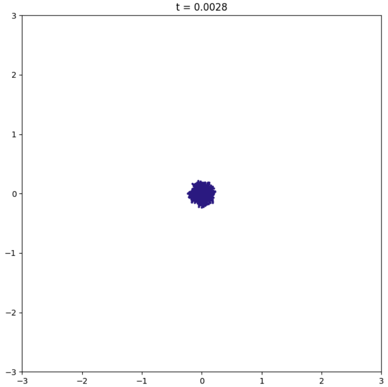
Example: Brownian Motion

$$dX_t = dW_t$$



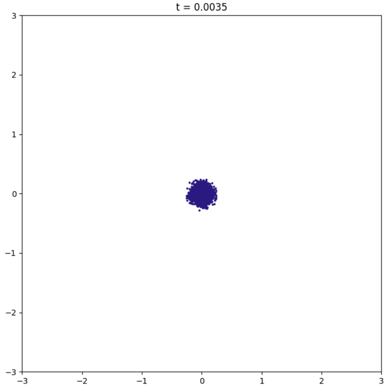
Example: Brownian Motion

$$dX_t = dW_t$$



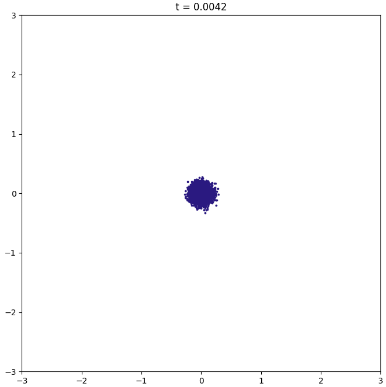
Example: Brownian Motion

$$dX_t = dW_t$$



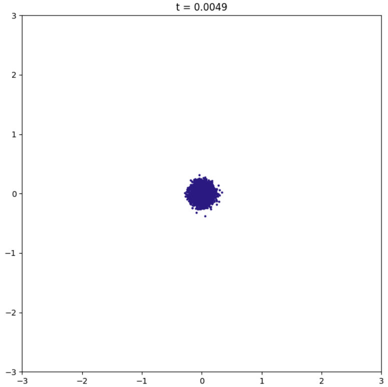
Example: Brownian Motion

$$dX_t = dW_t$$



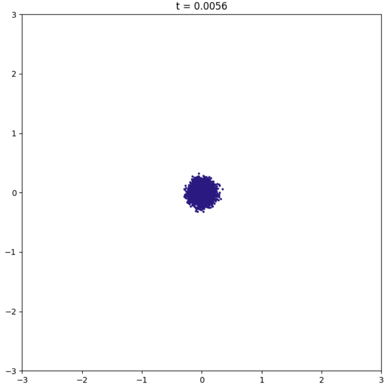
Example: Brownian Motion

$$dX_t = dW_t$$



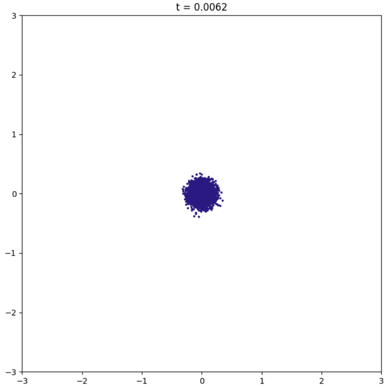
Example: Brownian Motion

$$dX_t = dW_t$$



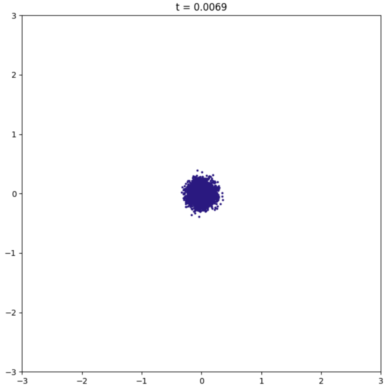
Example: Brownian Motion

$$dX_t = dW_t$$



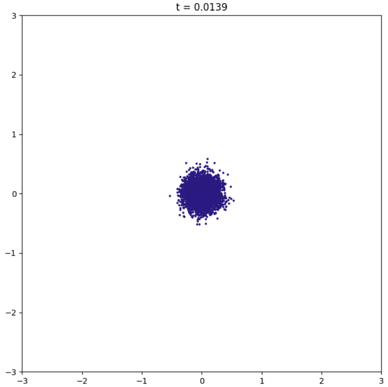
Example: Brownian Motion

$$dX_t = dW_t$$



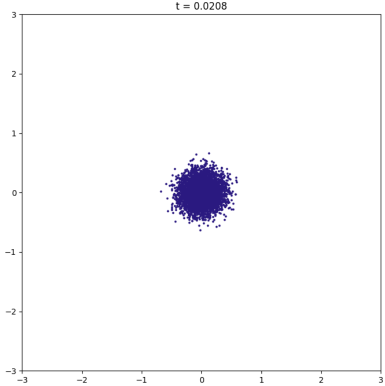
Example: Brownian Motion

$$dX_t = dW_t$$



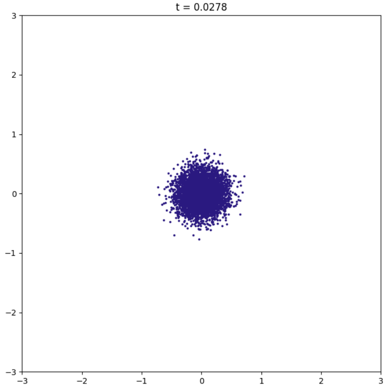
Example: Brownian Motion

$$dX_t = dW_t$$



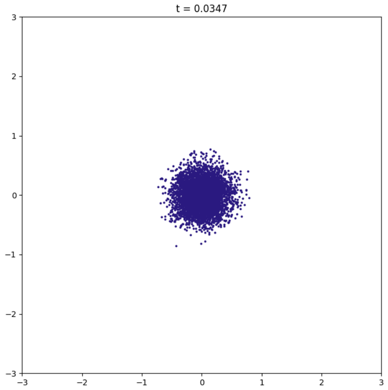
Example: Brownian Motion

$$dX_t = dW_t$$



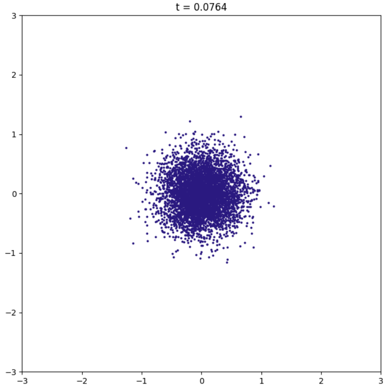
Example: Brownian Motion

$$dX_t = dW_t$$



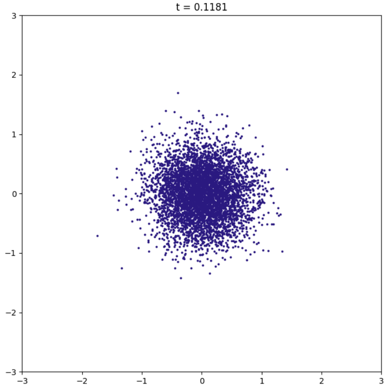
Example: Brownian Motion

$$dX_t = dW_t$$



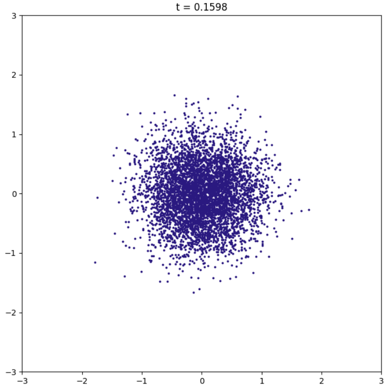
Example: Brownian Motion

$$dX_t = dW_t$$



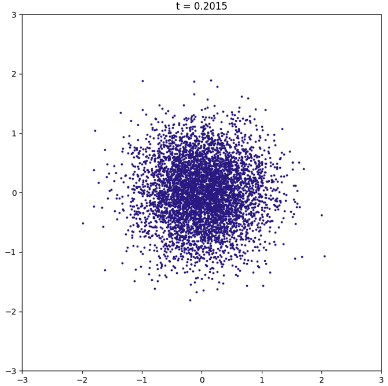
Example: Brownian Motion

$$dX_t = dW_t$$



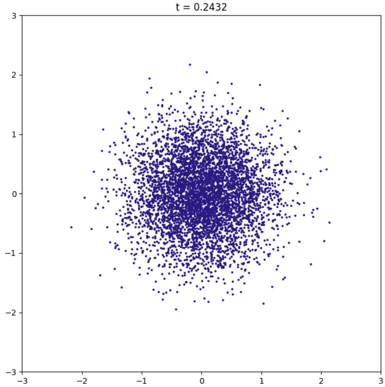
Example: Brownian Motion

$$dX_t = dW_t$$



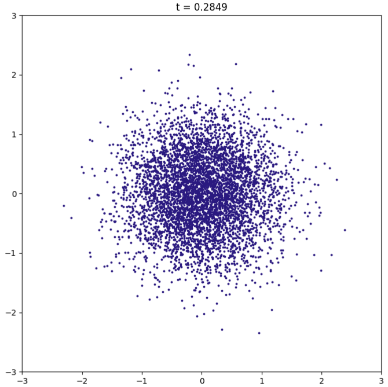
Example: Brownian Motion

$$dX_t = dW_t$$



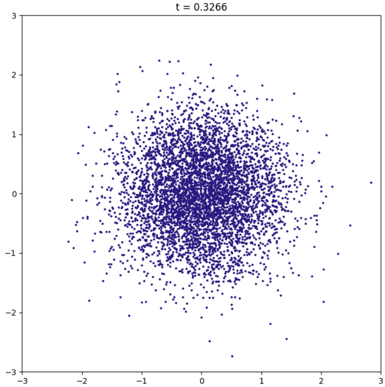
Example: Brownian Motion

$$dX_t = dW_t$$



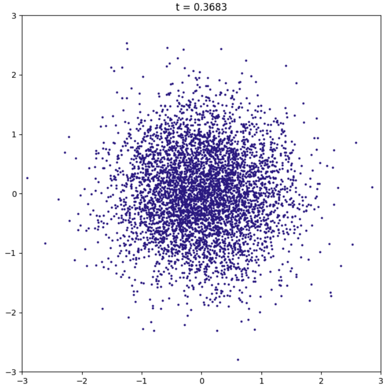
Example: Brownian Motion

$$dX_t = dW_t$$



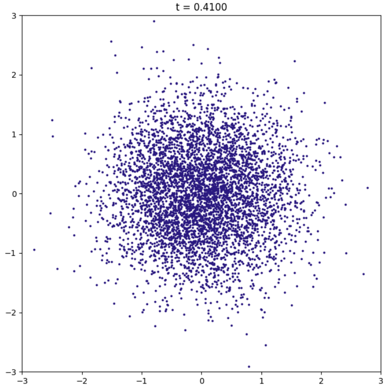
Example: Brownian Motion

$$dX_t = dW_t$$



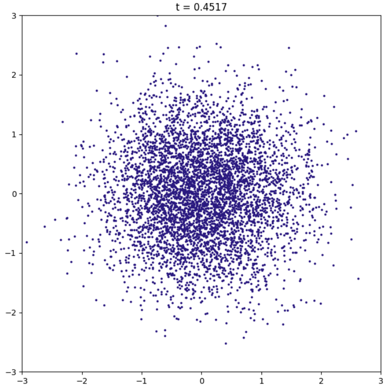
Example: Brownian Motion

$$dX_t = dW_t$$



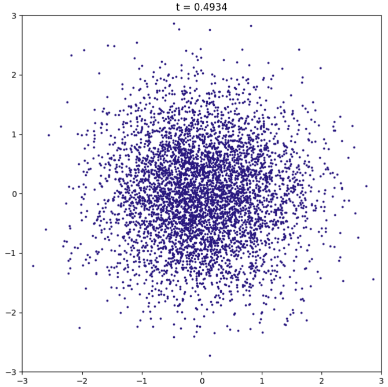
Example: Brownian Motion

$$dX_t = dW_t$$



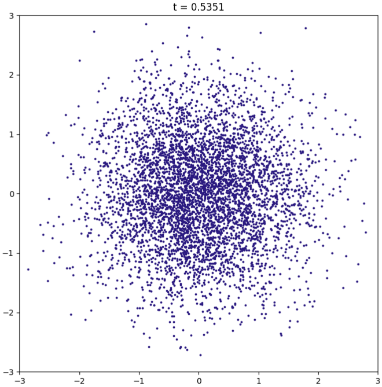
Example: Brownian Motion

$$dX_t = dW_t$$



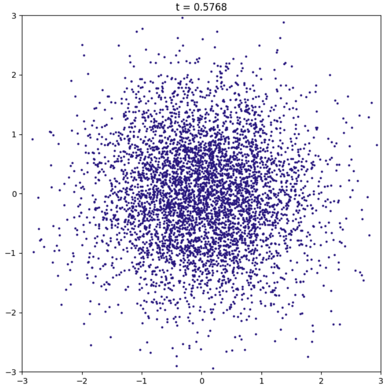
Example: Brownian Motion

$$dX_t = dW_t$$



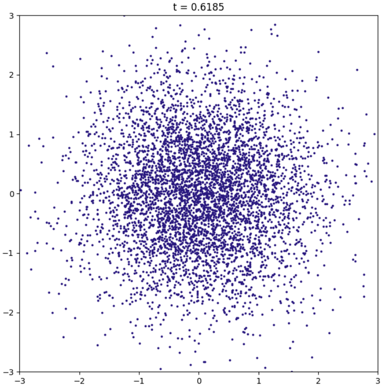
Example: Brownian Motion

$$dX_t = dW_t$$



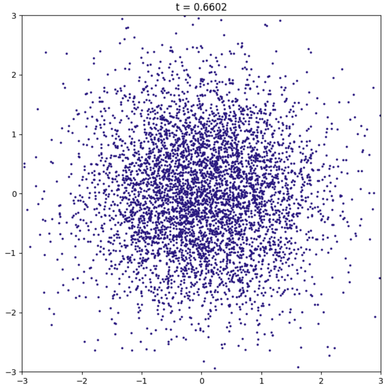
Example: Brownian Motion

$$dX_t = dW_t$$



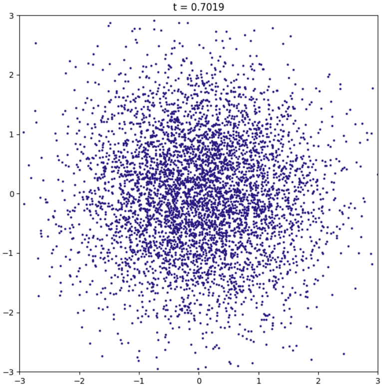
Example: Brownian Motion

$$dX_t = dW_t$$



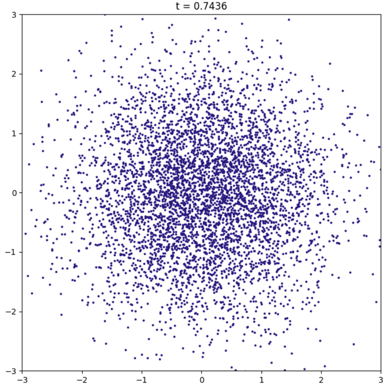
Example: Brownian Motion

$$dX_t = dW_t$$



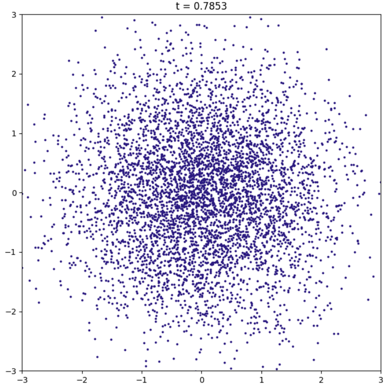
Example: Brownian Motion

$$dX_t = dW_t$$



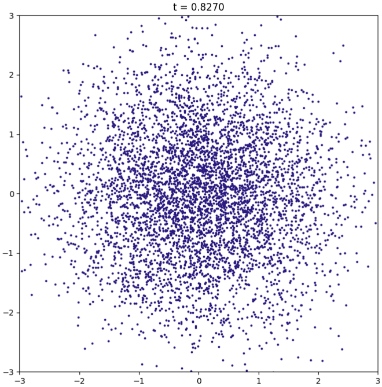
Example: Brownian Motion

$$dX_t = dW_t$$



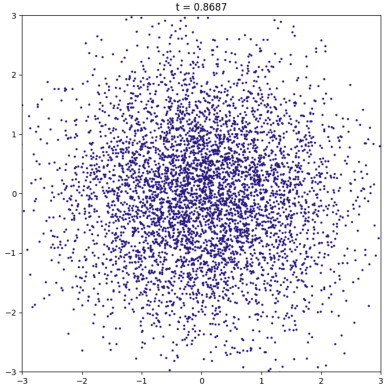
Example: Brownian Motion

$$dX_t = dW_t$$



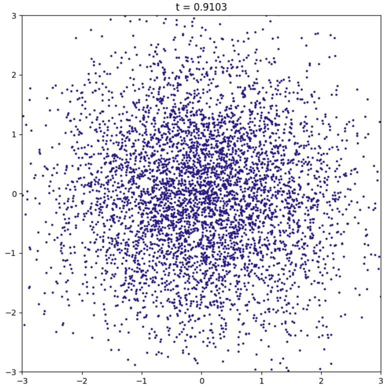
Example: Brownian Motion

$$dX_t = dW_t$$



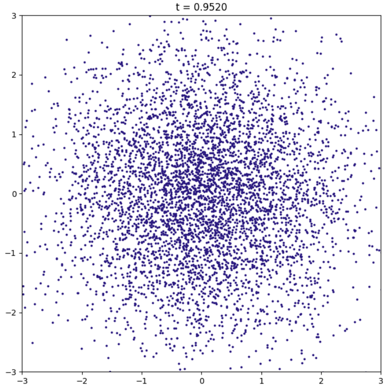
Example: Brownian Motion

$$dX_t = dW_t$$



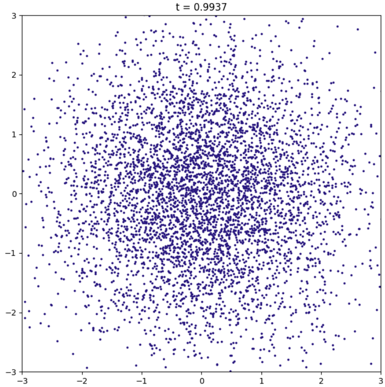
Example: Brownian Motion

$$dX_t = dW_t$$



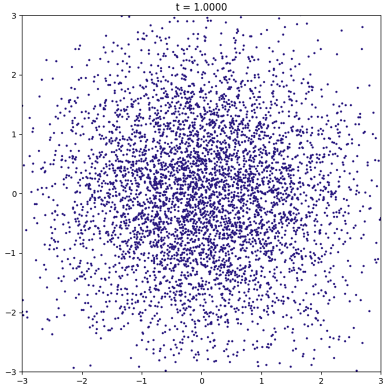
Example: Brownian Motion

$$dX_t = dW_t$$



Example: Brownian Motion

$$dX_t = dW_t$$



Stochastic Continuous Transformations

"Change of variables" for stochastic maps !!???

$$dX_t = dW_t$$

Stochastic Continuous Transformations

"Change of variables" for stochastic maps !!???

$$dX_t = dW_t$$

$$\frac{\partial p(x,t)}{\partial t} = \frac{\partial^2 p(x,t)}{\partial x^2}$$

Stochastic Continuous Transformations

So we have learned so far:

So we have learned so far:

$$dX_t = f(X_t, t)dt \rightarrow -\frac{\partial}{\partial x}[f(x,t)p(x,t)]$$

Stochastic Continuous Transformations

So we have learned so far:

$$dX_t = f(X_t, t)dt \to -\frac{\partial}{\partial x}[f(x,t)p(x,t)]$$

$$dX_t = dW_t \to \frac{\partial^2 p(x,t)}{\partial x^2}$$

Putting them together...In general we have

Stochastic Continuous Transformations

So we have learned so far:

$$dX_t = f(X_t, t)dt \rightarrow -\frac{\partial}{\partial x}[f(x,t)p(x,t)]$$

$$dX_t = dW_t \rightarrow \frac{\partial^2 p(x,t)}{\partial x^2}$$

Putting them together...In general we have

$$dX_t = f(X,t)dt + \sigma(X,t)dW_t$$

Stochastic Continuous Transformations

So we have learned so far:

$$dX_t = f(X_t, t)dt \rightarrow -\frac{\partial}{\partial x}[f(x,t)p(x,t)]$$

$$dX_t = dW_t \rightarrow \frac{\partial^2 p(x,t)}{\partial x^2}$$

Putting them together...In general we have

$$dX_t = f(X,t)dt + \sigma(X,t)dW_t$$

$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial}{\partial x}[f(x,t)p(x,t)] + \frac{\partial^2}{\partial x^2}[\sigma(x,t)p(x,t)]$$

Stochastic Continuous Transformations

So we have learned so far:

$$dX_t = f(X_t, t)dt \rightarrow -\frac{\partial}{\partial x}[f(x,t)p(x,t)]$$

$$dX_t = dW_t \rightarrow \frac{\partial^2 p(x,t)}{\partial x^2}$$

Putting them together...In general we have

$$dX_t = f(X,t)dt + \sigma(X,t)dW_t$$

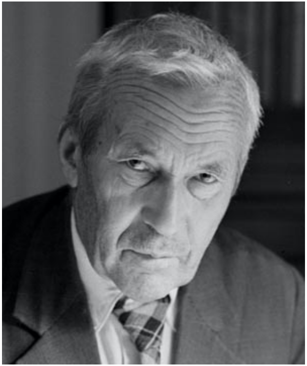
$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial}{\partial x}[f(x,t)p(x,t)] + \frac{\partial^2}{\partial x^2}[\sigma(x,t)p(x,t)]$$

Fokker-Planck Equation!

Stochastic Continuous Transformations

An even more general framework?

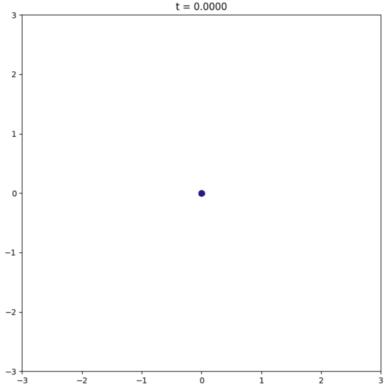
An even more general framework?



Now we get back to this one

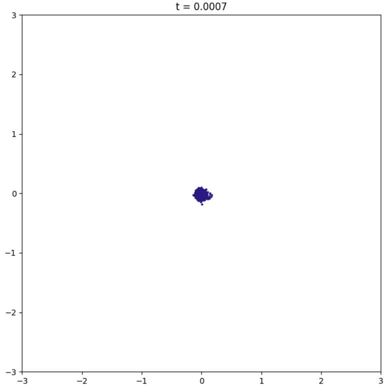
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



Example: Overdamped Langevin Dynamics

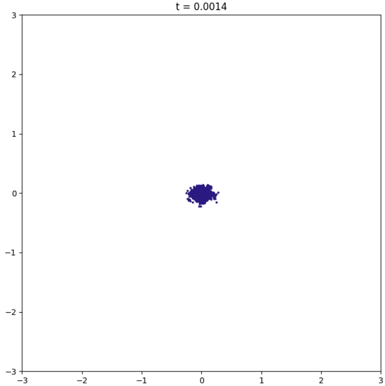
$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



Example: Overdamped Langevin Dynamics

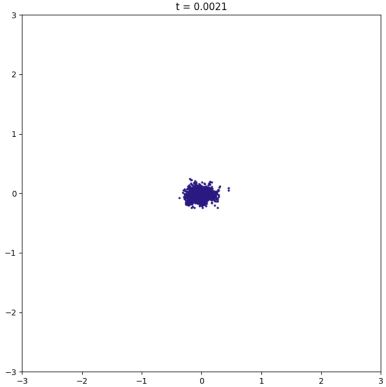
Probabilistic Modeling with Flows

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



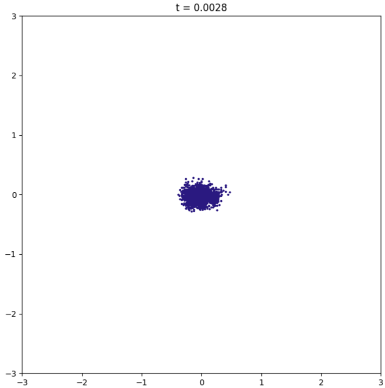
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



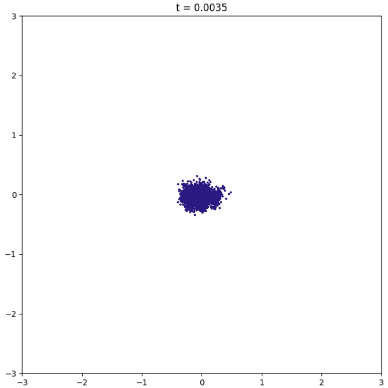
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



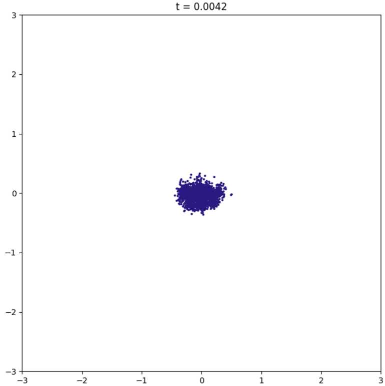
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau} dW_t$$



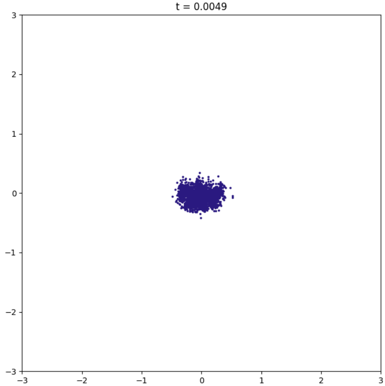
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



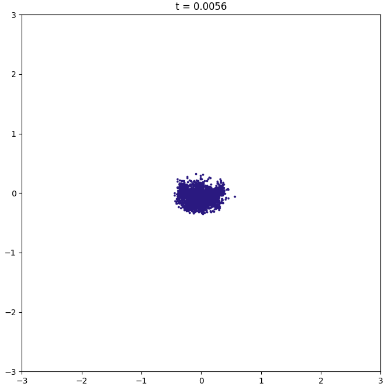
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



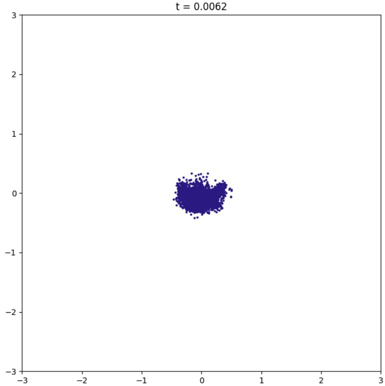
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t) dt + \sqrt{2\tau} dW_t$$



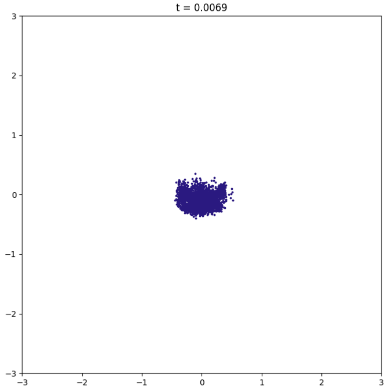
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



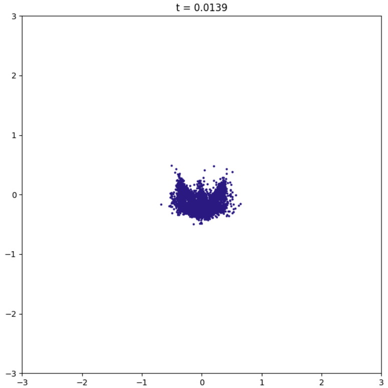
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t) dt + \sqrt{2\tau} dW_t$$



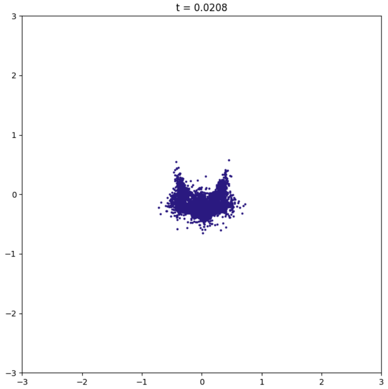
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau} dW_t$$



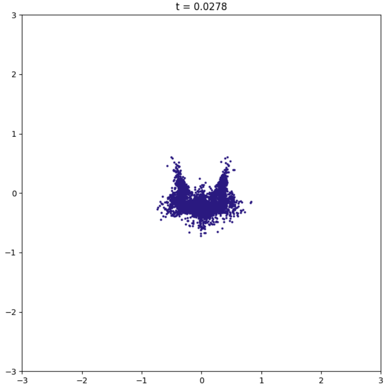
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t) dt + \sqrt{2\tau} dW_t$$



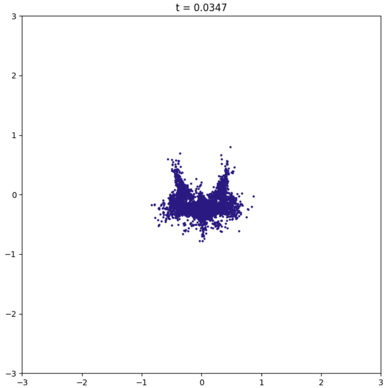
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



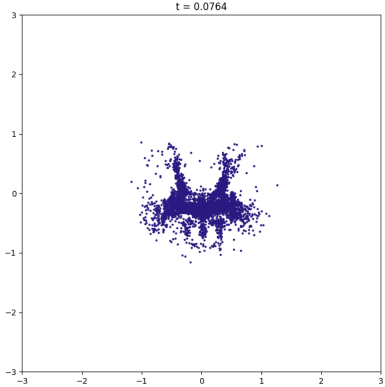
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



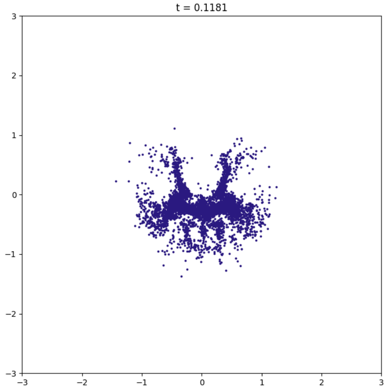
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t) dt + \sqrt{2\tau} dW_t$$



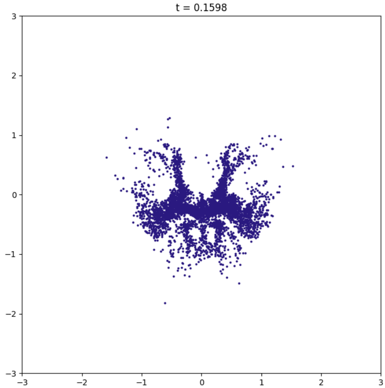
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t) dt + \sqrt{2\tau} dW_t$$



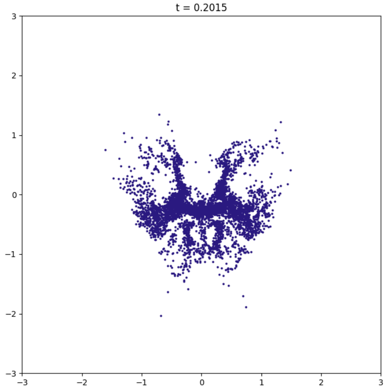
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



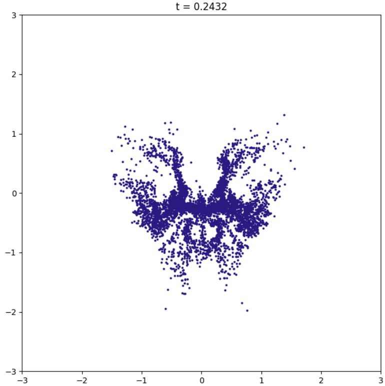
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



Example: Overdamped Langevin Dynamics

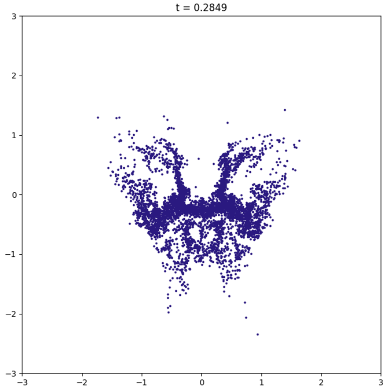
$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



Example: Overdamped Langevin Dynamics

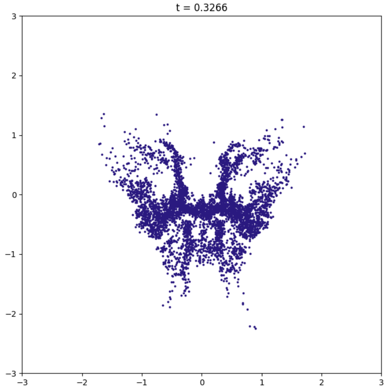
Probabilistic Modeling with Flows

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau} dW_t$$



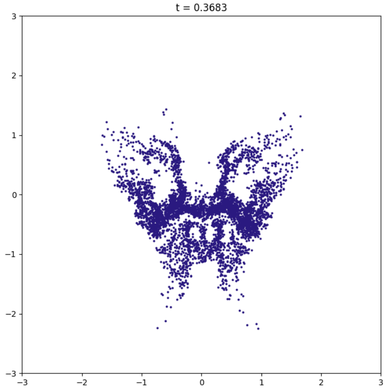
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



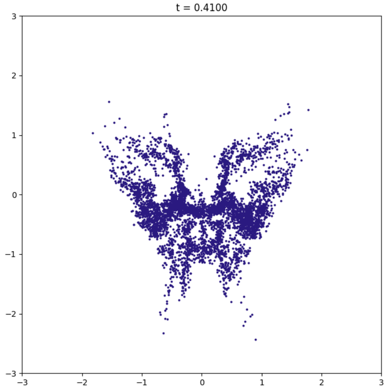
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



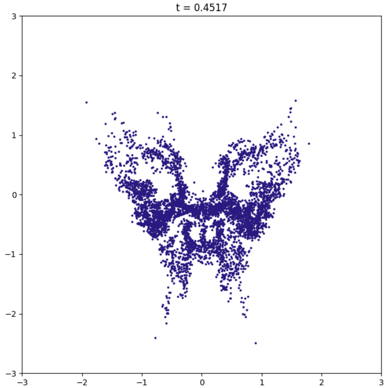
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



Example: Overdamped Langevin Dynamics

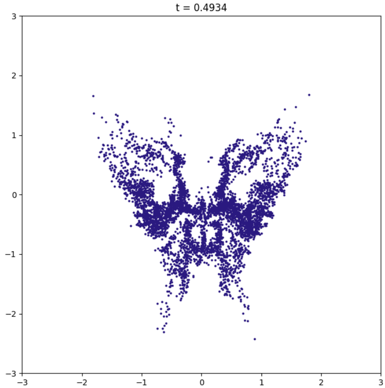
$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



Probabilistic Modeling with Flows

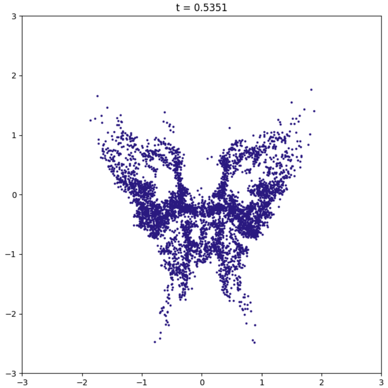
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



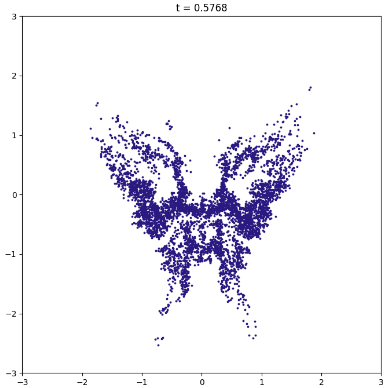
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



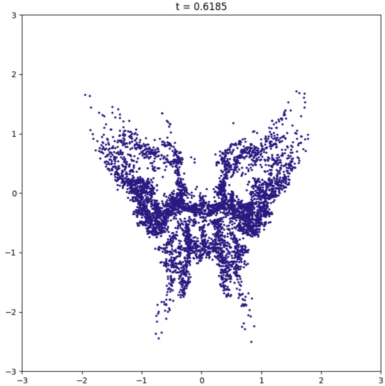
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



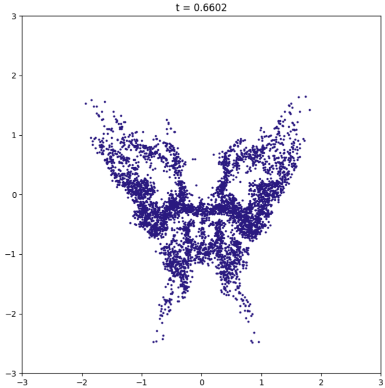
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



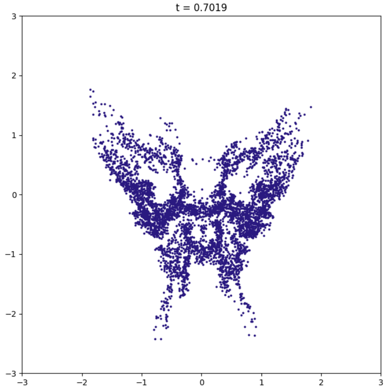
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



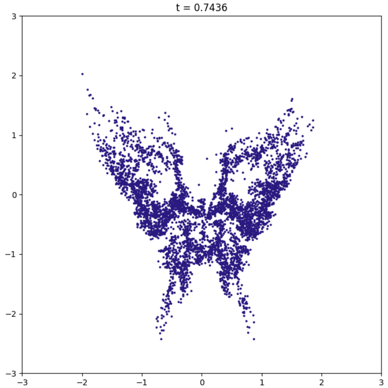
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



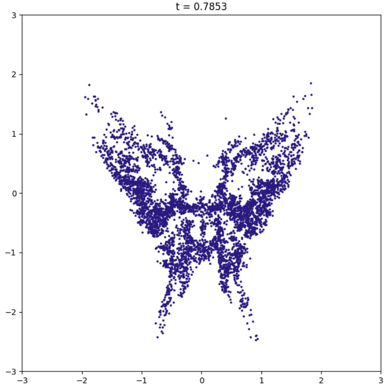
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



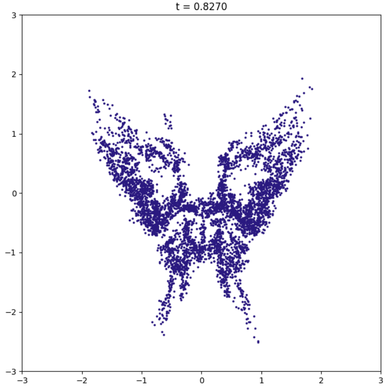
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t) dt + \sqrt{2\tau} dW_t$$



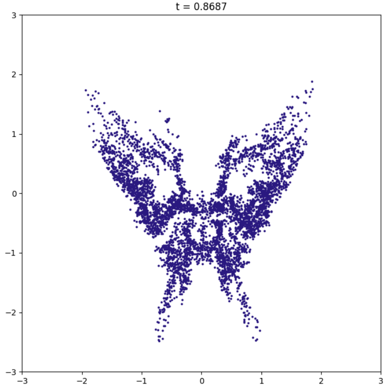
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



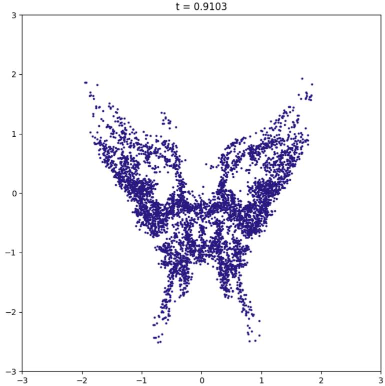
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



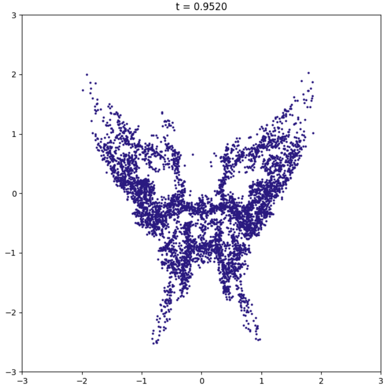
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



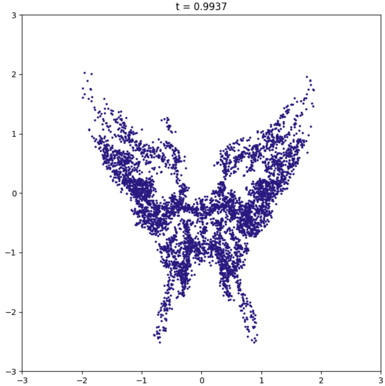
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t) dt + \sqrt{2\tau} dW_t$$



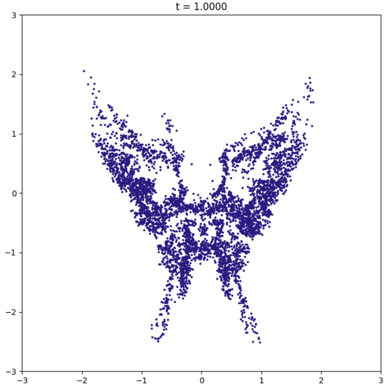
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



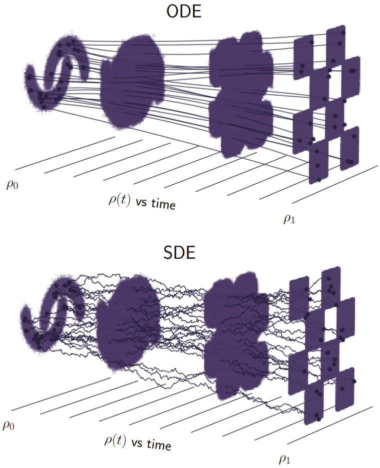
Example: Overdamped Langevin Dynamics

$$dX_t = \tau \nabla_x U(X_t)dt + \sqrt{2\tau}dW_t$$



Is there any way to train a parameterized ODE/SDE!?

Stochastic Interpolants



Stochastic Interpolants: A Unifying Framework for Flows and Diffusions, arXiv:2303.08797 [cs.LG]

Stochastic Interpolants

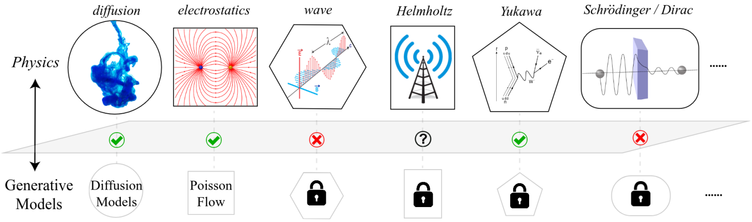
Without latent variable $x_t = (1 - t)x_0 + tx_1$

Stochastic Interpolants: A Unifying Framework for Flows and Diffusions, arXiv:2303.08797 [cs.LG]

Different Approaches to Study Generative Flows

Kolmogorv, Einstein, Feynman, Schrödinger, Smoluchowski, Pontryagin, Fokker, Planck, Monge, Kantorovich, ...

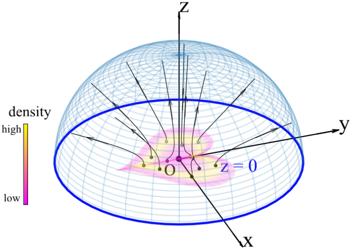
A Physics-Inspired Approach

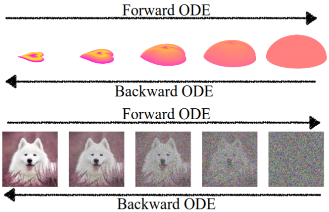


GenPhys: From Physical Processes to Generative Models, arXiv:2304.02637[cs.LG]

Example: Poisson Flow Generative Models

$$dX_t = E(x)\,dt$$





Poisson Flow Generative Models, arXiv:2209.11178[cs.LG]

A Game-Theoretic Approach

Generative modeling is more than designing particle dynamics!

A Game-Theoretic Approach

	Mean-field game (11)			
Model	$\mathcal{M}(\rho)$	$\mathcal{I}(\rho)$	$L(x, v)$	**Dynamics**
Continuous normalizing flow	$\mathcal{D}_{KL}(\rho \Vert \rho_{ref})$	0	0	$dx = v\,dt$
Score-based generative modeling	$-\mathbb{E}_\rho\left[\log \pi\right]$	0	$\frac{\lvert v\rvert^2}{2} - \nabla \cdot f$	$dx = (f + \sigma v)\,dt + \sigma\,dW_t$
Score-based probability flow	$-\frac{1}{2}\mathbb{E}_\rho\left[\log \pi\right]$	$\mathbb{E}_\rho\left[\frac{\lvert\sigma\nabla \log \rho\rvert^2}{8}\right]$	$\frac{\lvert v\rvert^2}{2} - \frac{\nabla \cdot f}{2}$	$dx = (f + \sigma v)\,dt$
Wasserstein gradient flow (WGF) $(\epsilon \to 0)$	$\mathcal{F}(\rho)e^{-T/\epsilon}$	$\frac{e^{-t/\epsilon}}{\epsilon}\mathcal{F}(\rho)$	$\frac{e^{-t/\epsilon}}{2}\lvert v\rvert^2$	$dx = v\,dt$
OT-Flow	$\mathcal{D}_{KL}(\rho \Vert \rho_{ref})$	0	$\frac{1}{2}\lvert v\rvert^2$	$dx = v\,dt$
Boltzmann generator	$\lambda\mathcal{D}_{KL}(\pi \Vert \rho) + (1-\lambda)\mathcal{D}_{KL}(\rho \Vert \pi)$	0	0	$dx = v\,dt$
Schrödinger bridge	$\rho = \pi$	0	$\frac{1}{2}\lvert v\rvert^2$	$dx = \sigma v\,dt + \sigma\,dW_t$
Generalized Schrödinger bridge	$\rho = \pi$	$\mathcal{I}(x, \rho)$	$\frac{1}{2}\lvert v\rvert^2$	$dx = \sigma v\,dt + \sigma\,dW_t$

A mean-field games laboratory for generative modeling, arXiv:2304.13534 [stat.ML]

References I

Benjamin J Zhang and Markos A Katsoulakis.
A mean-field games laboratory for generative modeling.
arXiv preprint arXiv:2304.13534, 2023.

Felix Draxler, Stefan Wahl, Christoph Schnörr, and Ullrich Köthe.
On the universality of coupling-based normalizing flows.
arXiv preprint arXiv:2402.06578, 2024.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden.
Stochastic interpolants: A unifying framework for flows and diffusions, 2023.

References II

- Ziming Liu, Di Luo, Yilun Xu, Tommi Jaakkola, and Max Tegmark.
  Genphys: From physical processes to generative models.
  arXiv preprint arXiv:2304.02637, 2023.

- Xingchao Liu, Chengyue Gong, and Qiang Liu.
  Flow straight and fast: Learning to generate and transfer data with rectified flow.
  arXiv preprint arXiv:2209.03003, 2022.

- Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola.
  Poisson flow generative models.
  Advances in Neural Information Processing Systems, 35:16782–16795, 2022.

References III

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

Michele Pavon, Giulio Trigila, and Esteban G Tabak. The data-driven schrödinger bridge. *Communications on Pure and Applied Mathematics*, 74(7):1545–1573, 2021.

References IV

Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker.
Normalizing flows: An introduction and review of current methods.
IEEE transactions on pattern analysis and machine intelligence, 43(11):3964–3979, 2020.

Jonas Köhler, Leon Klein, and Frank Noé.
Equivariant flows: exact likelihood generative learning for symmetric densities.
In *International conference on machine learning*, pages 5361–5370. PMLR, 2020.

References V

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud.
Neural ordinary differential equations.
Advances in neural information processing systems, 31, 2018.

Danilo Rezende and Shakir Mohamed.
Variational inference with normalizing flows.
In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.