Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein[1], Eric Weiss[2],
Niru Maheswaranathan[1], Surya Ganguli[1]

[1] Stanford University, [2] University of California at Berkeley

Outline

- Motivation: The promise of deep unsupervised learning
- Physical intuition: Diffusion processes and time reversal
- Diffusion probabilistic model: Derivation and experimental results

Outline

- **Motivation: The promise of deep unsupervised learning**
- Physical intuition: Diffusion processes and time reversal
- Diffusion probabilistic model: Derivation and experimental results

The Promise of Deep Unsupervised Learning

The Promise of Deep Unsupervised Learning

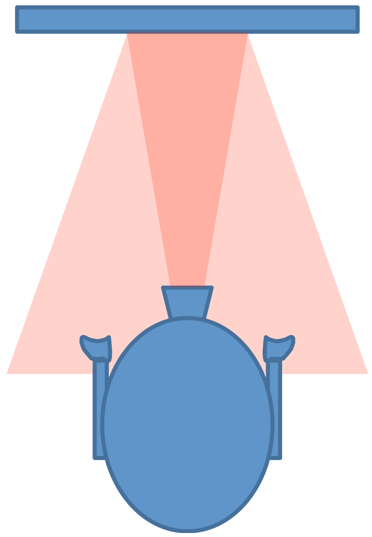
- Unknown features/labels

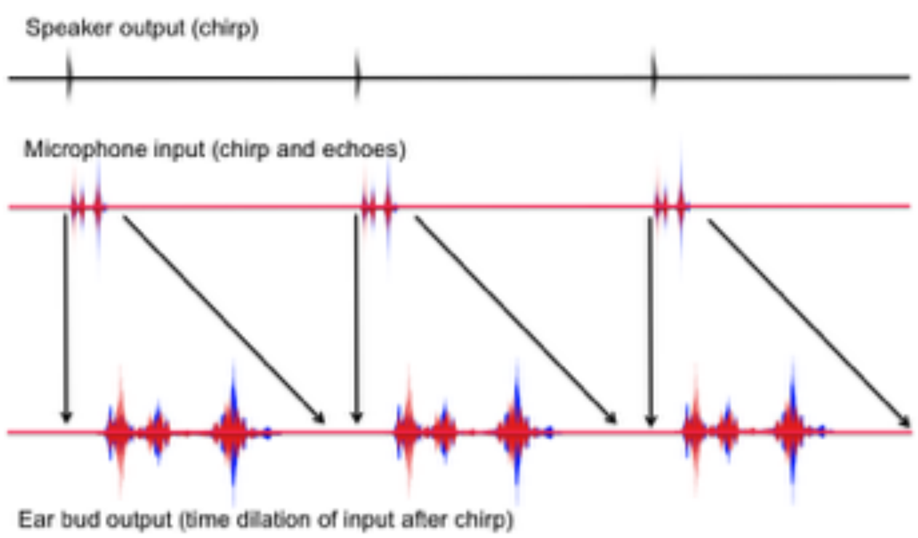
The Promise of Deep Unsupervised Learning

- Unknown features/labels
  - Novel modalities

The Promise of Deep Unsupervised Learning

- Unknown features/labels
    - Novel modalities





[Trans Biomed Eng, 2015]

The Promise of Deep Unsupervised Learning

- Unknown features/labels
  - Novel modalities

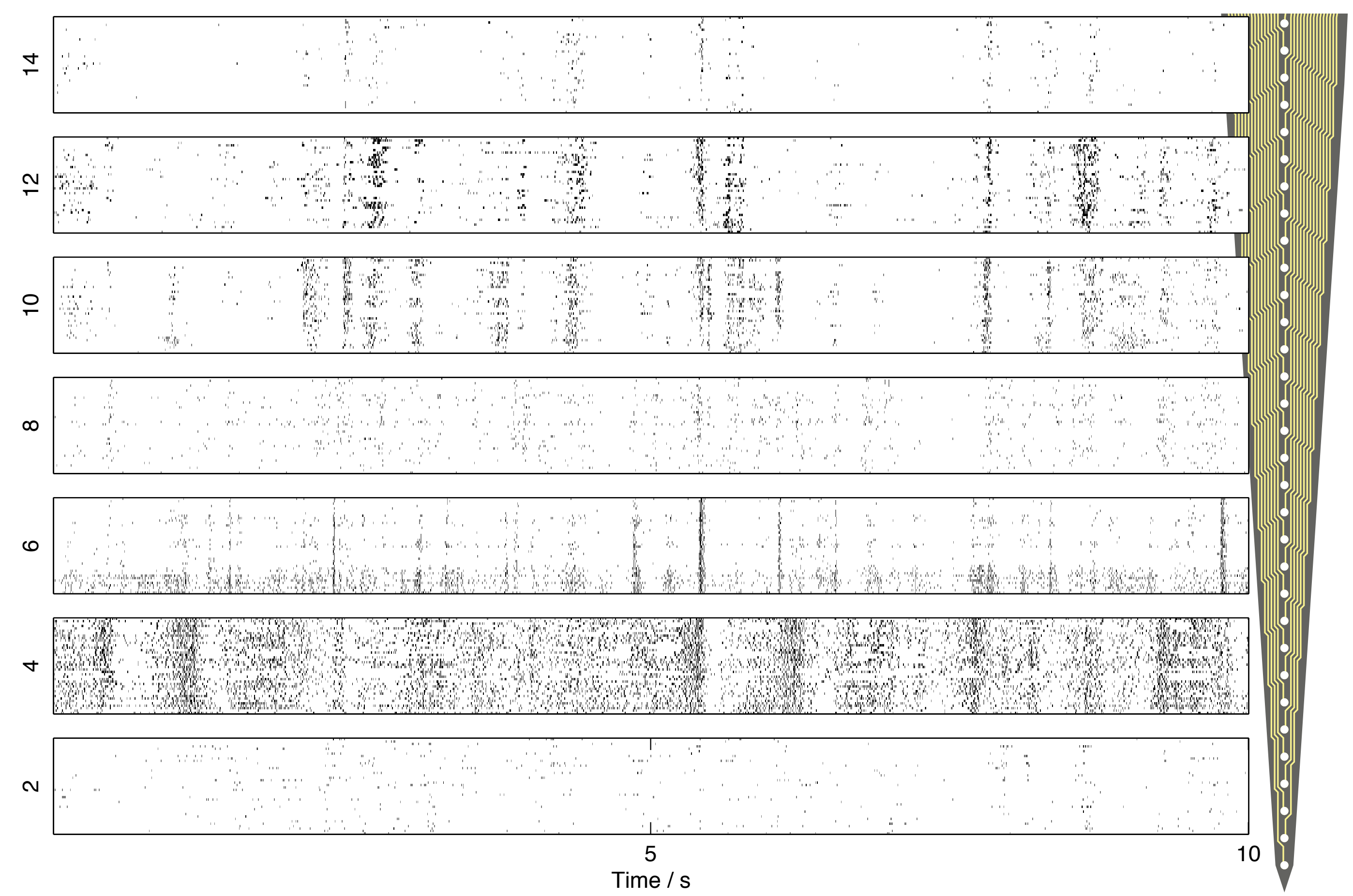
The Promise of Deep Unsupervised Learning

- Unknown features/labels
  - Novel modalities
  - Exploratory data analysis

The Promise of Deep Unsupervised Learning

- Unknown features/labels
  - Novel modalities
  - Exploratory data analysis

7 exemplar multiunits responding to 40 repeated trials of natural video in cat V1



[PLoS Comp Bio 2014] [Neuron 2013]

The Promise of Deep Unsupervised Learning

- Unknown features/labels
  - Novel modalities
  - Exploratory data analysis

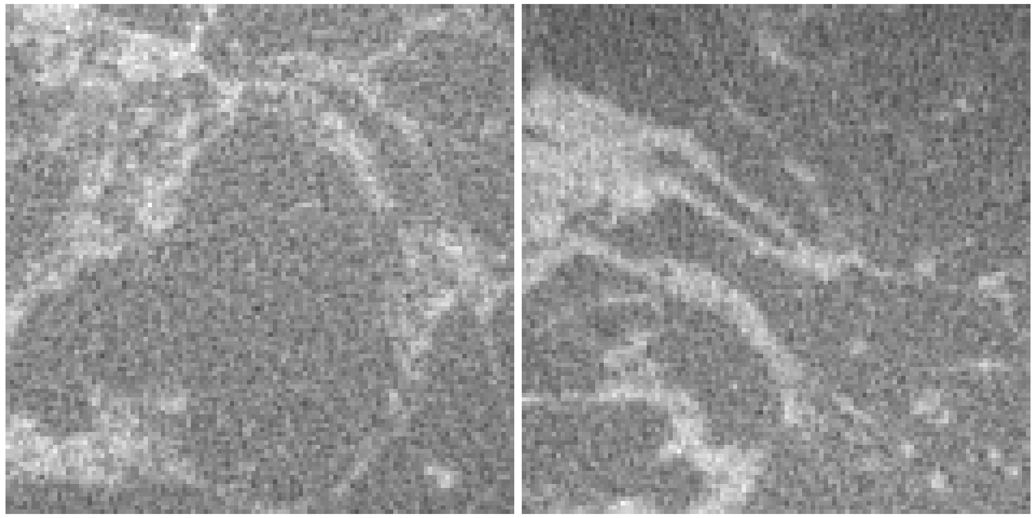
The Promise of Deep Unsupervised Learning

- Unknown features/labels
    - Novel modalities
    - Exploratory data analysis
- Expensive labels

The Promise of Deep Unsupervised Learning

- Unknown features/labels
  - Novel modalities
  - Exploratory data analysis
- Expensive labels

Coronal breast CT



[SPIE 2009] [Med Phys 2014]

The Promise of Deep Unsupervised Learning

- Unknown features/labels
  - Novel modalities
  - Exploratory data analysis
- Expensive labels

The Promise of Deep Unsupervised Learning

- Unknown features/labels
    - Novel modalities
    - Exploratory data analysis
- Expensive labels
- Unpredictable tasks / one shot learning

Outline

- Motivation: The promise of deep unsupervised learning
- **Physical intuition: Diffusion processes and time reversal**
- Diffusion probabilistic model: Derivation and experimental results

Outline

- Motivation: The promise of deep unsupervised learning
- **Physical intuition: Diffusion processes and time reversal**
  - Destroy structure in data
  - Carefully characterize the destruction
  - Learn how to **reverse time**
- Diffusion probabilistic model: Derivation and experimental results

Observation 1: Diffusion Destroys Structure

- Dye density represents probability density

Observation 1: Diffusion Destroys Structure

- Dye density represents probability density
- Goal: Learn structure of probability density

Observation 1: Diffusion Destroys Structure

- Dye density represents probability density
- Goal: Learn structure of probability density
- Observation: Diffusion destroys structure

Observation 1: Diffusion Destroys Structure

- Dye density represents probability density
- Goal: Learn structure of probability density
- Observation: Diffusion destroys structure

Observation 1: Diffusion Destroys Structure

- Dye density represents probability density
- Goal: Learn structure of probability density
- Observation: Diffusion destroys structure

Data distribution → Uniform distribution

Core Idea: Recover Structure by Reversing Time

- What if we could reverse time?

Core Idea: Recover Structure by Reversing Time

- What if we could reverse time?

Core Idea: Recover Structure by Reversing Time

- What if we could reverse time?

Data distribution ← Uniform distribution

Core Idea: Recover Structure by Reversing Time

- What if we could reverse time?
- Recover data distribution by starting from uniform distribution and running dynamics backwards

Data distribution ← Uniform distribution

Core Idea: Recover Structure by Reversing Time

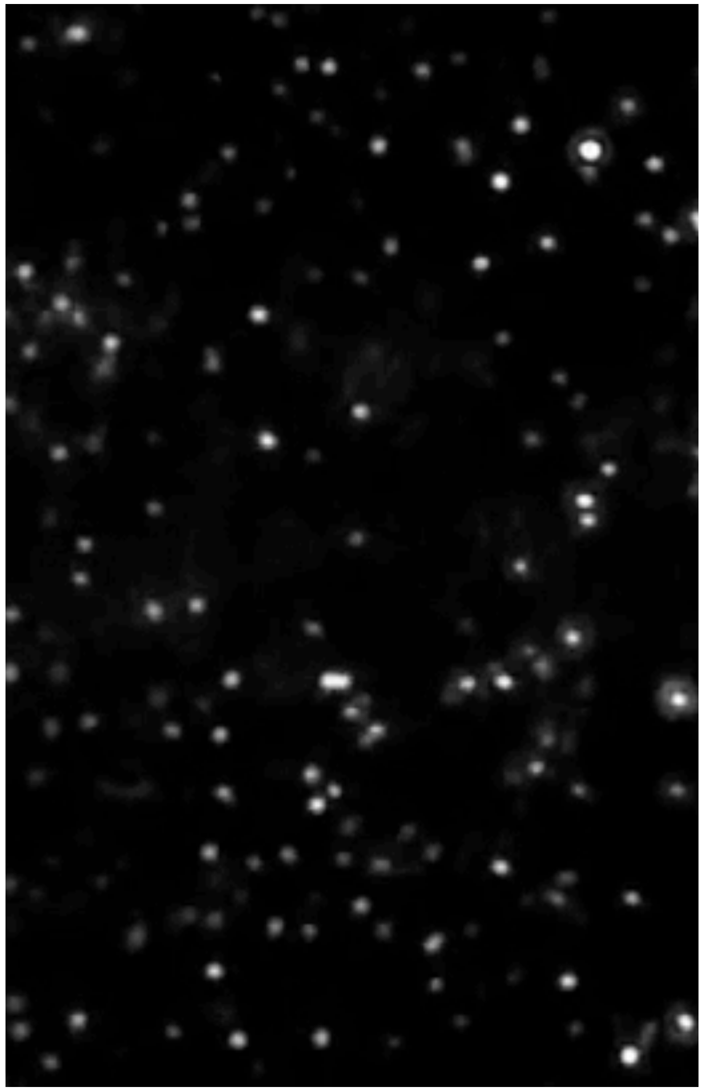
- What if we could reverse time?
- Recover data distribution by starting from uniform distribution and running dynamics backwards

Data distribution ← Uniform distribution

Core Idea: Recover Structure by Reversing Time

Core Idea: Recover Structure by Reversing Time

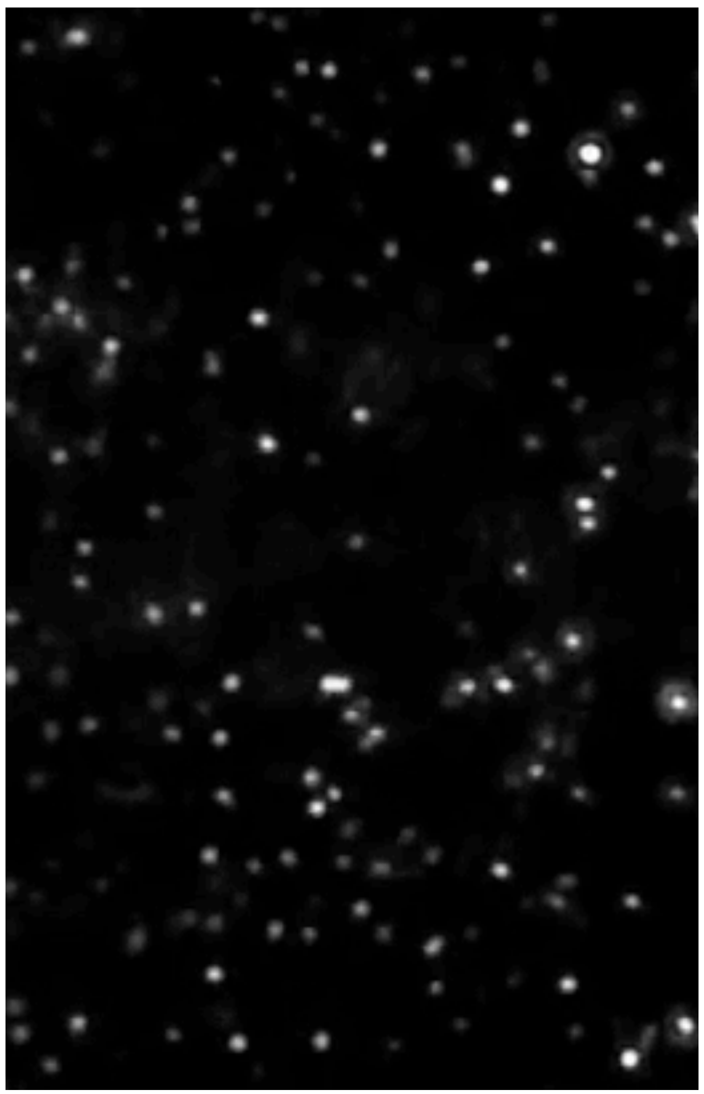
Observation 2: Microscopic Diffusion is Time Reversible



© Rutger Saly

- Microscopic view
- Brownian motion

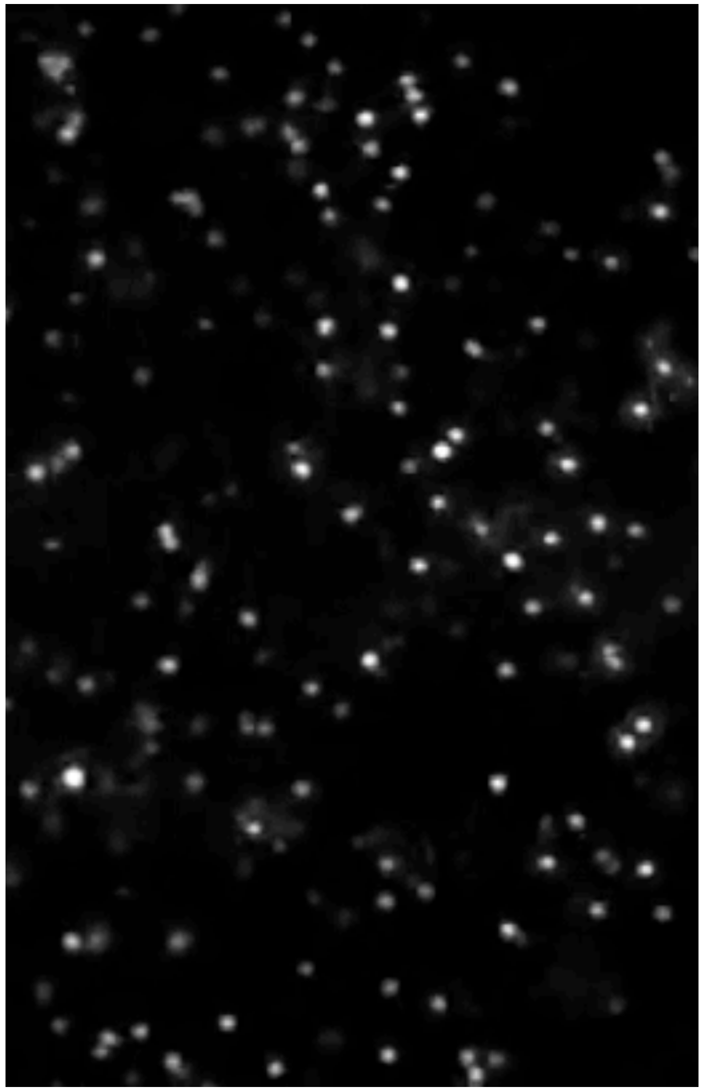
Observation 2: Microscopic Diffusion is Time Reversible



© Rutger Saly

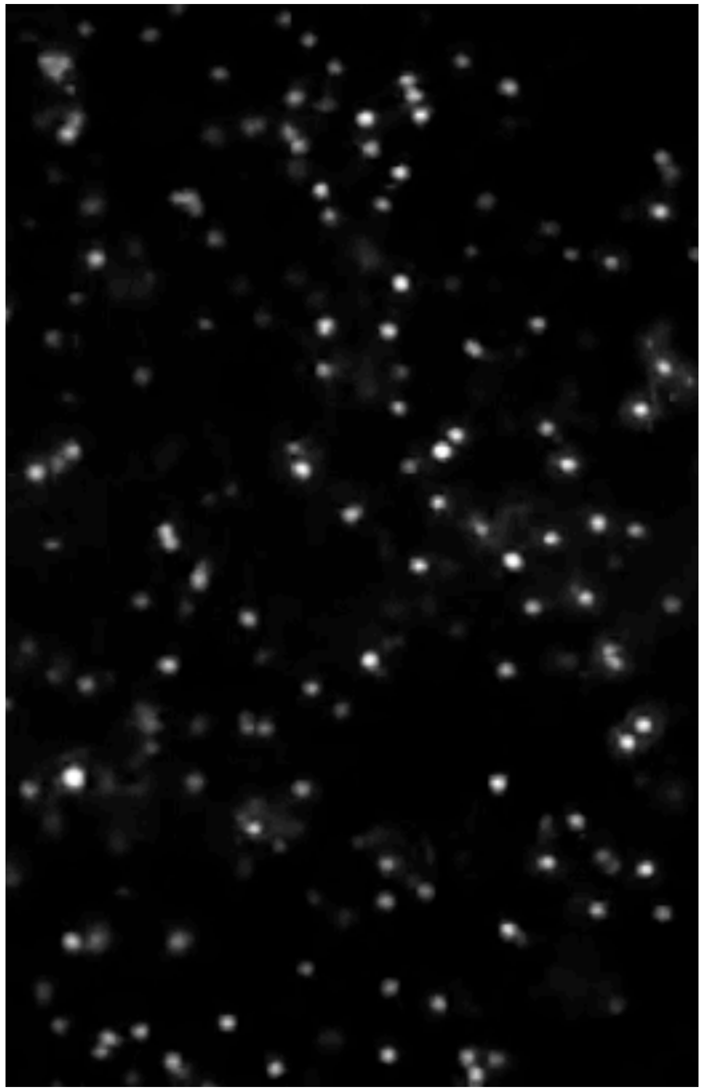
- Microscopic view
- Brownian motion

Observation 2: Microscopic Diffusion is Time Reversible



- Microscopic view
- Brownian motion

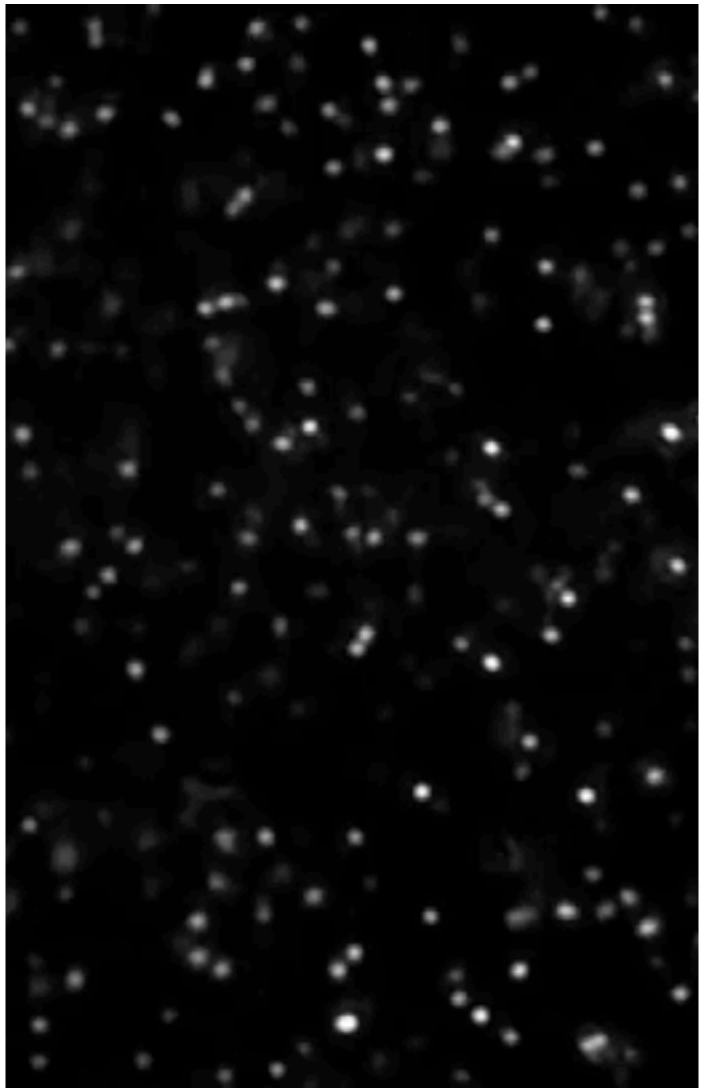
Observation 2: Microscopic Diffusion is Time Reversible



© Rutger Saly

- Microscopic view
- Brownian motion

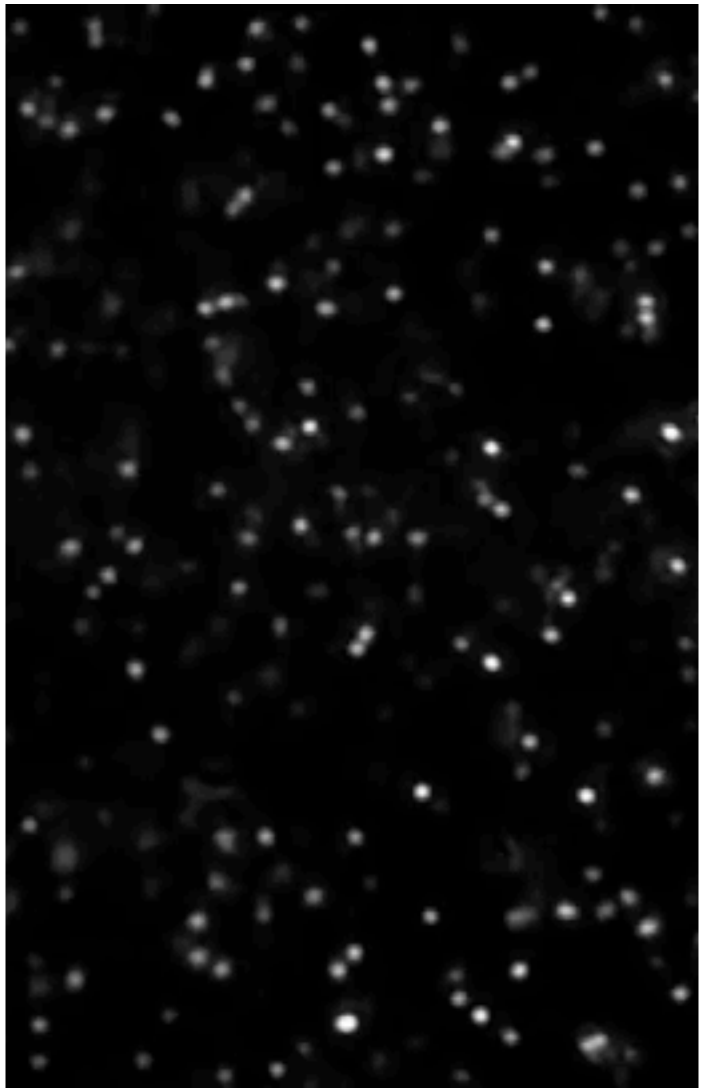
Observation 2: Microscopic Diffusion is Time Reversible



© Rutger Saly

- Microscopic view
- Brownian motion
- Position updates are small Gaussians

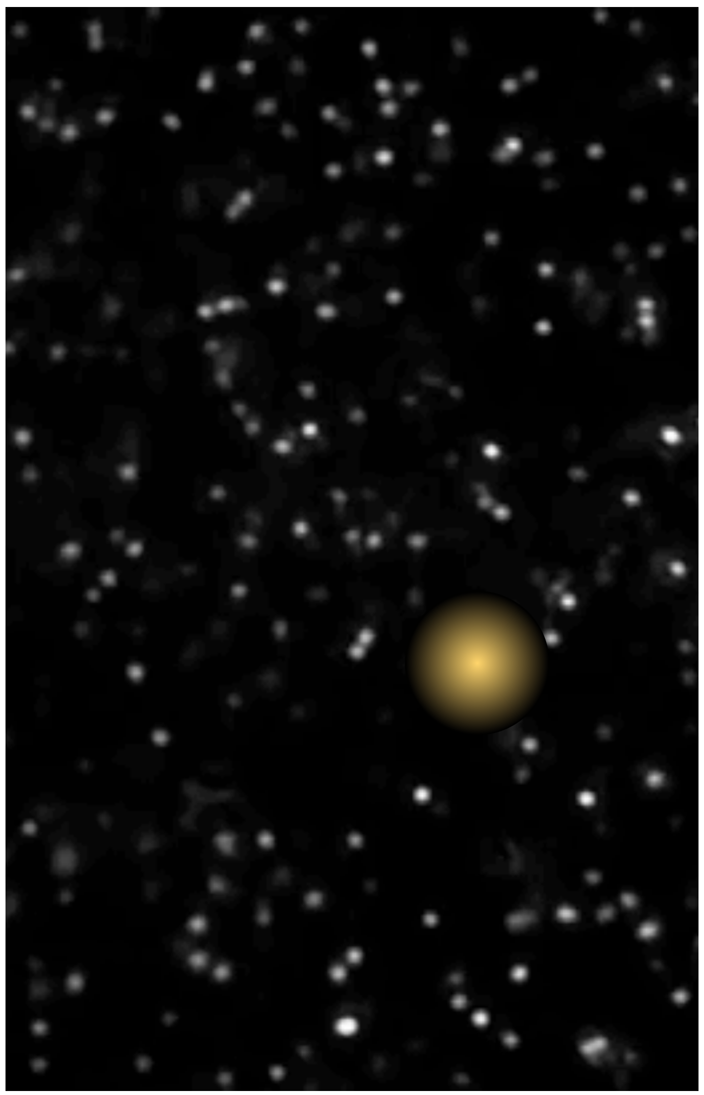
Observation 2: Microscopic Diffusion is Time Reversible



© Rutger Saly

- Microscopic view
- Brownian motion
- Position updates are small Gaussians

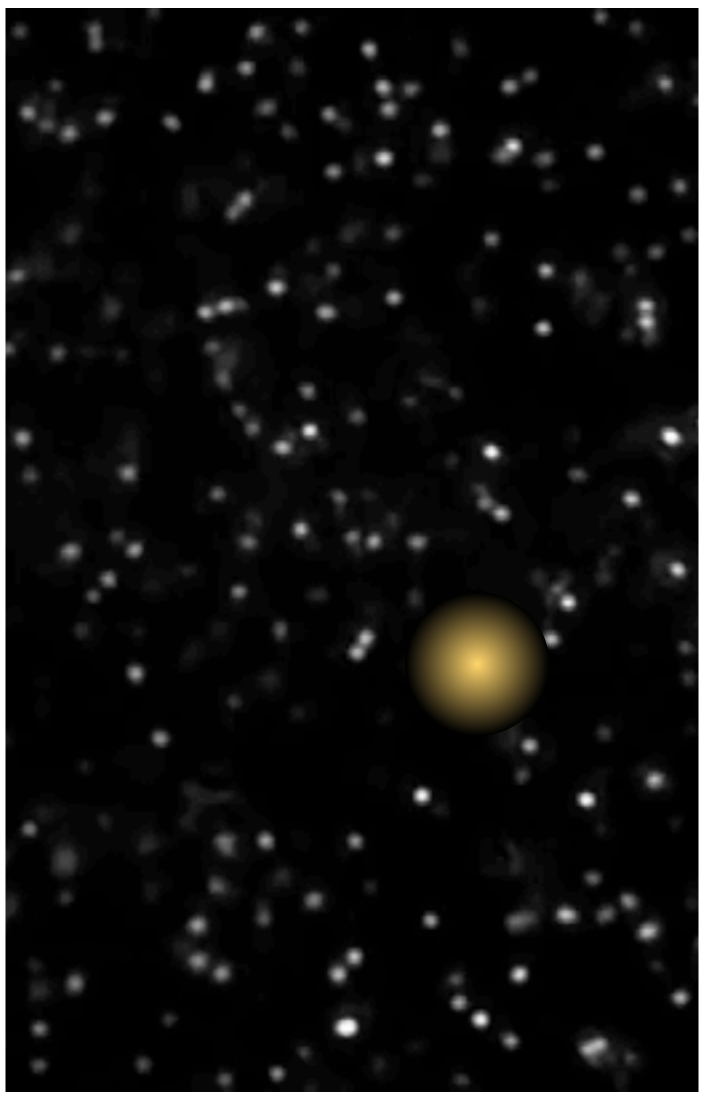
Observation 2: Microscopic Diffusion is Time Reversible



© Rutger Saly

- Microscopic view
- Brownian motion
- Position updates are small Gaussians

Observation 2: Microscopic Diffusion is Time Reversible



© Rutger Saly

- Microscopic view
- Brownian motion
- Position updates are small Gaussians
  - Both forwards and backwards in time

Overview of Diffusion-based Probabilistic Models

Overview of Diffusion-based Probabilistic Models

- Destroy all structure in data distribution using diffusion process

Overview of Diffusion-based Probabilistic Models

- Destroy all structure in data distribution using diffusion process
- Learn reversal of diffusion process
  - Estimate function for mean and covariance of each step in the reverse diffusion process (binomial rate for binary data)

Overview of Diffusion-based Probabilistic Models

- Destroy all structure in data distribution using diffusion process
- Learn reversal of diffusion process
  - Estimate function for mean and covariance of each step in the reverse diffusion process (binomial rate for binary data)
- Reverse diffusion process is the model of the data

Outline

- Motivation: The promise of deep unsupervised learning
- Physical intuition: Diffusion processes and time reversal
- **Diffusion probabilistic model: Derivation and experimental results**
  - Algorithm
  - Deep convolutional network: Universal function approximator
  - Multiplying distributions: Inputation, denoising, computing posteriors

Outline

- Motivation: The promise of deep unsupervised learning
- Physical intuition: Diffusion processes and time reversal
- Diffusion probabilistic model: Derivation and experimental results
    - **Algorithm**
    - Deep convolutional network: Universal function approximator
    - Multiplying distributions: Inputation, denoising, computing posteriors

Destroy All Structure in Data using Diffusion Process

Data
distribution

$$q\left(\mathbf{x}^{(0)}\right)$$

Destroy All Structure in Data using Diffusion Process

Data distribution

Forward diffusion

$$q\left(\mathbf{x}^{(0)}\right)$$

$$q\left(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}\right) = \mathcal{N}\left(\mathbf{x}^{(t)}; \mathbf{x}^{(t-1)}\sqrt{1-\beta_t}, \mathbf{I}\beta_t\right)$$

Destroy All Structure in Data using Diffusion Process

Data distribution

Forward diffusion

$$q\left(\mathbf{x}^{(0)}\right)$$

$$q\left(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}\right) = \mathcal{N}\left(\mathbf{x}^{(t)}; \underline{\mathbf{x}^{(t-1)}\sqrt{1-\beta_t}}, \mathbf{I}\beta_t\right)$$

Decay towards origin

Destroy All Structure in Data using Diffusion Process

Data distribution

Forward diffusion

$$q\left(\mathbf{x}^{(0)}\right)$$

$$q\left(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}\right) = \mathcal{N}\left(\mathbf{x}^{(t)}; \underbrace{\mathbf{x}^{(t-1)}\sqrt{1-\beta_t}}_{\text{Decay towards origin}}, \underbrace{\mathbf{I}\beta_t}_{\text{Add small noise}}\right)$$

Destroy All Structure in Data using Diffusion Process

Data distribution

Forward diffusion

Noise distribution

$$q\left(\mathbf{x}^{(0)}\right)$$

$$q\left(\mathbf{x}^{(T)}\right) \approx \mathcal{N}\left(\mathbf{x}^{(T)}; 0, \mathbf{I}\right)$$

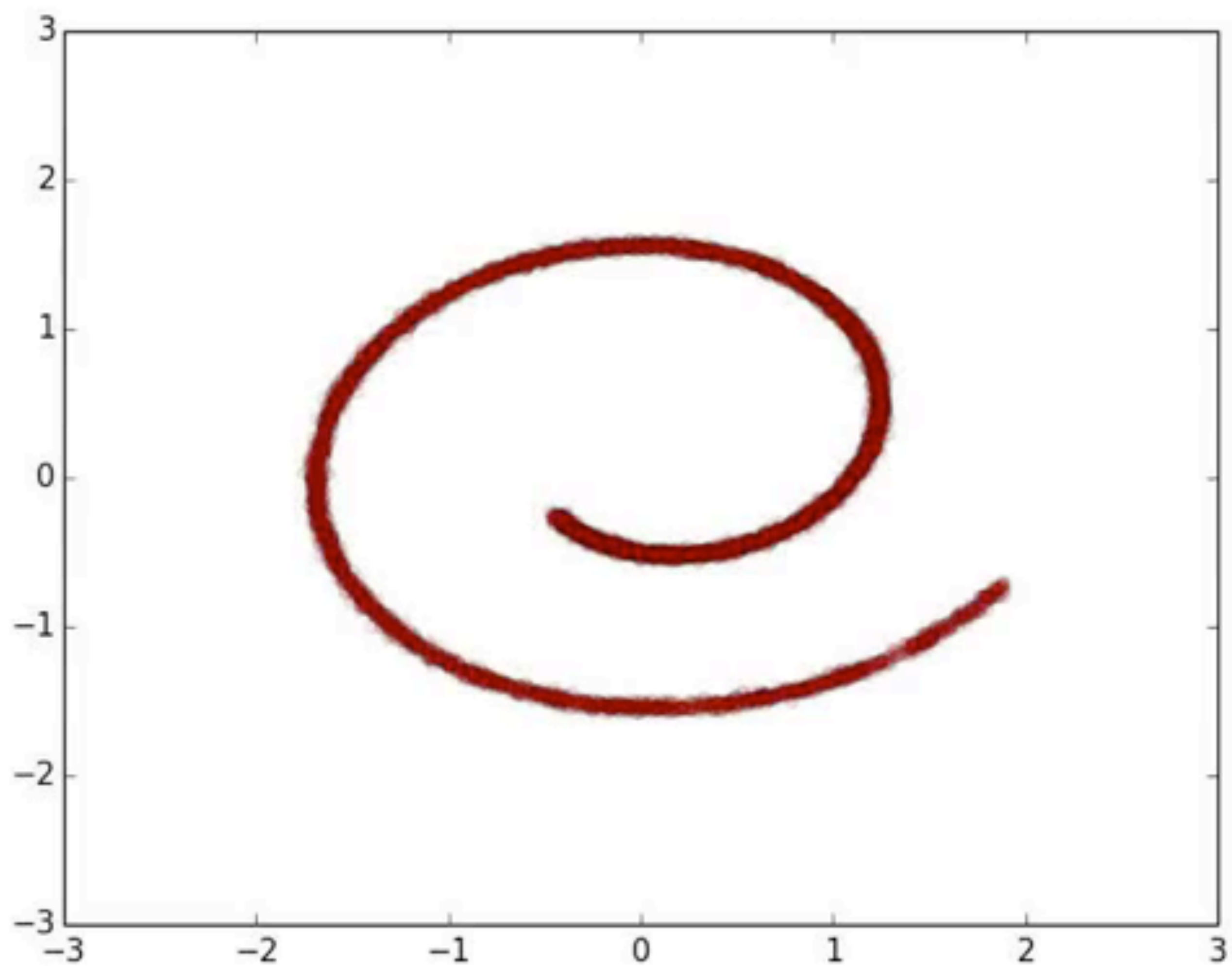
$$q\left(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}\right) = \mathcal{N}\left(\mathbf{x}^{(t)}; \underline{\mathbf{x}^{(t-1)}\sqrt{1-\beta_t}}, \underline{\mathbf{I}\beta_t}\right)$$

Decay towards origin

Add small noise

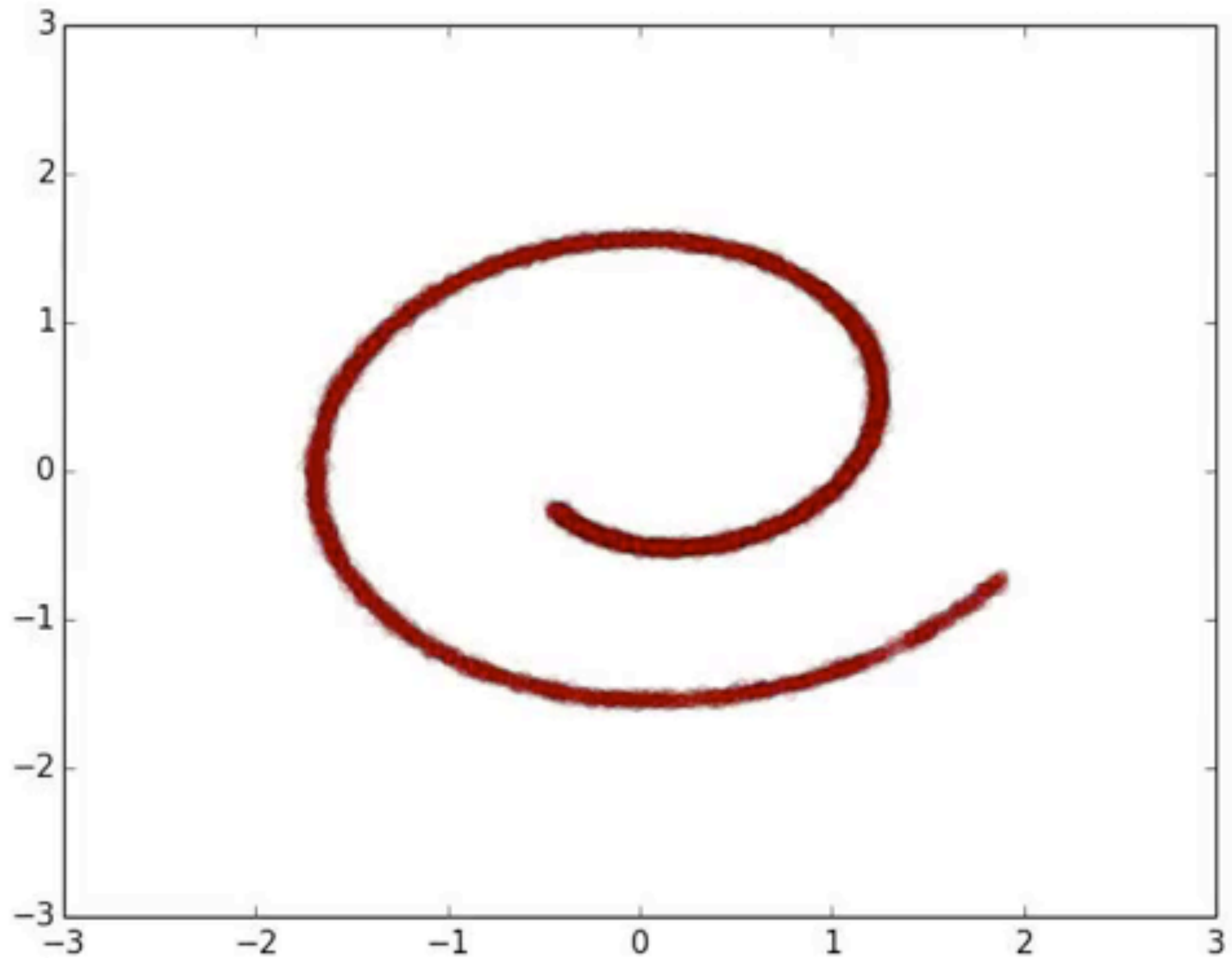
Forward Diffusion Process on Swiss Roll

- Start at data
- Run Gaussian diffusion until samples become Gaussian blob



Forward Diffusion Process on Swiss Roll

- Start at data
- Run Gaussian diffusion until samples become Gaussian blob



Recover Structure in Data using Reversal of Diffusion Process

Noise distribution

$$p\left(\mathbf{x}^{(T)}\right) = \mathcal{N}\left(\mathbf{x}^{(T)}; 0, \mathbf{I}\right)$$

Recover Structure in Data using Reversal of Diffusion Process

Reverse diffusion

Noise distribution

$$p\left(\mathbf{x}^{(T)}\right) = \mathcal{N}\left(\mathbf{x}^{(T)}; 0, \mathbf{I}\right)$$

$$p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}^{(t-1)}; f_{\mu}\left(\mathbf{x}^{(t)}, t\right), f_{\Sigma}\left(\mathbf{x}^{(t)}, t\right)\right)$$

Recover Structure in Data using Reversal of Diffusion Process

Reverse diffusion

Noise distribution

$$p\left(\mathbf{x}^{(T)}\right) = \mathcal{N}\left(\mathbf{x}^{(T)}; 0, \mathbf{I}\right)$$

$$p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}^{(t-1)}; \underbrace{f_\mu\left(\mathbf{x}^{(t)}, t\right), f_\Sigma\left(\mathbf{x}^{(t)}, t\right)}\right)$$

Learned drift and covariance functions

Recover Structure in Data using Reversal of Diffusion Process

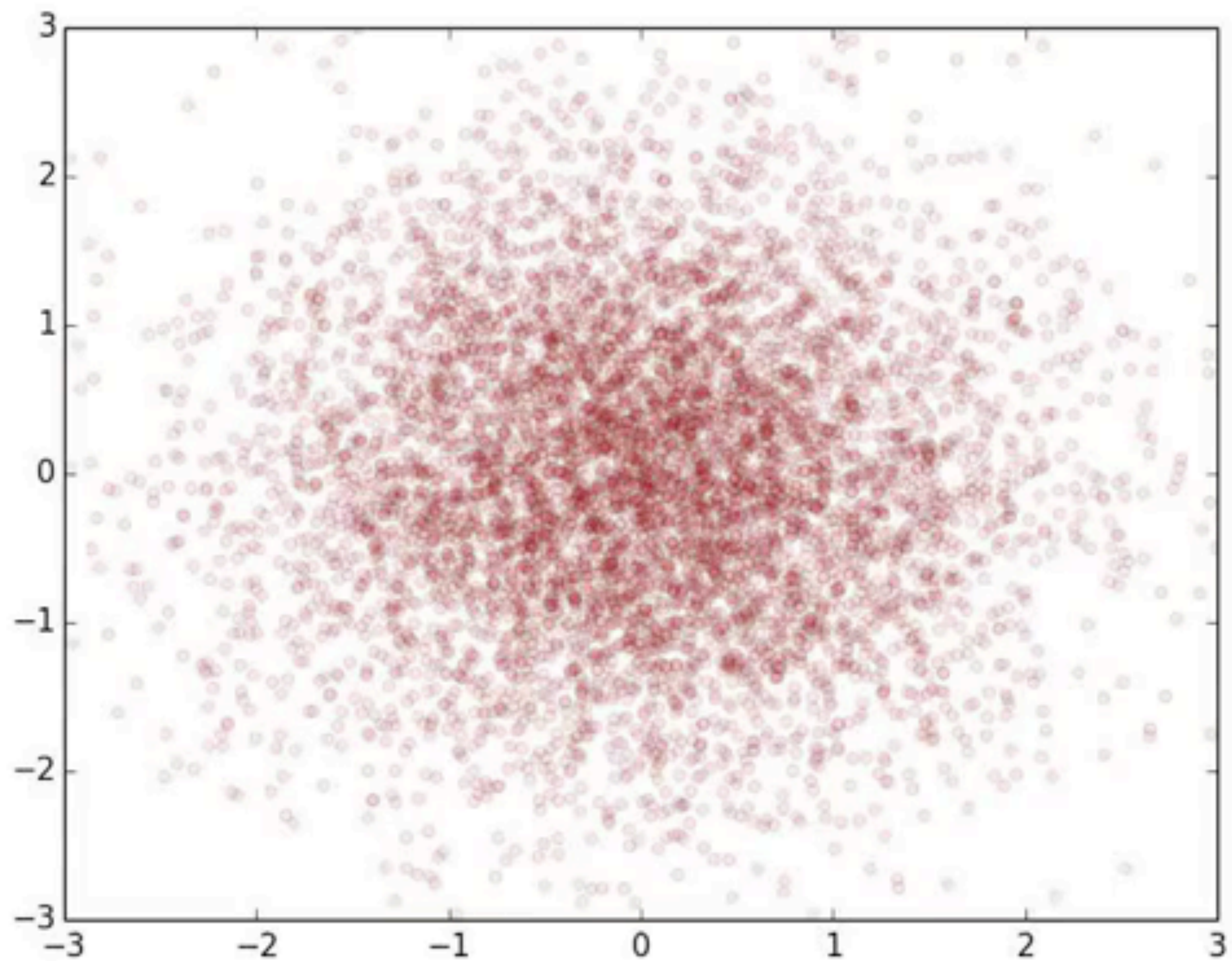
Data distribution	Reverse diffusion	Noise distribution
$p\left(\mathbf{x}^{(0)}\right) \approx q\left(\mathbf{x}^{(0)}\right)$	$\longleftarrow$	$p\left(\mathbf{x}^{(T)}\right) = \mathcal{N}\left(\mathbf{x}^{(T)}; 0, \mathbf{I}\right)$

$$p\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}^{(t-1)}; \underline{f_\mu\left(\mathbf{x}^{(t)}, t\right), f_\Sigma\left(\mathbf{x}^{(t)}, t\right)}\right)$$

Learned drift and covariance functions

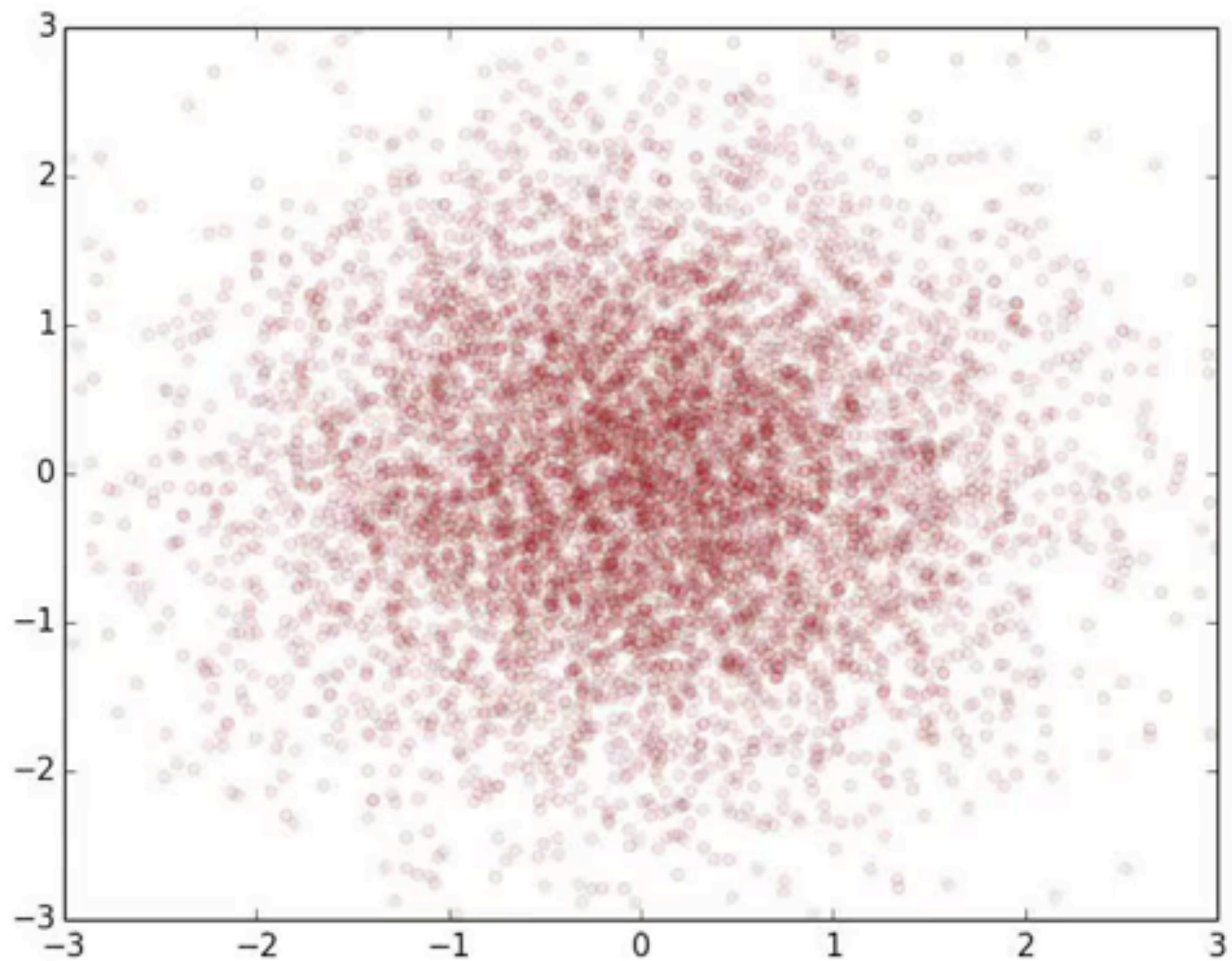
Learned Reverse Diffusion Process on Swiss Roll

- Start at Gaussian blob
- Run Gaussian diffusion until samples become data distribution

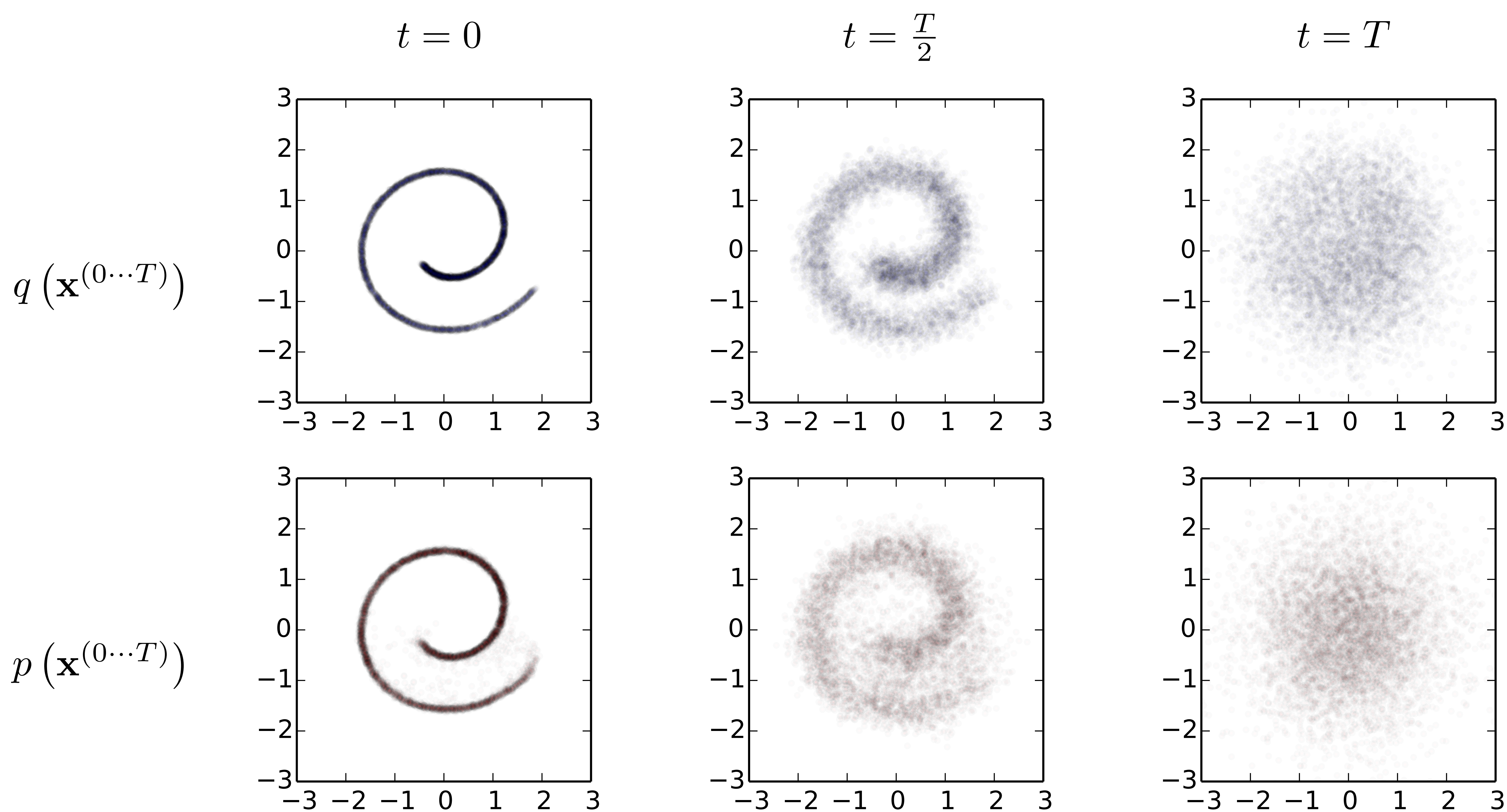


Learned Reverse Diffusion Process on Swiss Roll

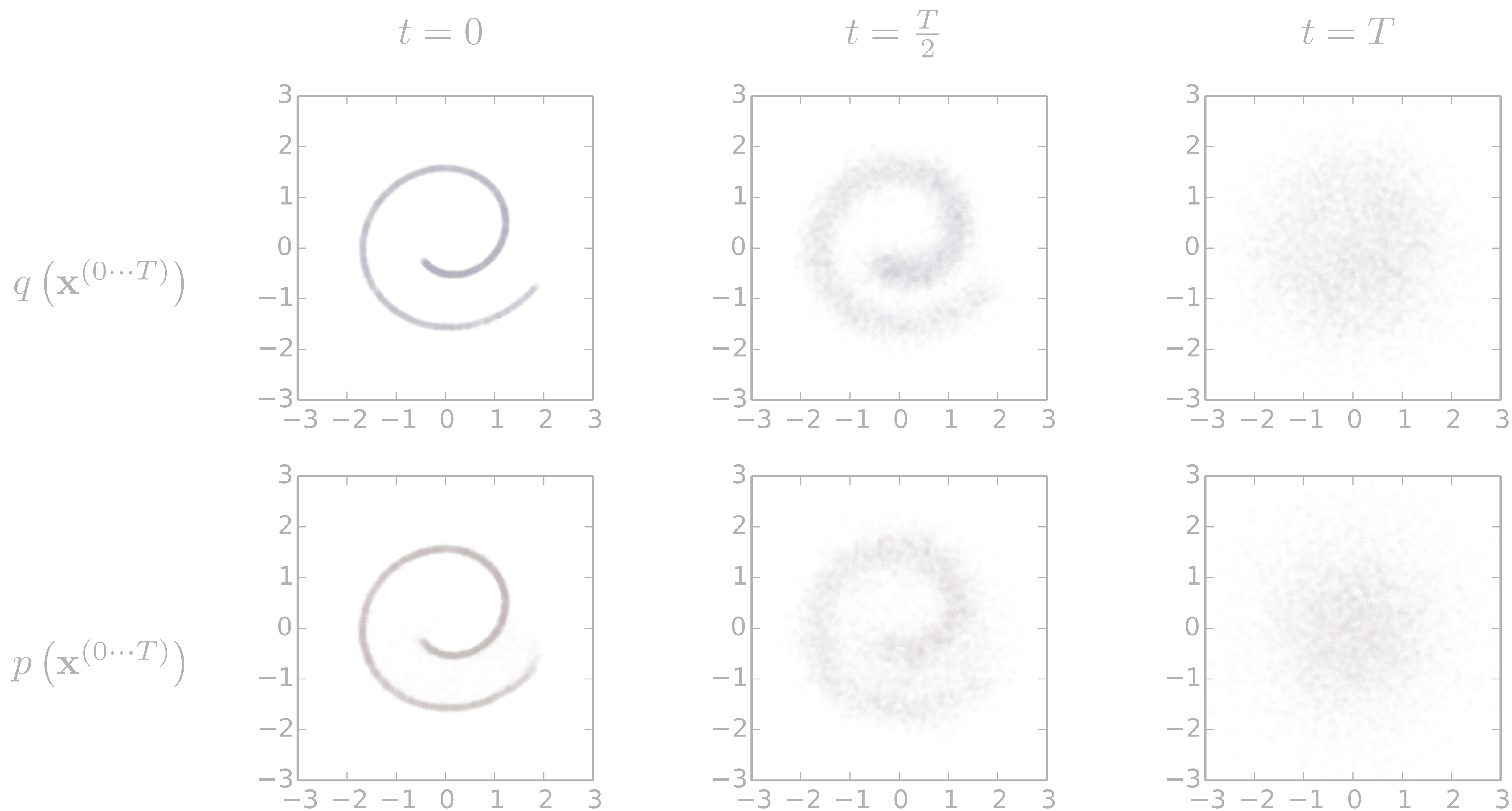
- Start at Gaussian blob
- Run Gaussian diffusion until samples become data distribution



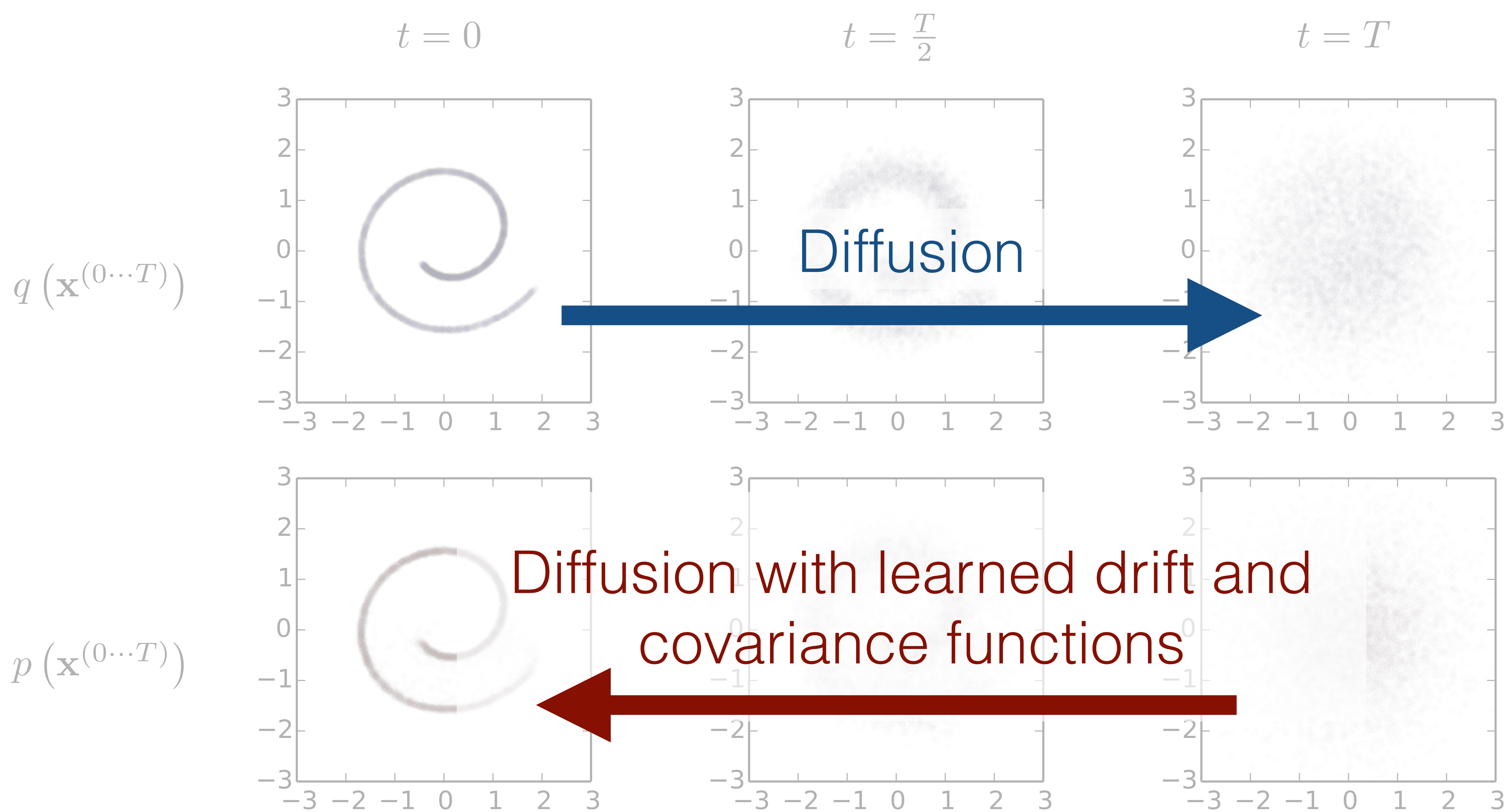
Summary of Forward and Reverse Diffusion on Swiss Roll



Summary of Forward and Reverse Diffusion on Swiss Roll



Summary of Forward and Reverse Diffusion on Swiss Roll



Training the Reverse Diffusion Process

Model probability

$$p\left(\mathbf{x}^{(0)}\right) = \int d\mathbf{x}^{(1\cdots T)} p\left(\mathbf{x}^{(0\cdots T)}\right)$$

Training the Reverse Diffusion Process

Model probability

$$p\left(\mathbf{x}^{(0)}\right) = \int d\mathbf{x}^{(1\cdots T)} p\left(\mathbf{x}^{(0\cdots T)}\right)$$

Annealed importance sampling / Jarzynski equality

$$p\left(\mathbf{x}^{(0)}\right) = \int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)}\right) \frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)}\right)}$$

Training the Reverse Diffusion Process

Model probability

$$p\left(\mathbf{x}^{(0)}\right) = \int d\mathbf{x}^{(1\cdots T)} p\left(\mathbf{x}^{(0\cdots T)}\right)$$

Annealed importance sampling / Jarzynski equality

$$p\left(\mathbf{x}^{(0)}\right) = \int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}\right) \frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}\right)}$$

Log Likelihood

$$L = \int d\mathbf{x}^{(0)} q\left(\mathbf{x}^{(0)}\right) \log\left[\int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)}\right) \frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)}\right)}\right]$$

Model probability

$$p\left(\mathbf{x}^{(0)}\right) = \int d\mathbf{x}^{(1\cdots T)} p\left(\mathbf{x}^{(0\cdots T)}\right)$$

Annealed importance sampling / Jarzynski equality

$$p\left(\mathbf{x}^{(0)}\right) = \int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}\right) \frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}\right)}$$

Log Likelihood

$$L = \int d\mathbf{x}^{(0)} q\left(\mathbf{x}^{(0)}\right) \log\left[\int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)}\right) \frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)}\right)}\right]$$

Jensen's inequality

$$L \geq \int d\mathbf{x}^{(0\cdots T)} q\left(\mathbf{x}^{(0\cdots T)}\right) \log\left[\frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}\right)}\right]$$

$$p\left(\mathbf{x}^{(0)}\right) = \int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)}\right) \frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)}\right)}$$

Log Likelihood

$$L = \int d\mathbf{x}^{(0)} q\left(\mathbf{x}^{(0)}\right) \log\left[\int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)}\right) \frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)}\right)}\right]$$

Jensen's inequality

$$L \geq \int d\mathbf{x}^{(0\cdots T)} q\left(\mathbf{x}^{(0\cdots T)}\right) \log\left[\frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)}\right)}\right]$$

… algebra …

$$L \geq -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q\left(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}\right) D_{KL}\left(q\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) \middle|\middle| p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right)\right)$$

$$+ \text{const}$$

$$L \geq -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q\left(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}\right) D_{KL}\left(q\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) \middle\| p\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}\right)\right)$$
$$+ \text{const}$$

$$L \geq -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q\left(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}\right) D_{KL}\left(\underbrace{q\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right)}_{\text{Gaussian}} \middle|\middle| p\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}\right)\right)$$

$$+ \text{const}$$

$$L \geq -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q\left(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}\right) D_{KL}\left(q\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) \middle\| p\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}\right)\right) + \text{const}$$

$\boxed{\text{Gaussian}}$

$$L \geq -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q\left(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}\right) D_{KL}\left(q\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) \middle|\middle| p\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}\right)\right)$$

$$+ \text{const}$$

Gaussian

$$p\left(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}^{(t-1)}; f_{\mu}\left(\mathbf{x}^{(t)}, t\right), f_{\Sigma}\left(\mathbf{x}^{(t)}, t\right)\right)$$

$$L \geq -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q\left(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}\right) D_{KL}\left(q\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) \middle|\middle| p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right)\right)$$
$$+ \text{const}$$

Gaussian

$$p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}^{(t-1)}; f_{\mu}\left(\mathbf{x}^{(t)}, t\right), f_{\Sigma}\left(\mathbf{x}^{(t)}, t\right)\right)$$

Training

$$\underset{f_{\mu}\left(\mathbf{x}^{(t)}, t\right), f_{\Sigma}\left(\mathbf{x}^{(t)}, t\right)}{\operatorname{argmin}} \mathbb{E}\left[D_{KL}\left(q\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) \middle|\middle| p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right)\right)\right]$$

$$L \geq -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q\left(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}\right) D_{KL}\left(\underline{q\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right)} \middle\| \underline{p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right)}\right) + \text{const}$$

Gaussian

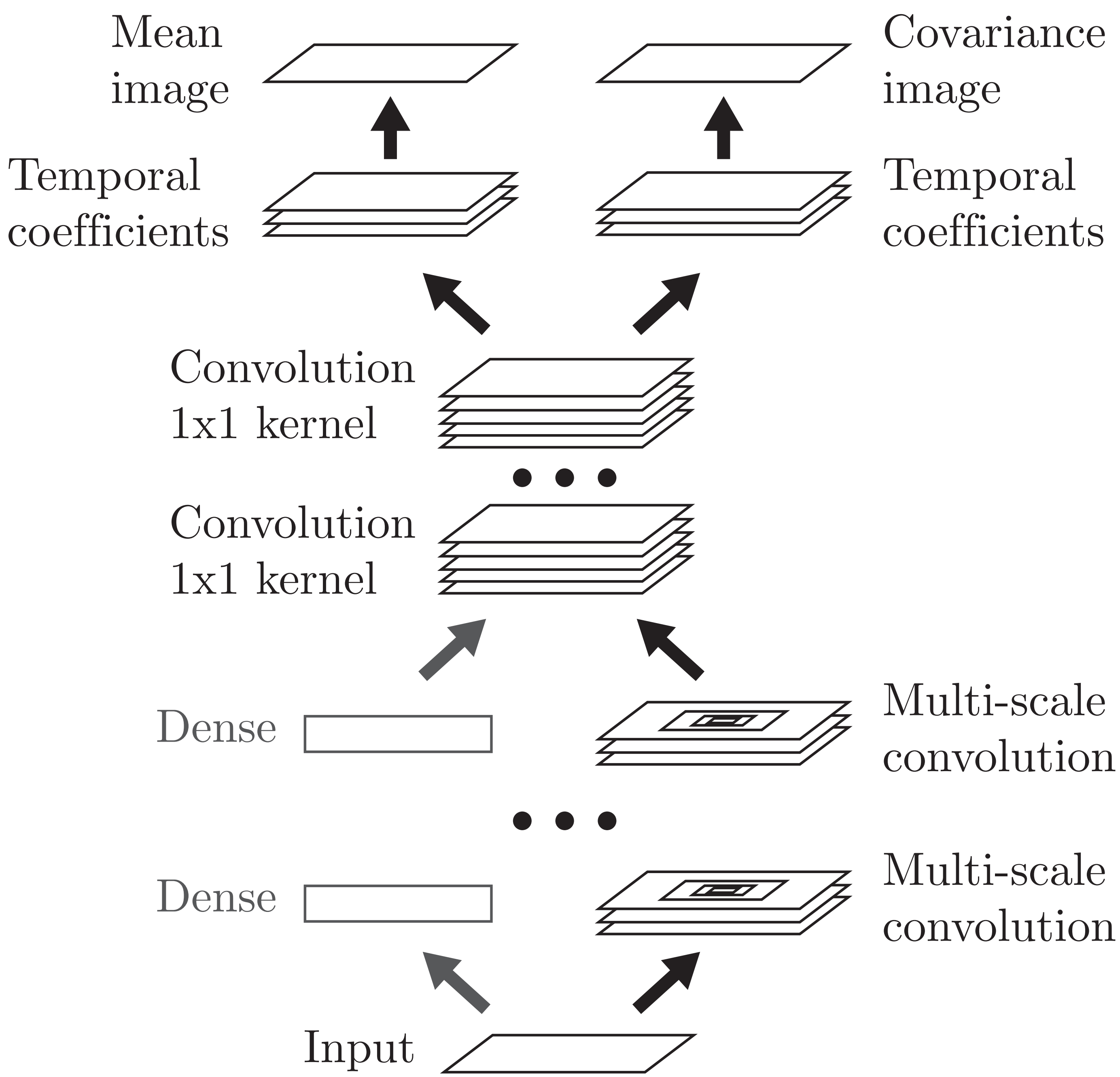
$$p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}^{(t-1)}; f_{\mu}\left(\mathbf{x}^{(t)}, t\right), f_{\Sigma}\left(\mathbf{x}^{(t)}, t\right)\right)$$

Training

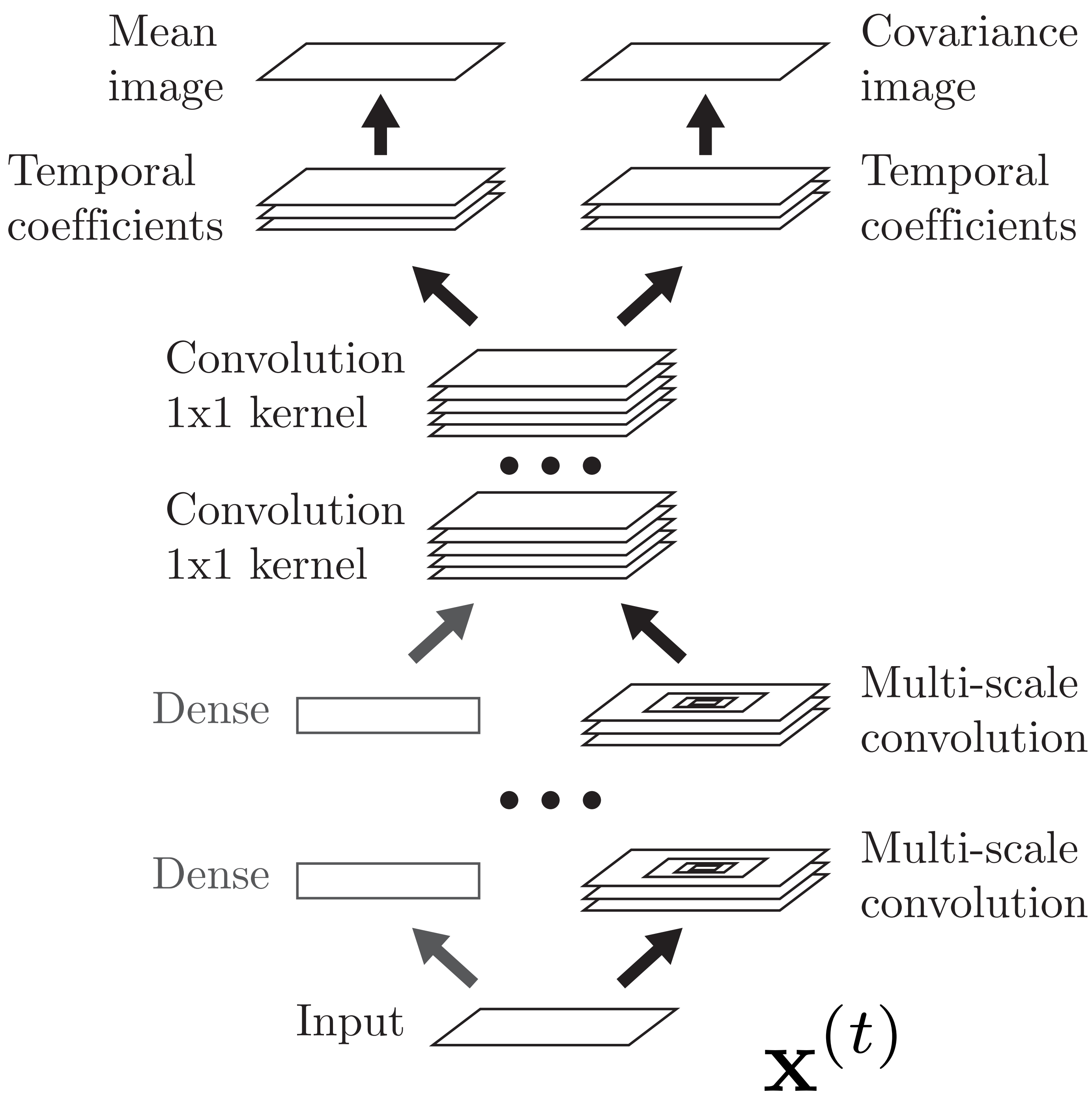
$$\underset{f_{\mu}\left(\mathbf{x}^{(t)}, t\right), f_{\Sigma}\left(\mathbf{x}^{(t)}, t\right)}{\operatorname{argmin}} \mathbb{E}\left[D_{KL}\left(q\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) \middle\| p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right)\right)\right]$$

Unsupervised learning $\rightarrow$ Regression

Use Deep Network as Function Approximator for Images

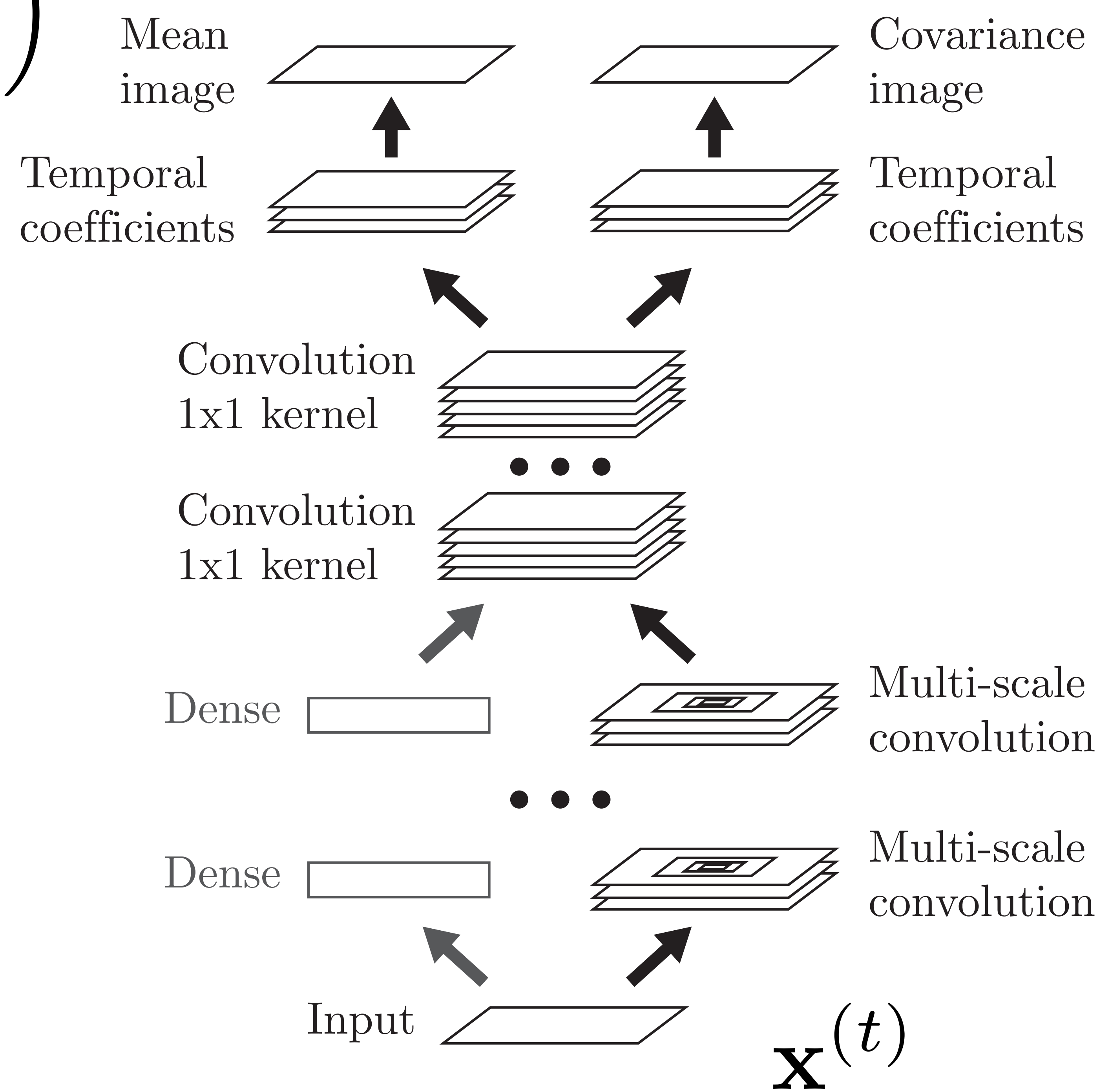


Use Deep Network as Function Approximator for Images



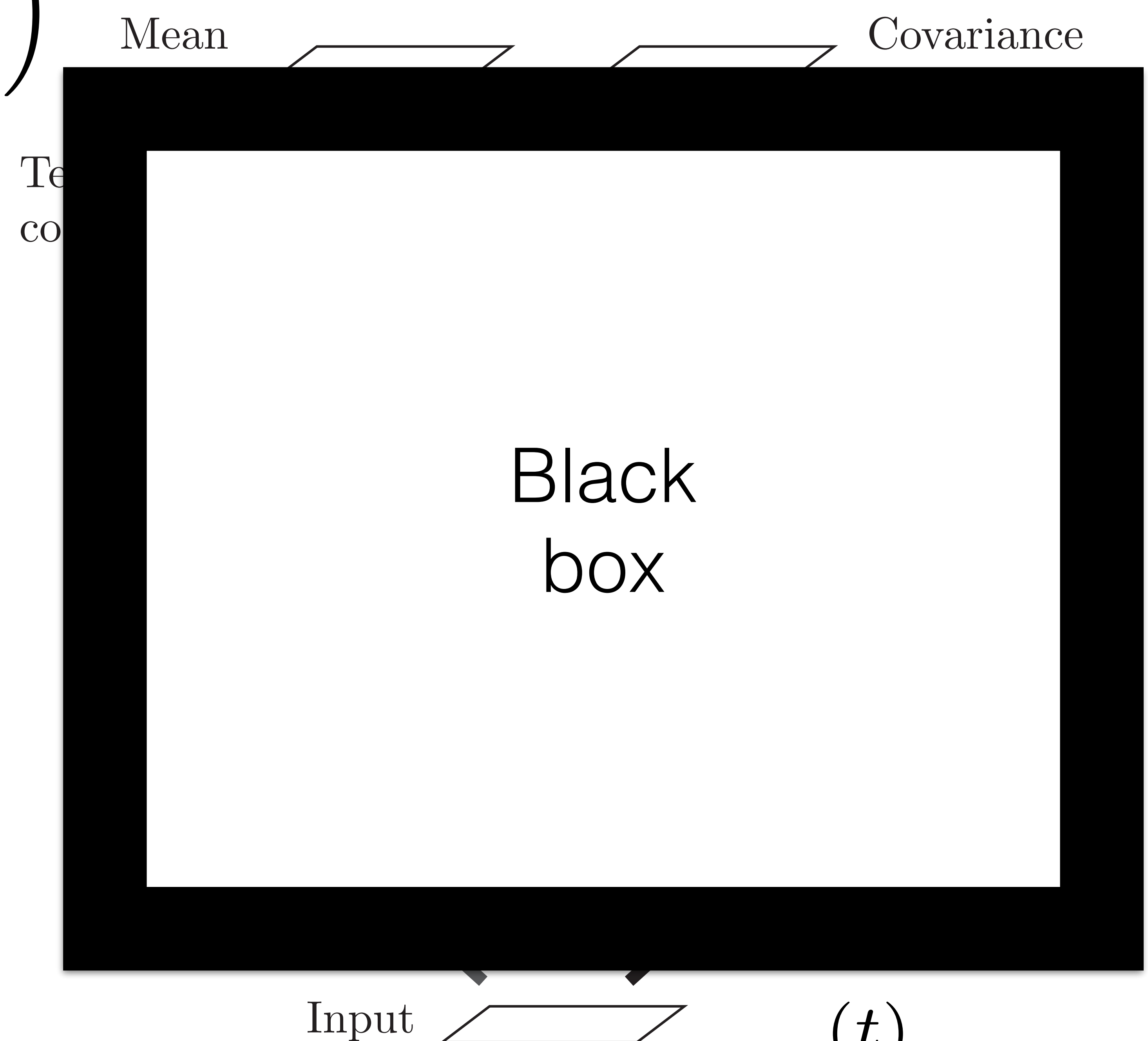
Use Deep Network as Function Approximator for Images

$$f_\mu\left(\mathbf{x}^{(t)}, t\right) \qquad f_\Sigma\left(\mathbf{x}^{(t)}, t\right)$$



Use Deep Network as Function Approximator for Images

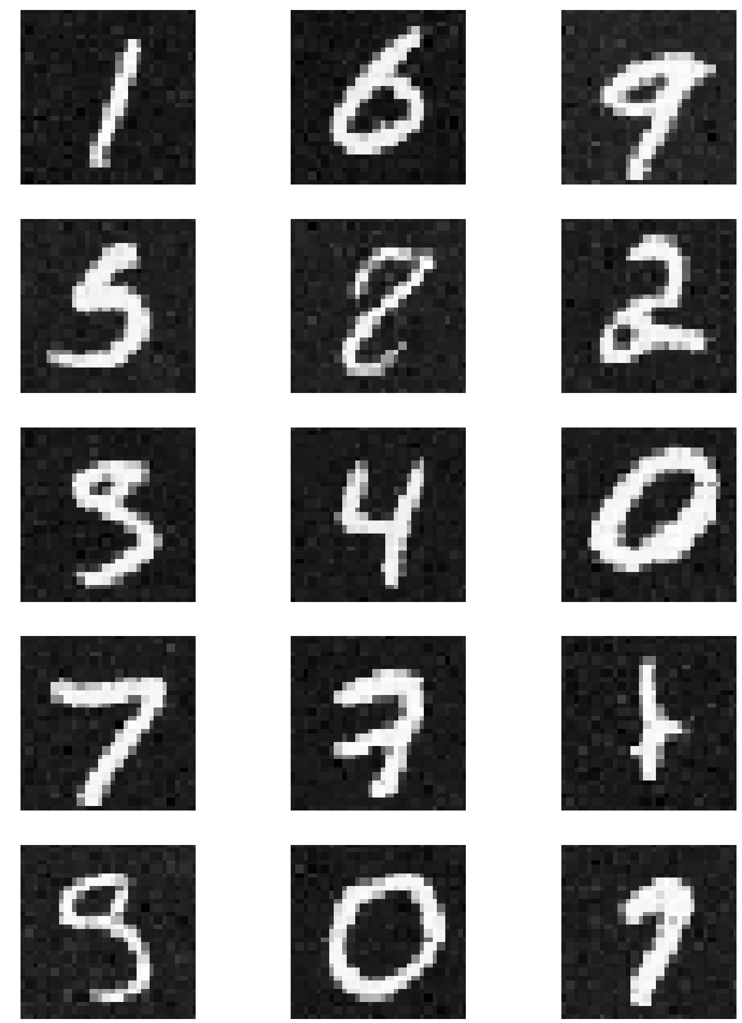
$$f_\mu\left(\mathbf{x}^{(t)}, t\right) \qquad f_\Sigma\left(\mathbf{x}^{(t)}, t\right)$$



$$\mathbf{x}^{(t)}$$

Diffusion Probabilistic Model Applied to MNIST

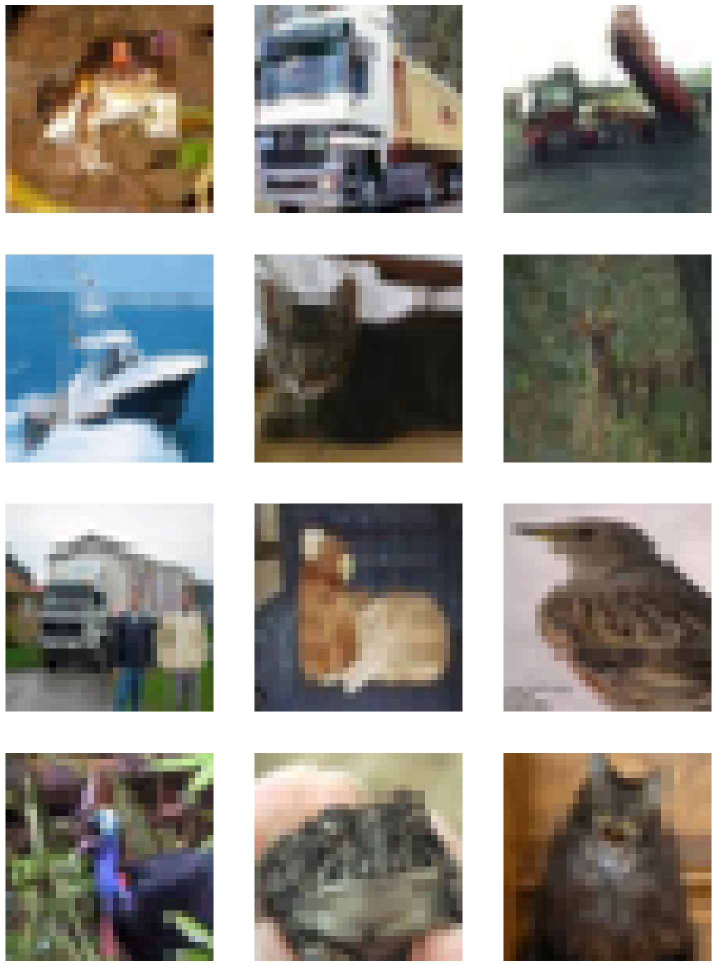
Model	**Log likelihood estimate***
Stacked CAE	121 ± 1.6 bits
DBN	138 ± 2 bits
Deep GSN	214 ± 1.1 bits
Diffusion	**220 ± 1.9 bits**
Adversarial net	225 ± 2 bits



Samples from diffusion model

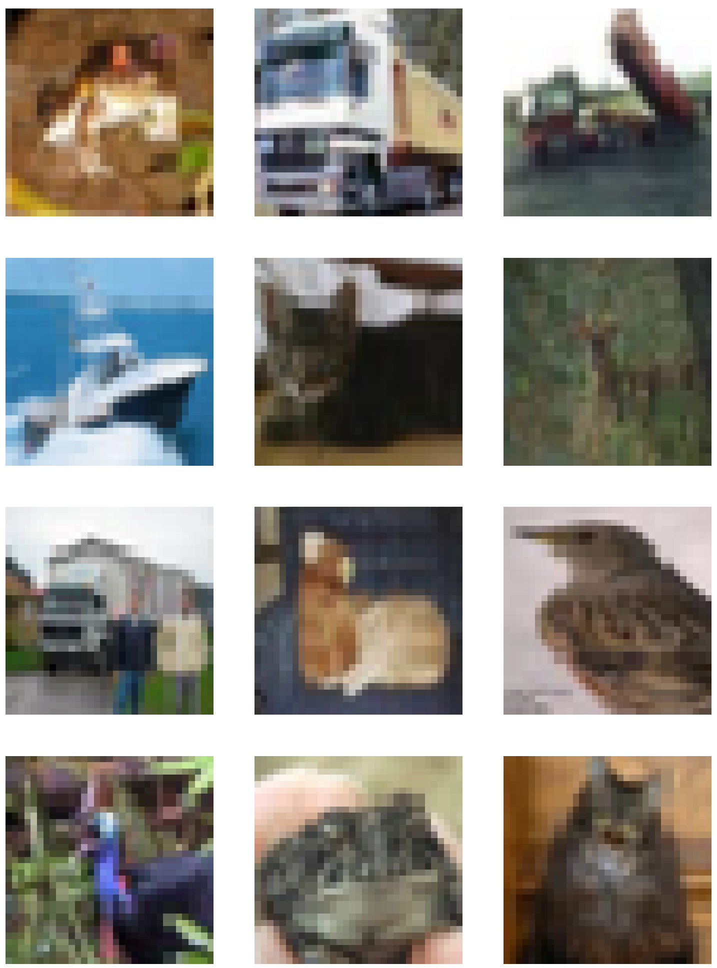
* via Parzen window code from [Goodfellow *et al*, 2014]

Diffusion Probabilistic Model Applied to CIFAR-10

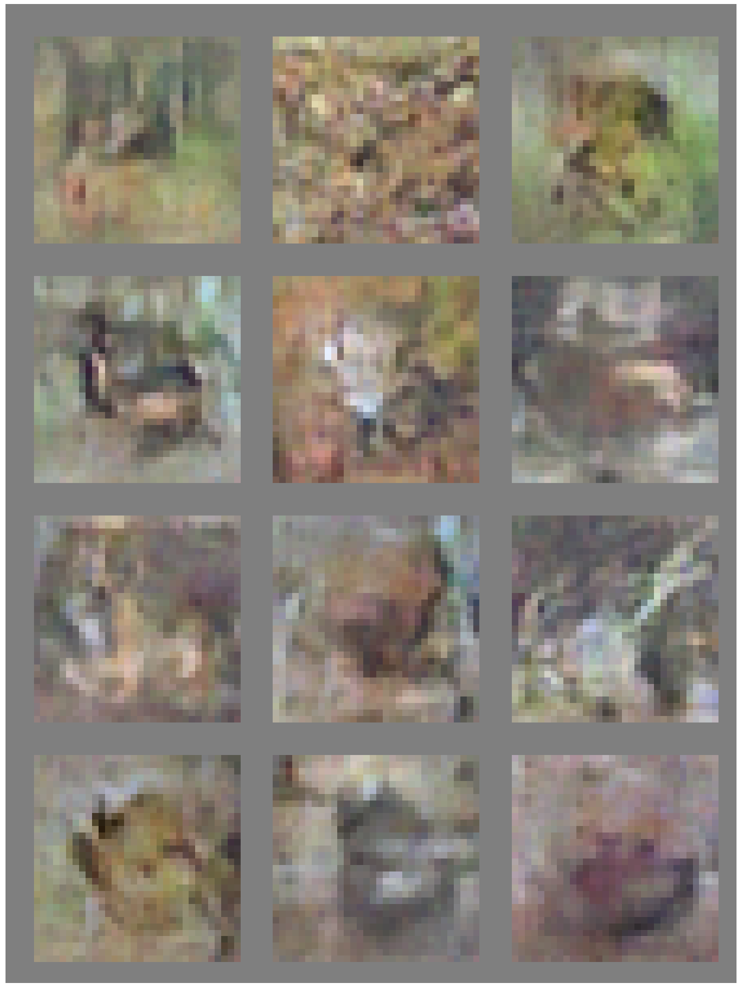


Training Data

Diffusion Probabilistic Model Applied to CIFAR-10

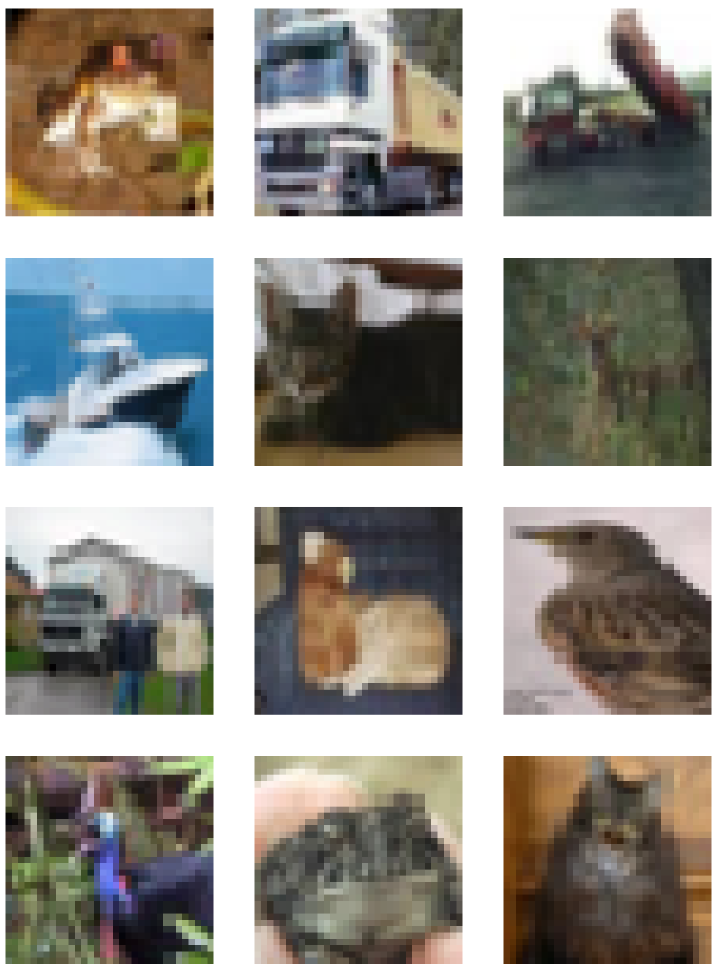


Training Data

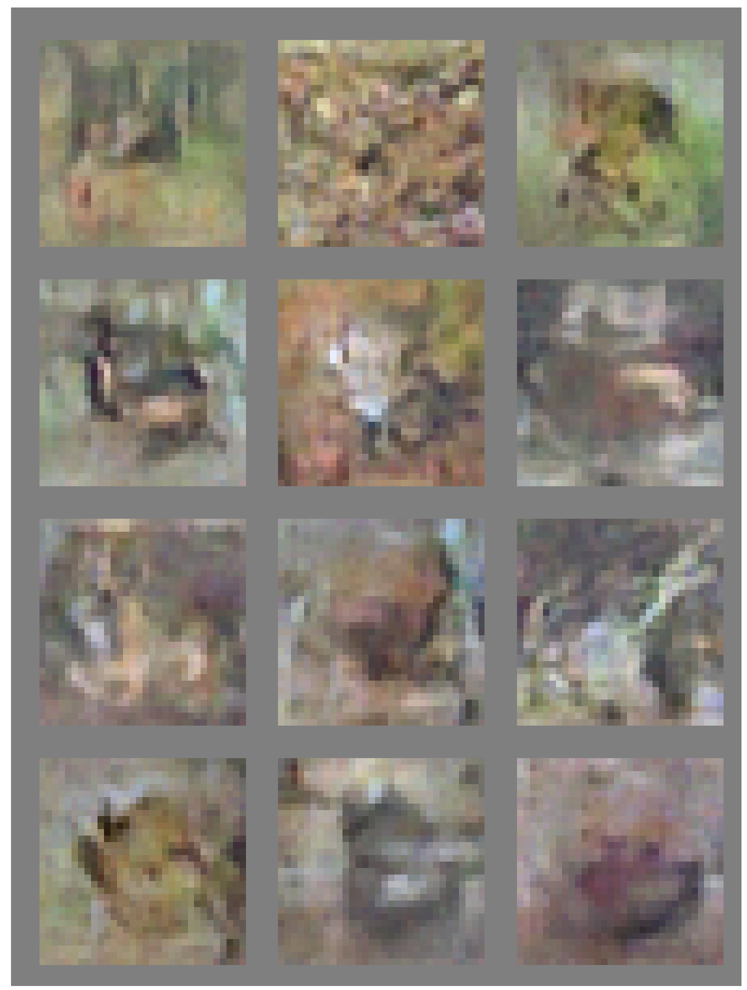


Samples from
Generative Adverserial
[Goodfellow *et al*, 2014]

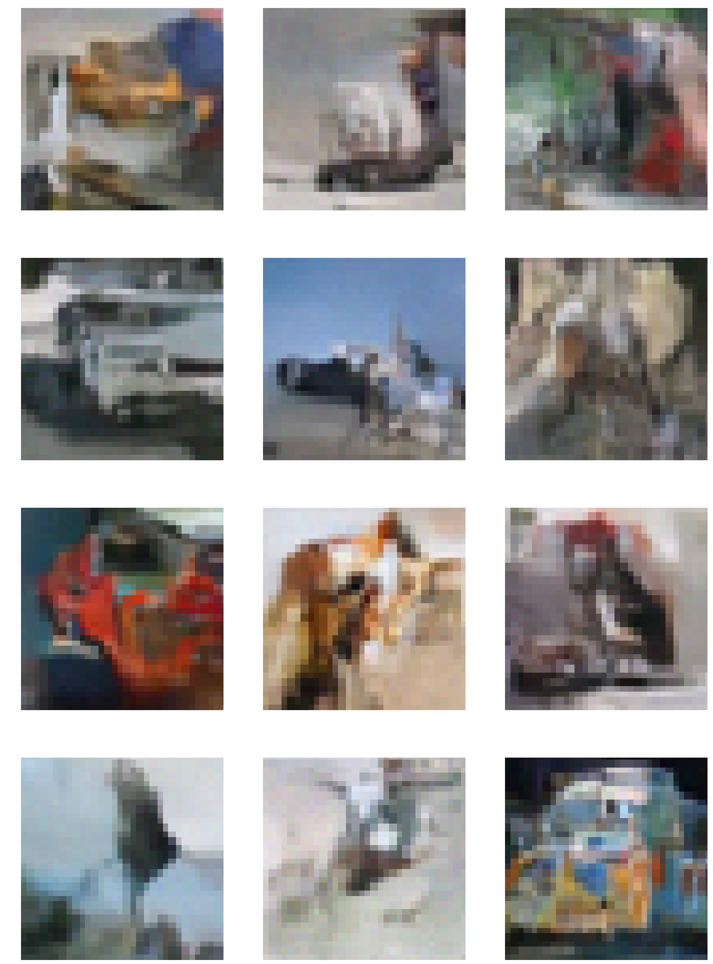
Diffusion Probabilistic Model Applied to CIFAR-10



Training Data

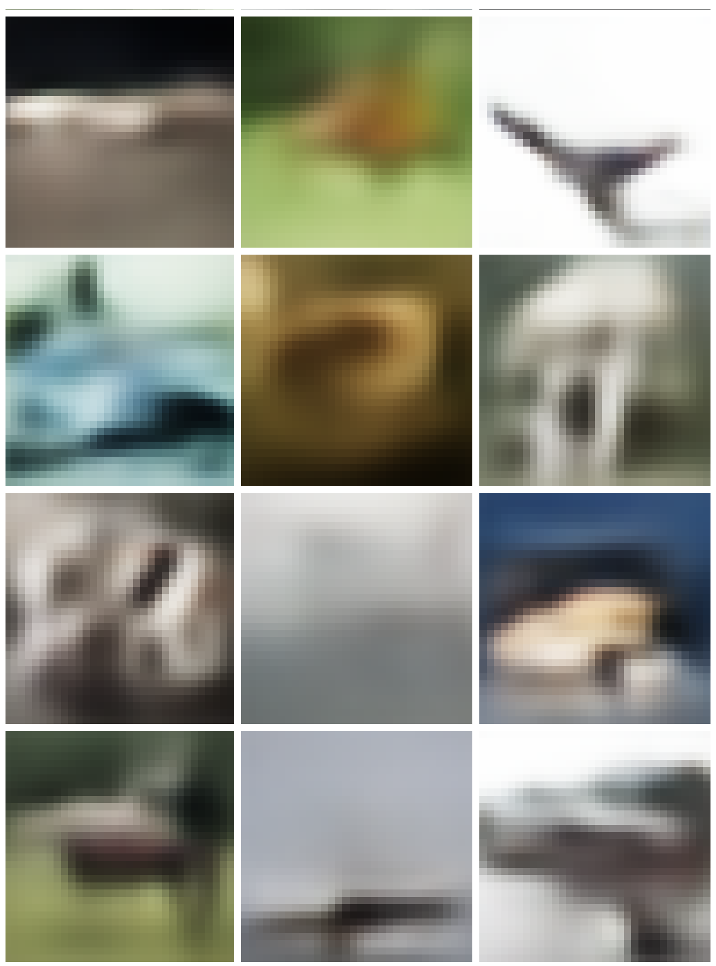


Samples from Generative Adverserial [Goodfellow *et al*, 2014]

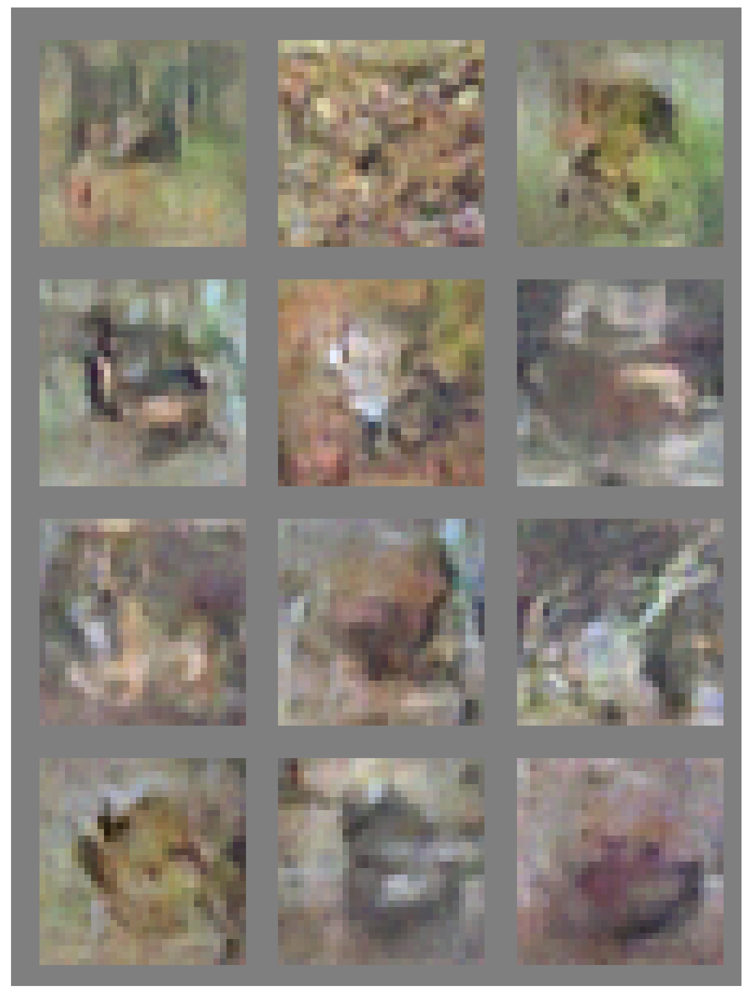


Samples from diffusion model

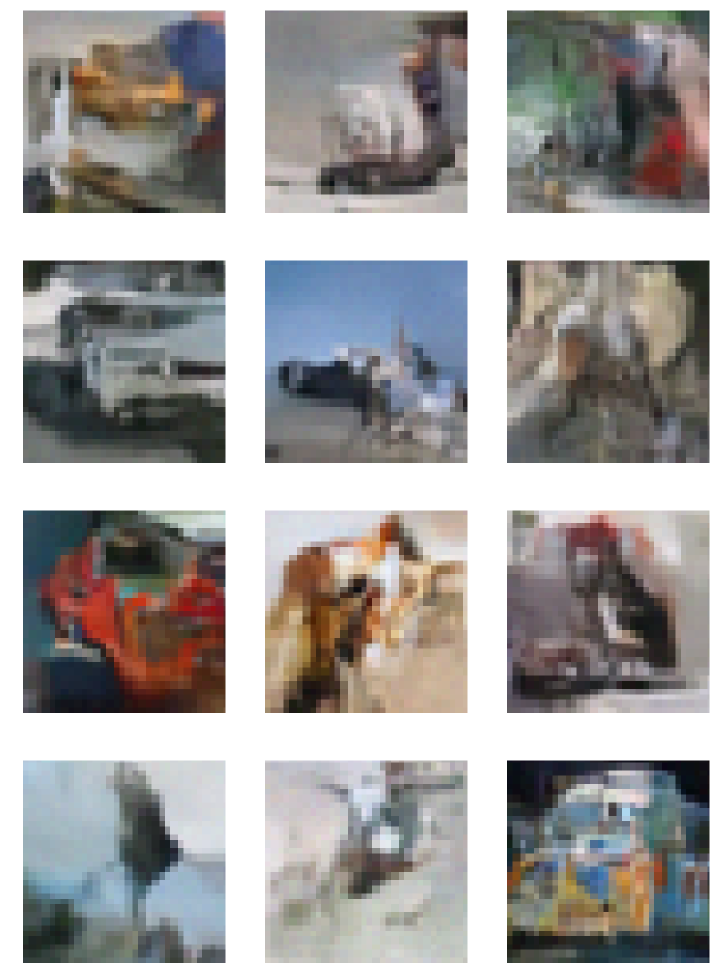
Diffusion Probabilistic Model Applied to CIFAR-10



Samples from
DRAW
[Gregor *et al*, 2015]

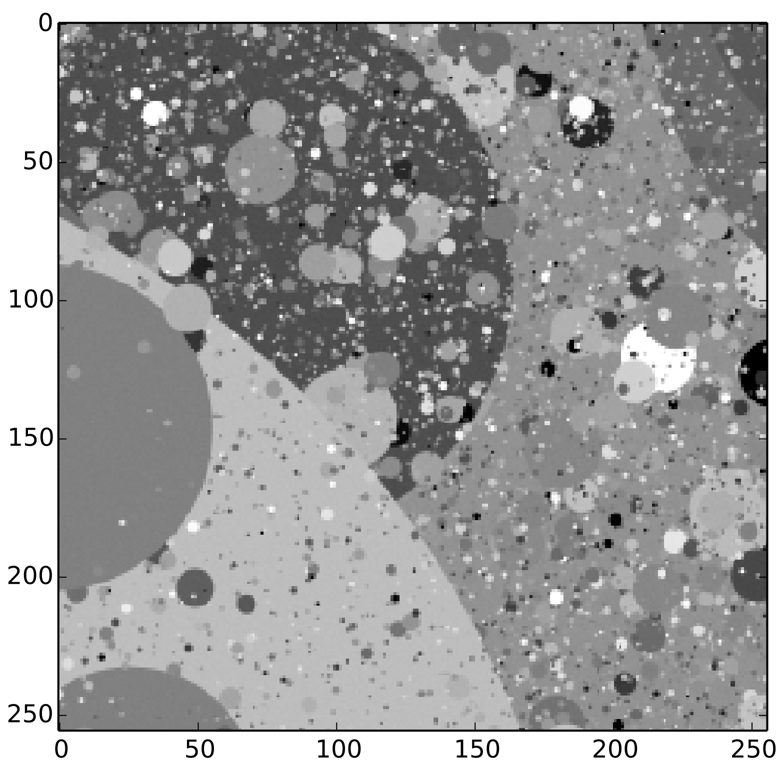


Samples from
Generative Adverserial
[Goodfellow *et al*, 2014]



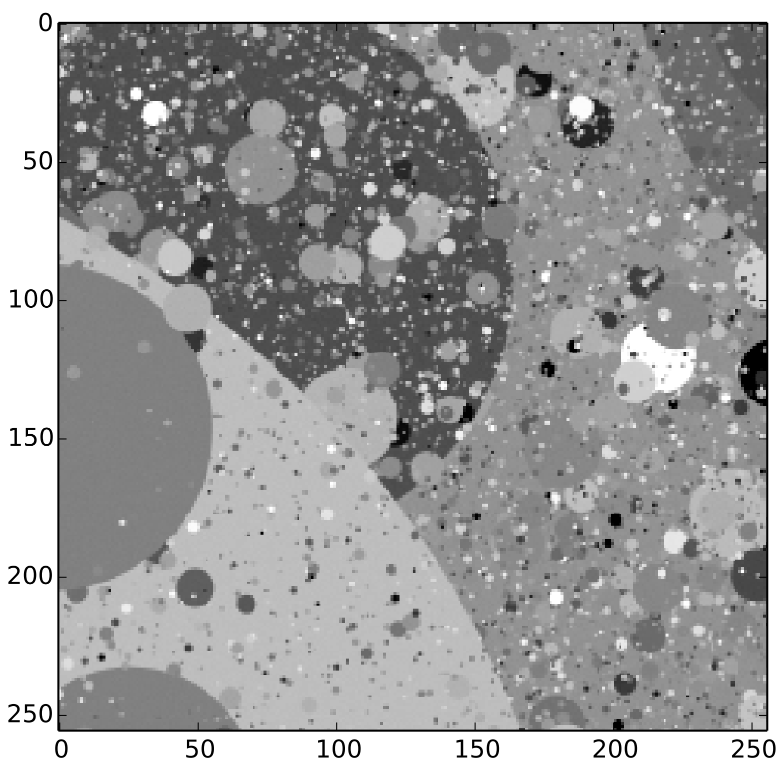
Samples from
diffusion model

Diffusion Probabilistic Model Applied to Dead Leaves

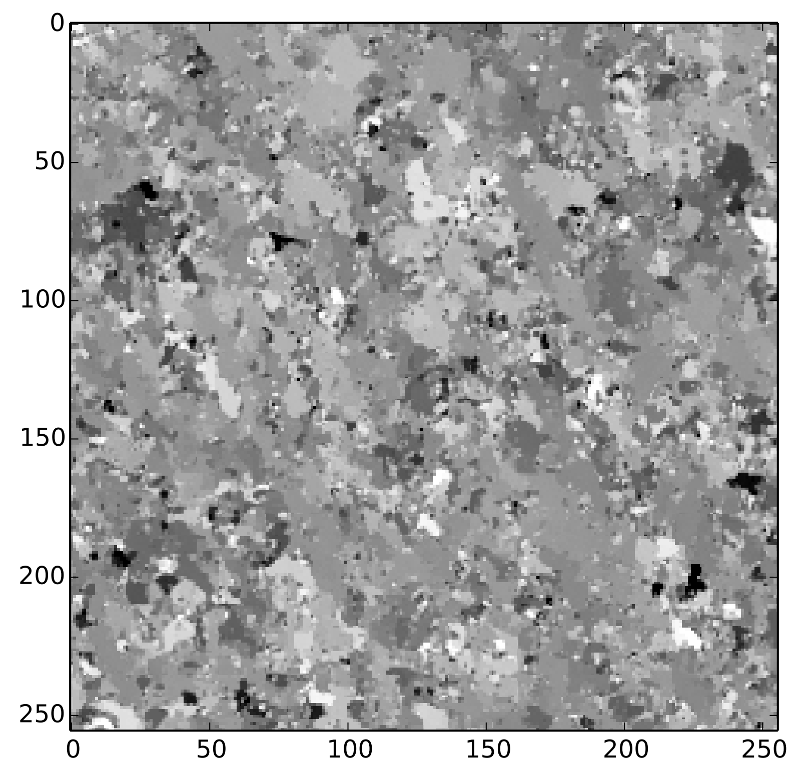


Training Data

Diffusion Probabilistic Model Applied to Dead Leaves

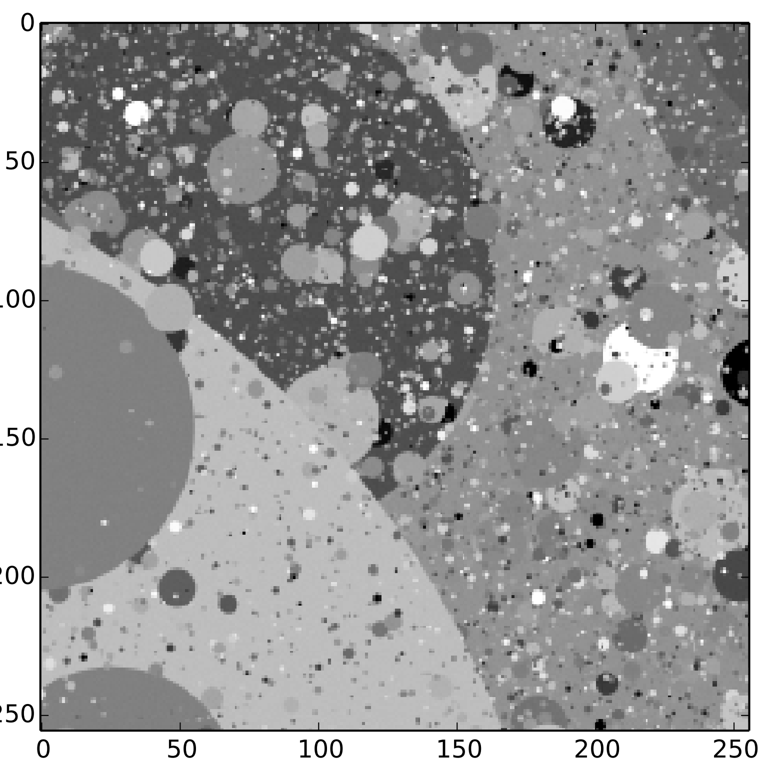


Training Data

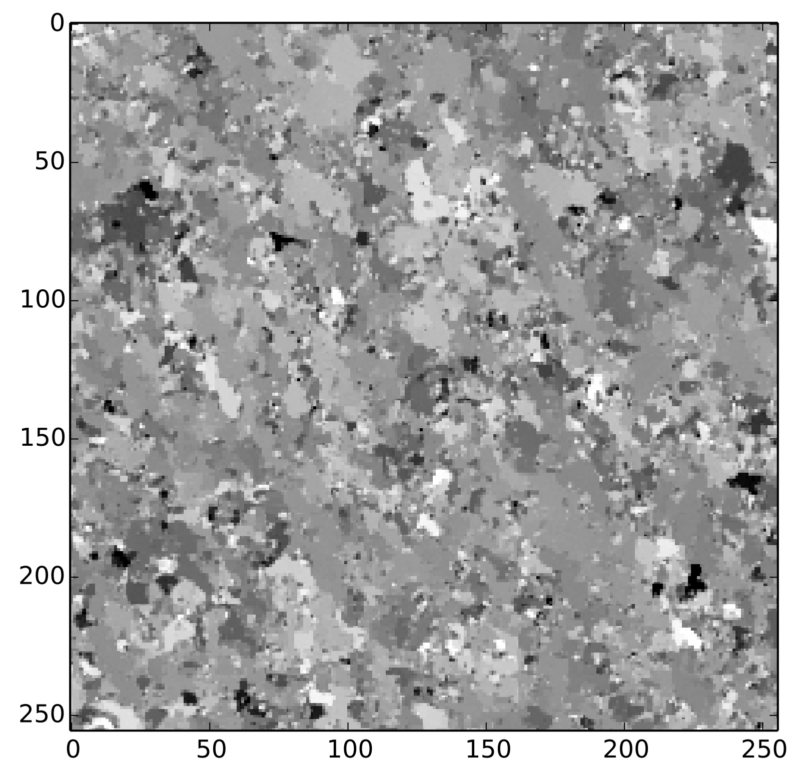


Sample from
[Theis *et al*, 2012]

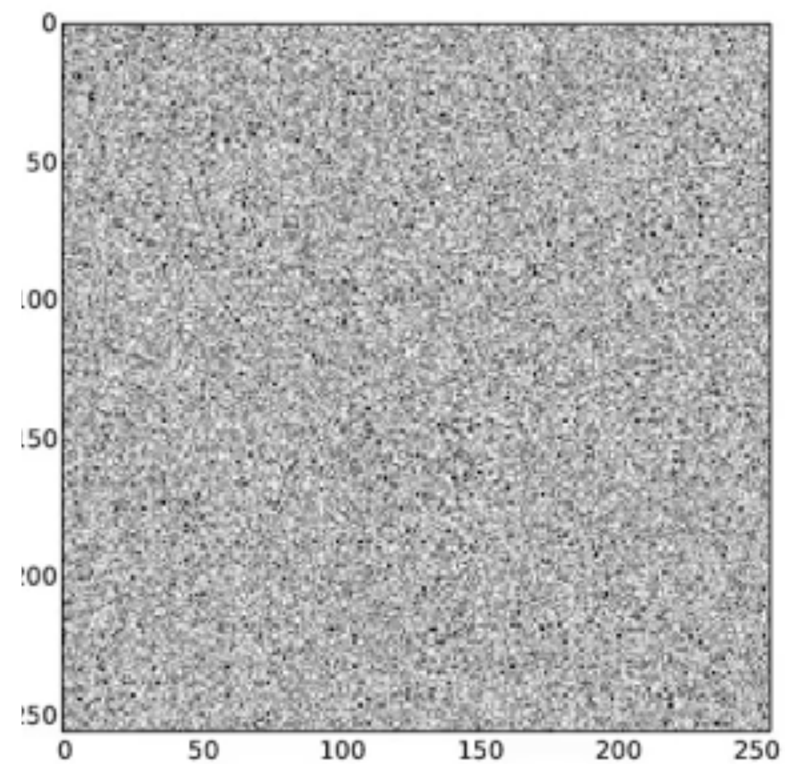
Diffusion Probabilistic Model Applied to Dead Leaves



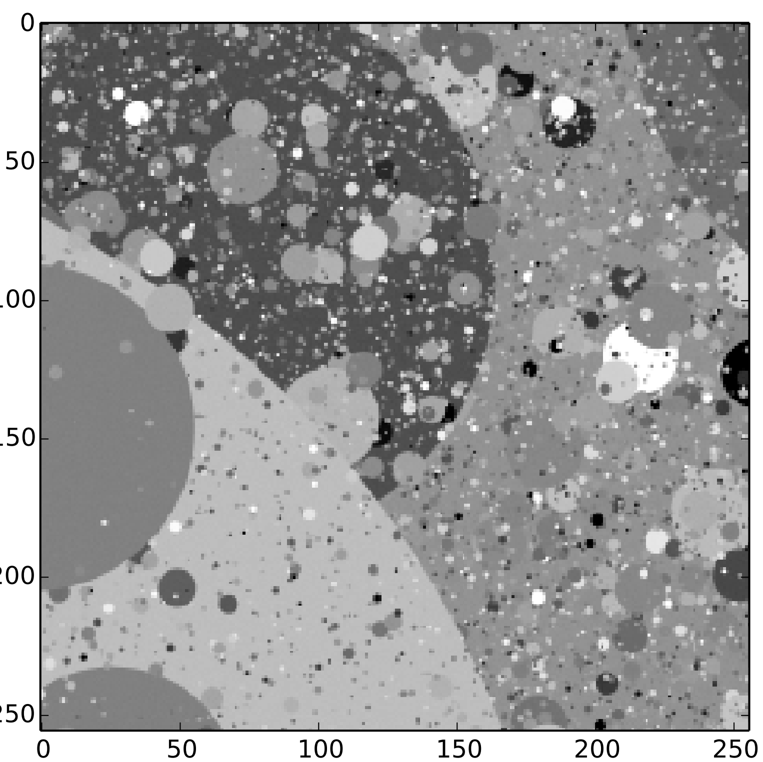
Training Data



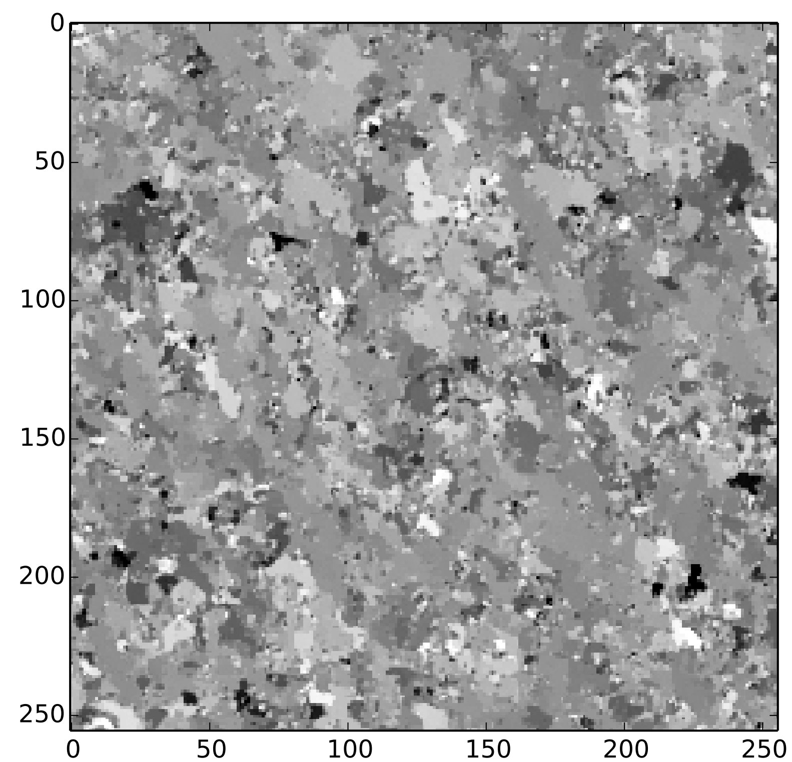
Sample from
[Theis *et al*, 2012]



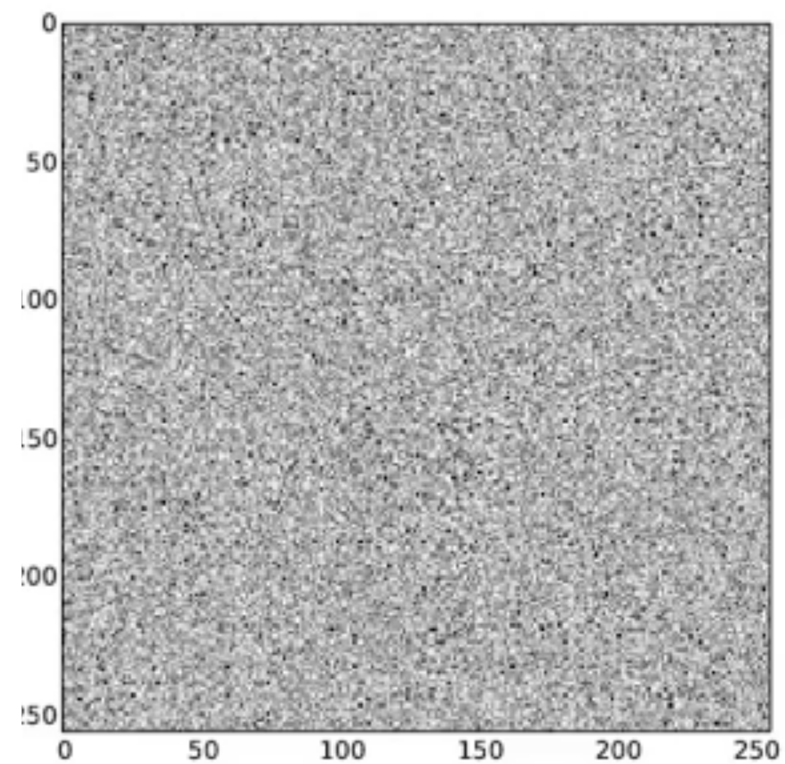
Diffusion Probabilistic Model Applied to Dead Leaves



Training Data



Sample from
[Theis *et al*, 2012]

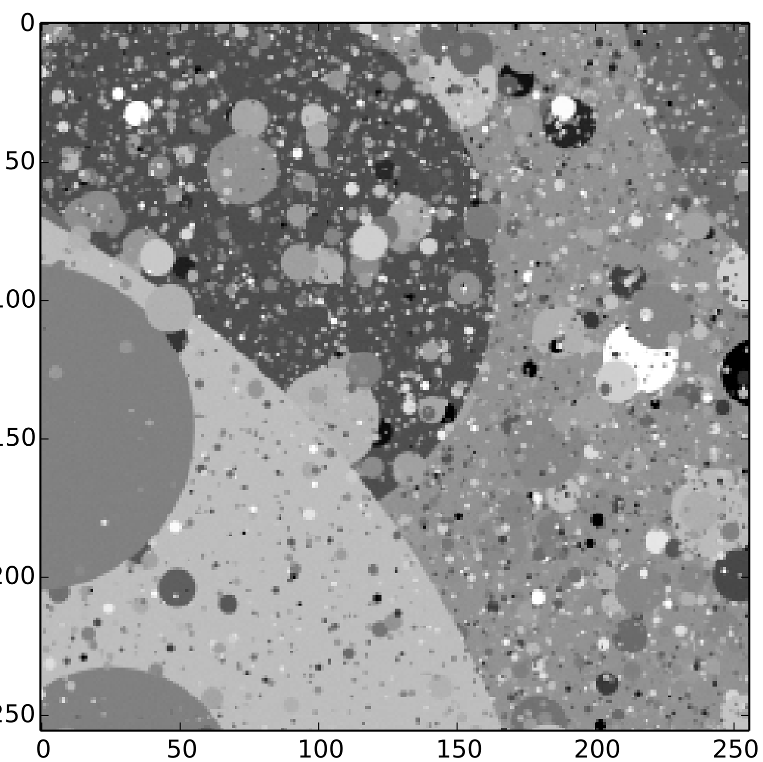


Sample from
diffusion model

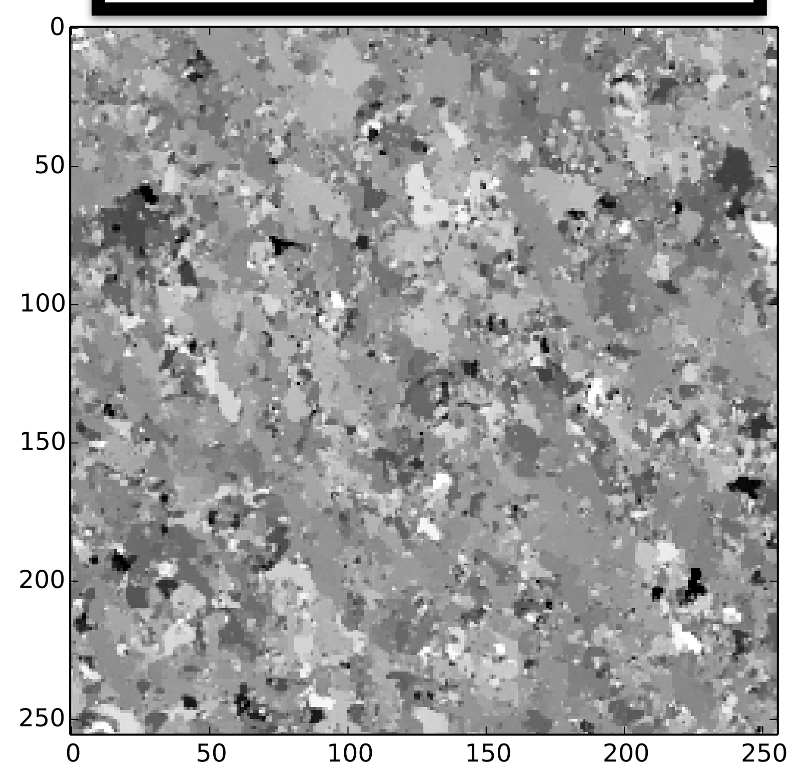
Diffusion Probabilistic Model Applied to Dead Leaves

Log likelihood
1.24 bits/pixel

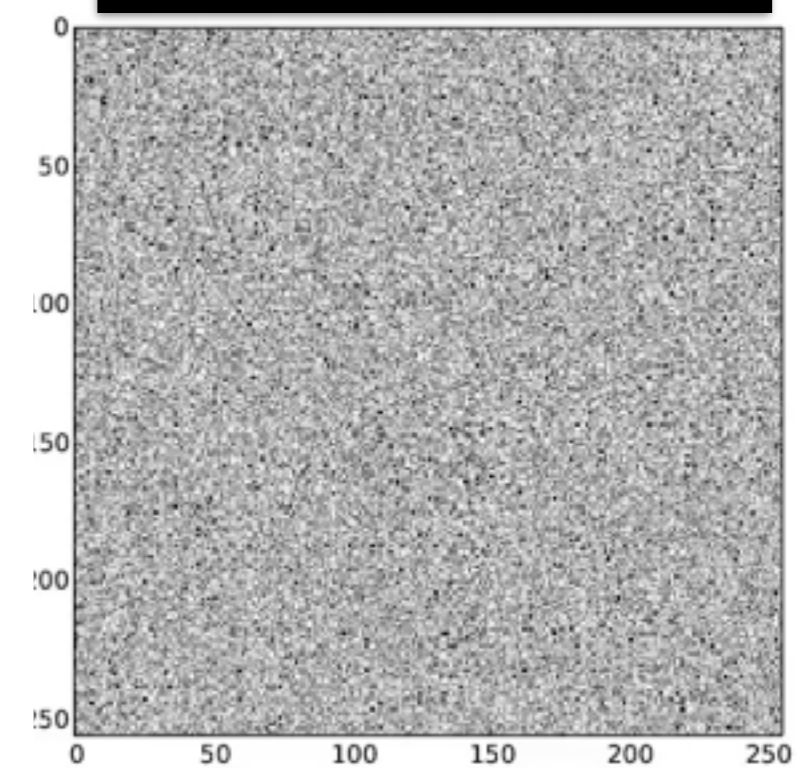
Log likelihood
1.49 bits/pixel



Training Data



Sample from
[Theis *et al*, 2012]



Sample from
diffusion model

Outline

- Motivation: The promise of deep unsupervised learning
- Physical intuition: Diffusion processes and time reversal
- Diffusion probabilistic model: Derivation and experimental results
  - Algorithm
  - **Deep convolutional network: Universal function approximator**
  - Multiplying distributions: Inputation, denoising, computing posteriors

Deep Networks

- Extremely flexible, parametric, function approximation

Deep Networks

- Extremely flexible, parametric, function approximation

- **Single layer:** linear transformation, pointwise nonlinearity

Deep Networks

- Extremely flexible, parametric, function approximation

- **Single layer:** linear transformation, pointwise nonlinearity

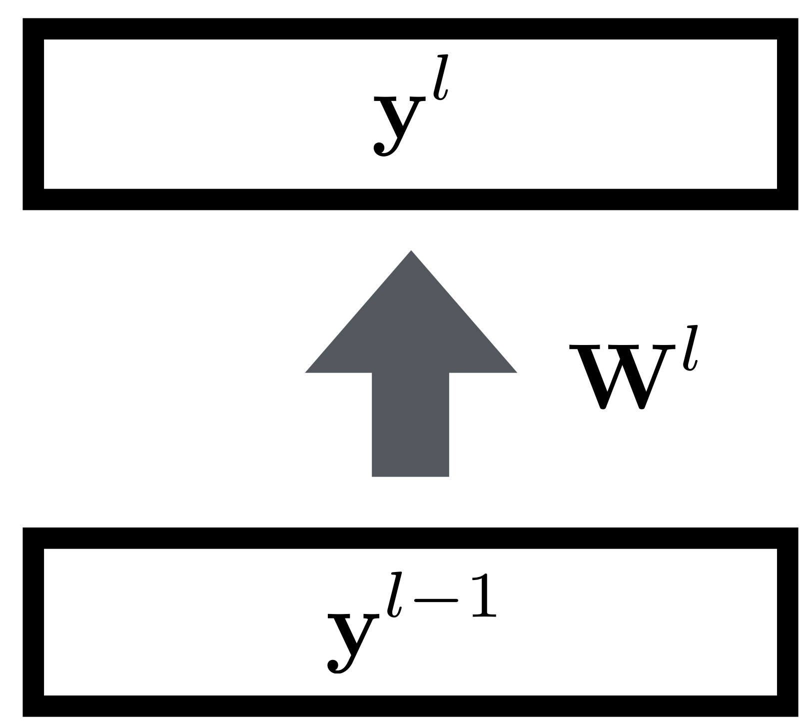
$$\mathbf{y}^l = \sigma\left(\mathbf{W}^l \mathbf{y}^{l-1}\right)$$

Deep Networks

- Extremely flexible, parametric, function approximation

- **Single layer:** linear transformation, pointwise nonlinearity

$$\mathbf{y}^{l} = \sigma\left(\mathbf{W}^{l}\mathbf{y}^{l-1}\right)$$



Deep Networks

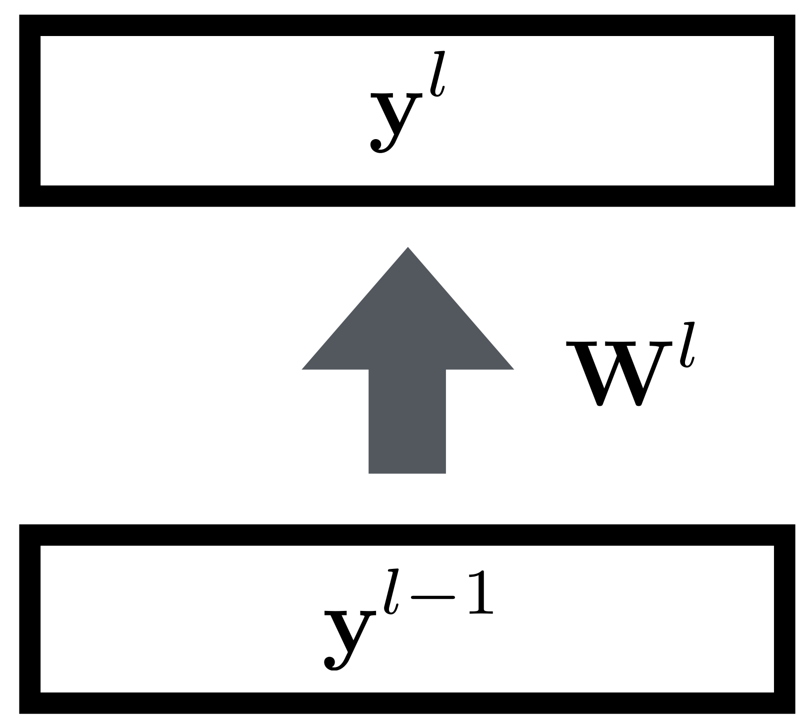
- Extremely flexible, parametric, function approximation

- **Single layer:** linear transformation, pointwise nonlinearity

$$\mathbf{y}^l = \sigma\left(\mathbf{W}^l \mathbf{y}^{l-1}\right)$$

$$\sigma\left(u\right) \equiv \text{leaky ReLU}$$

$$= \begin{cases} u & u \geq 0 \\ 0.05u & u < 0 \end{cases}$$



Deep Networks

- Extremely flexible, parametric, function approximation

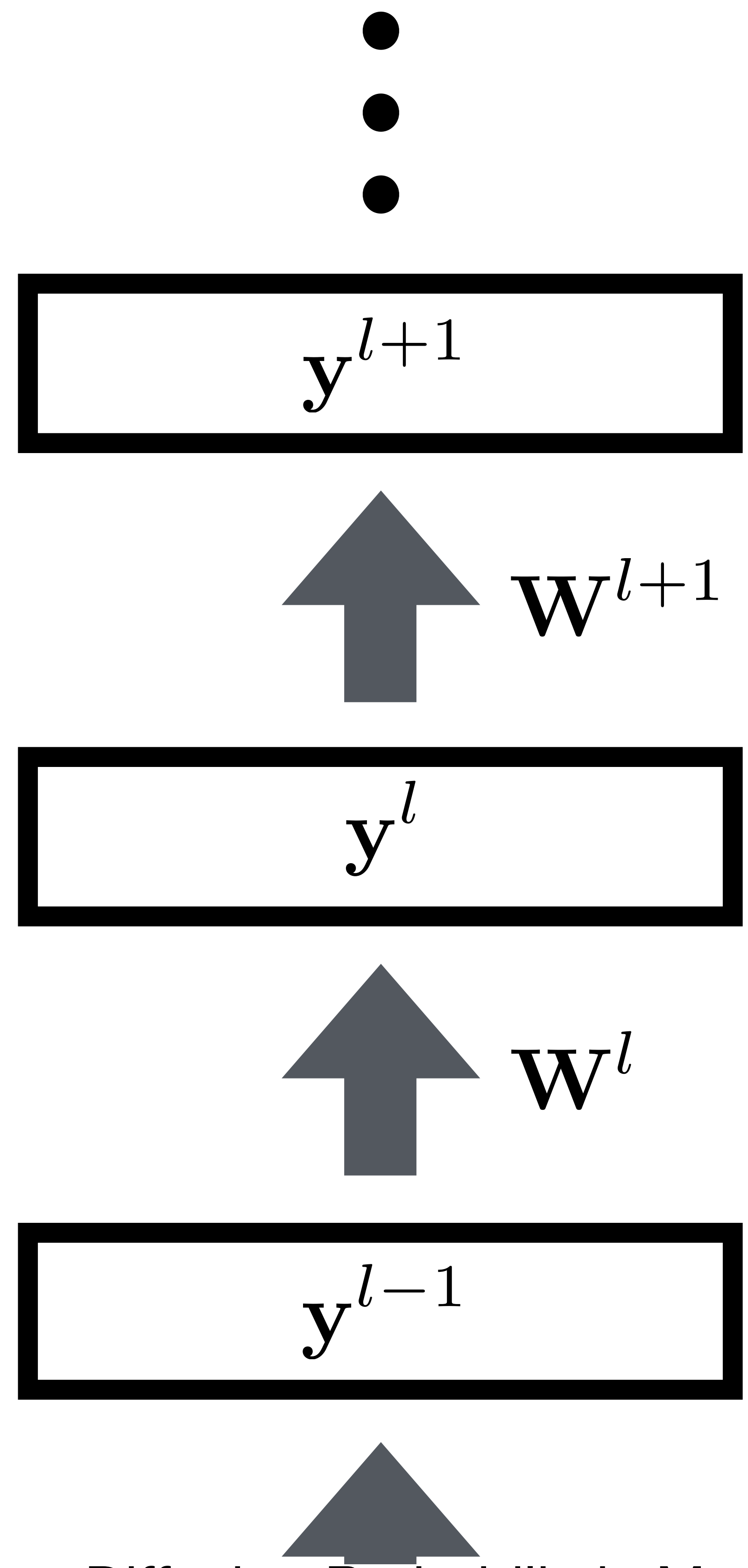
- **Single layer:** linear transformation, pointwise nonlinearity

Deep Networks

- Extremely flexible, parametric, function approximation
- **Single layer:** linear transformation, pointwise nonlinearity
- **Deep network:** stack single layers

Deep Networks

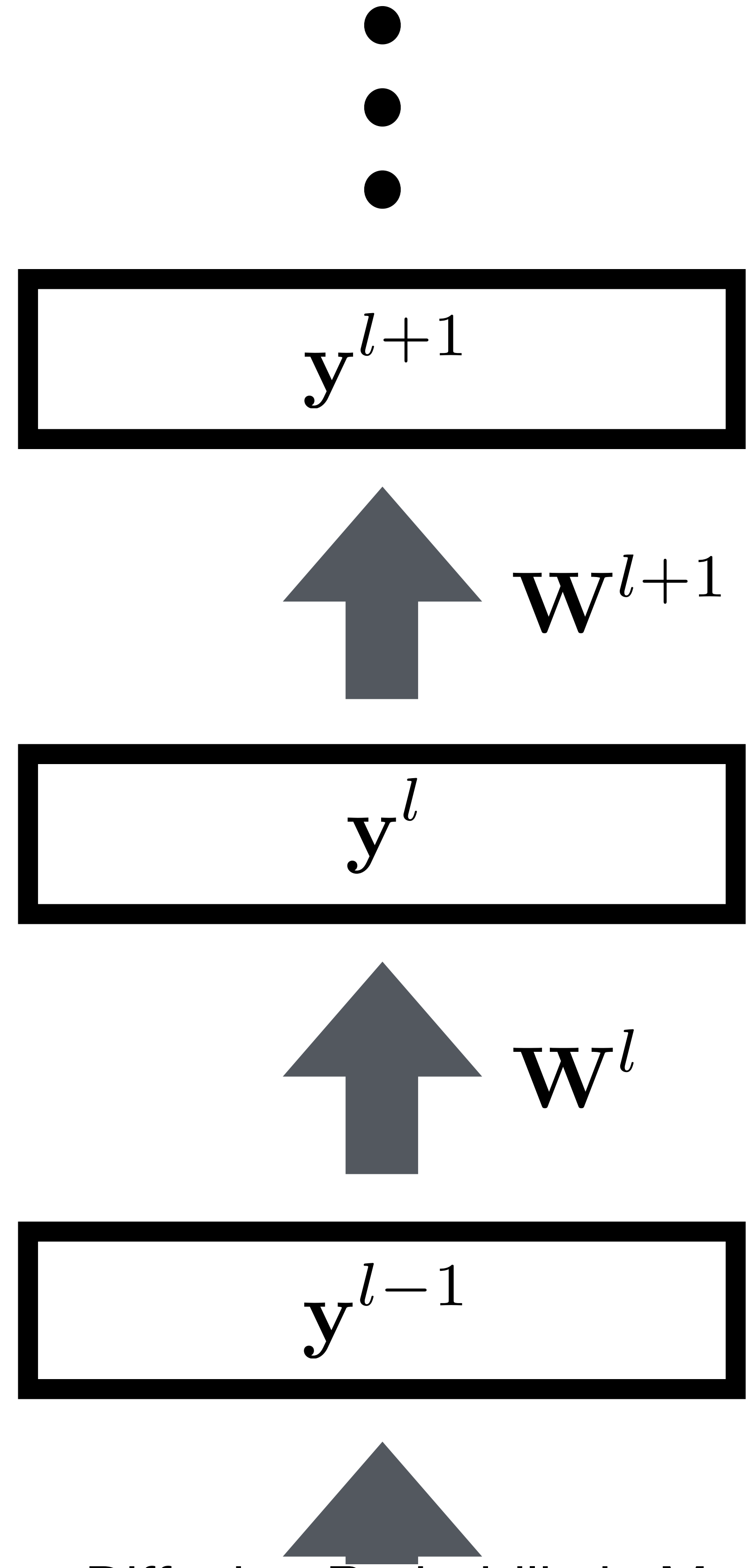
- Extremely flexible, parametric, function approximation
- **Single layer:** linear transformation, pointwise nonlinearity
- **Deep network:** stack single layers



Deep Networks

- Extremely flexible, parametric, function approximation
- **Single layer:** linear transformation, pointwise nonlinearity
- **Deep network:** stack single layers

$$\mathbf{y}^L = \sigma\left(\mathbf{W}^L \sigma\left(\mathbf{W}^{L-1} \cdots \sigma\left(\mathbf{W}^1 \mathbf{y}^0\right)\right)\right)$$

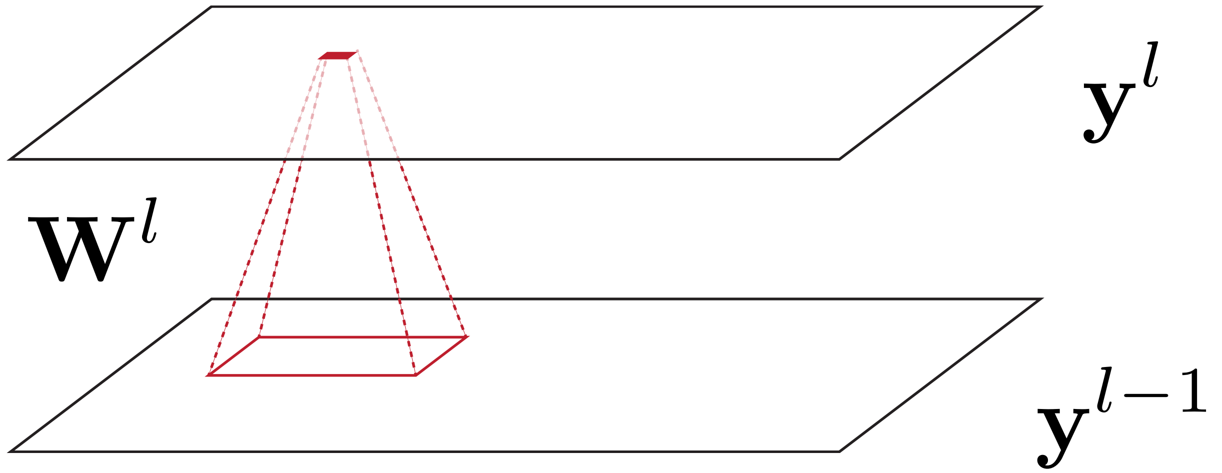


Convolutional Neural Network

- Single convolutional layer:
  - Same linear transform for every pixel
  - Pointwise nonlinearity

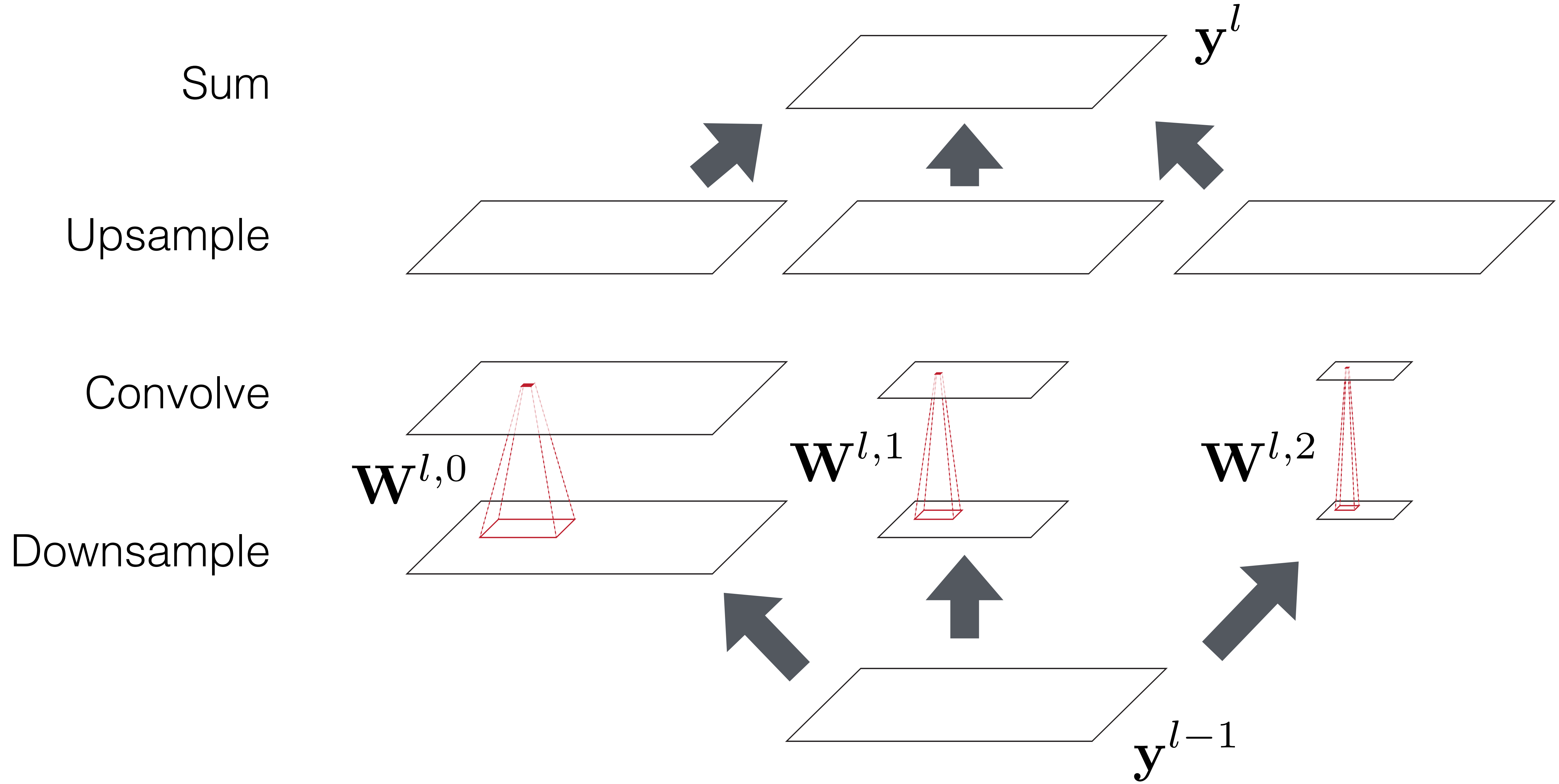
Convolutional Neural Network

- Single convolutional layer:
  - Same linear transform for every pixel
  - Pointwise nonlinearity



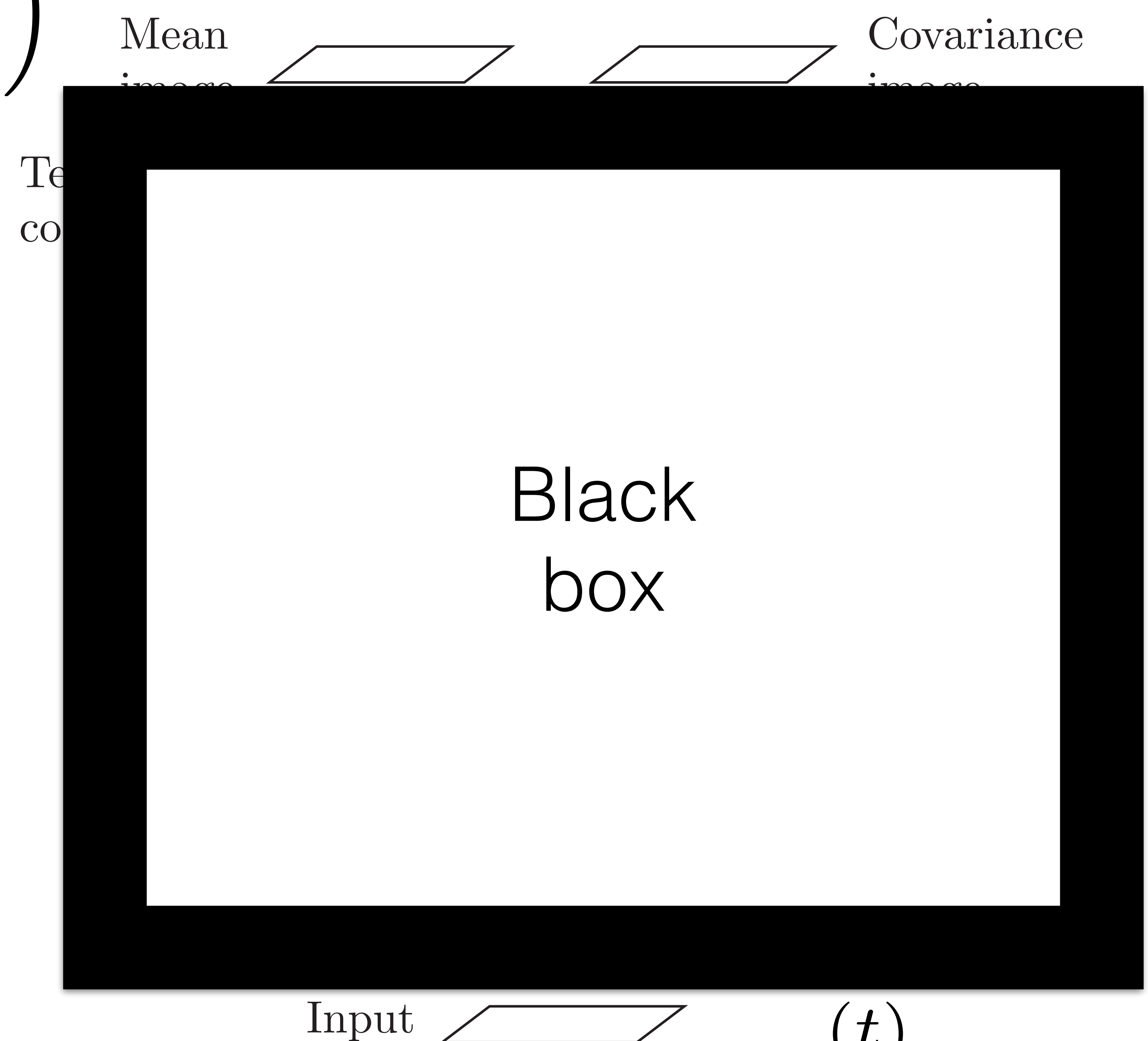
Multiscale Convolution

- Single multi-scale convolutional layer:



Deep Network Architecture for Diffusion

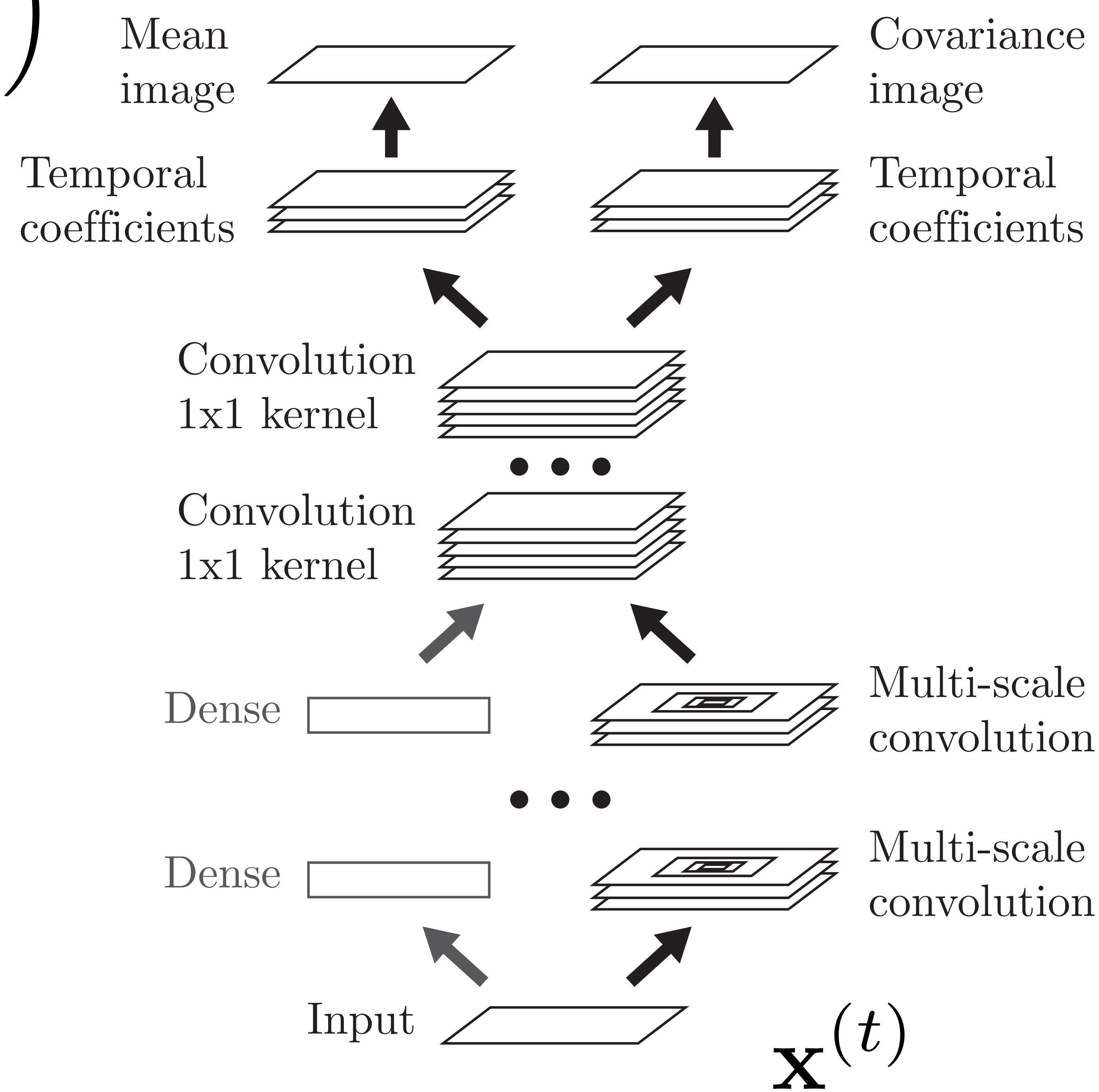
$$f_\mu\left(\mathbf{x}^{(t)}, t\right) \qquad f_\Sigma\left(\mathbf{x}^{(t)}, t\right)$$



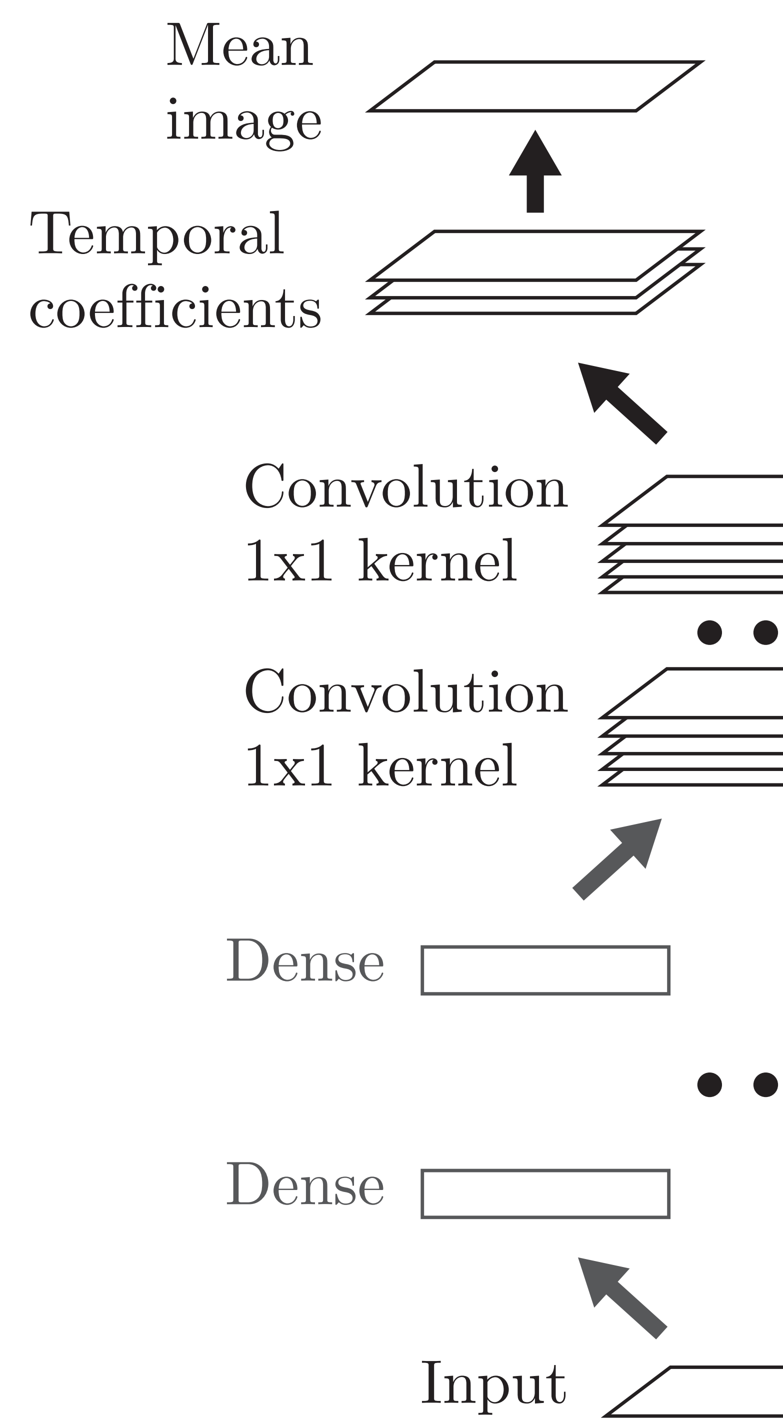
$$\mathbf{x}^{(t)}$$

Deep Network Architecture for Diffusion

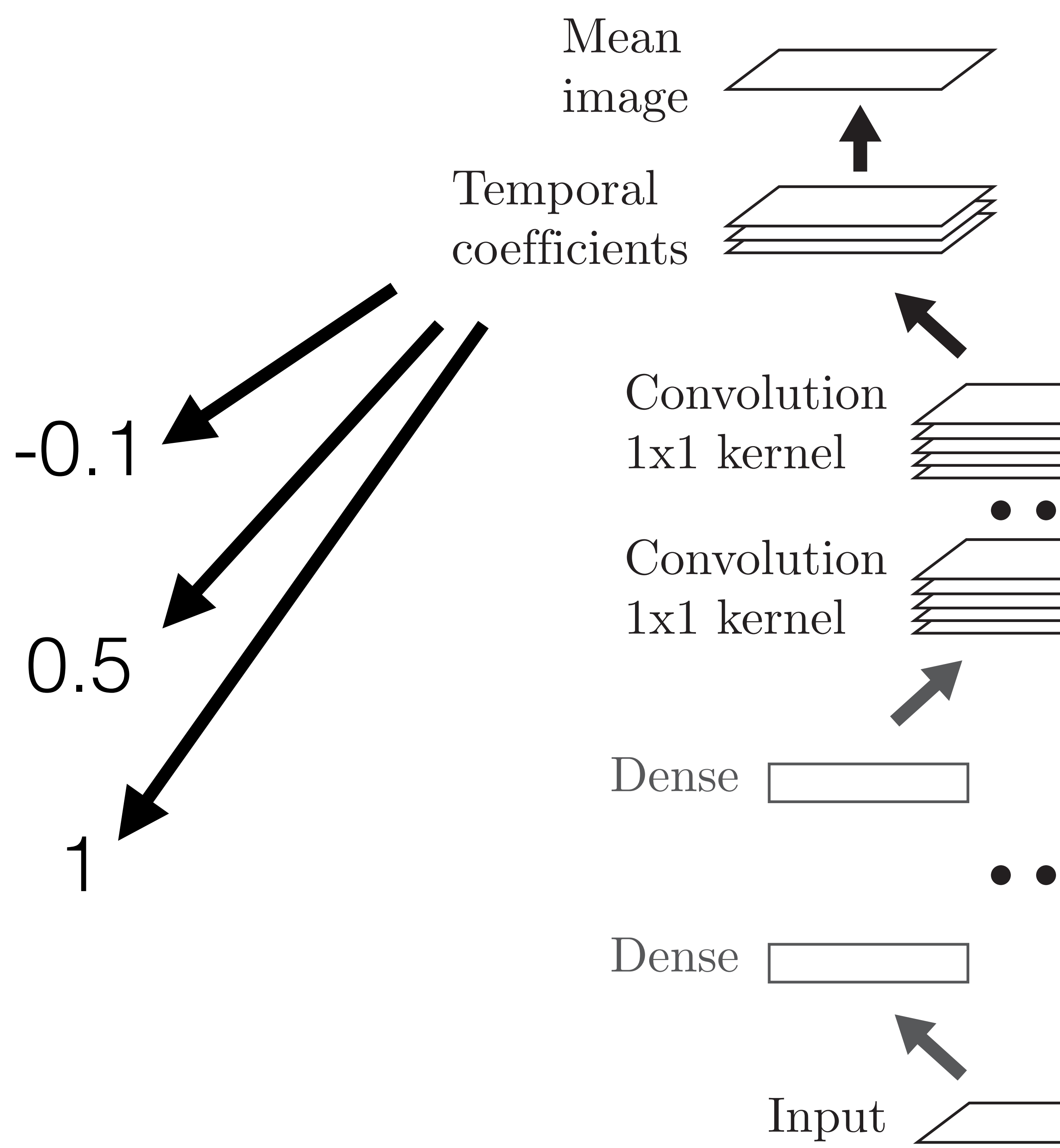
$$f_\mu\left(\mathbf{x}^{(t)}, t\right) \qquad f_\Sigma\left(\mathbf{x}^{(t)}, t\right)$$



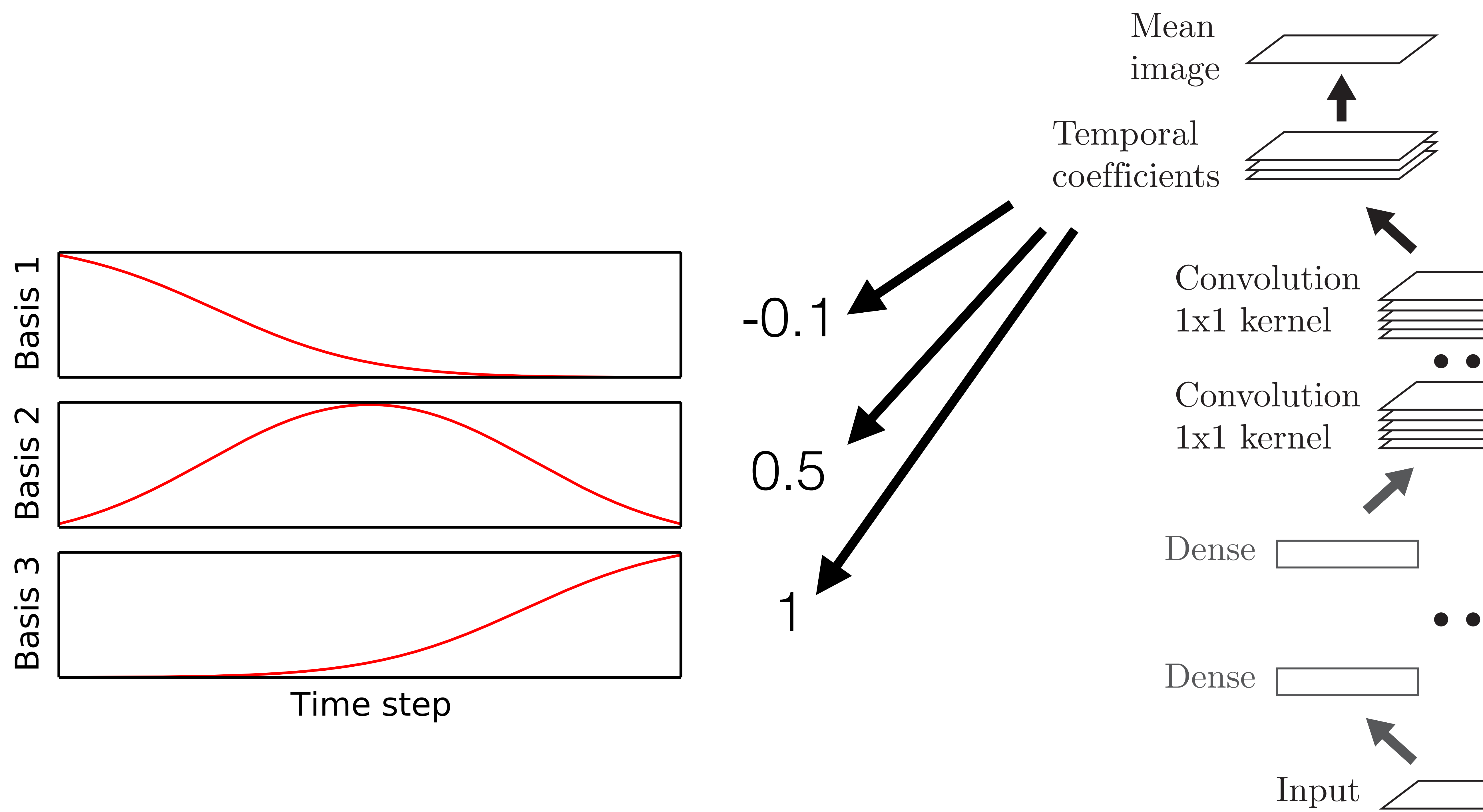
Time Dependence using Temporal Basis



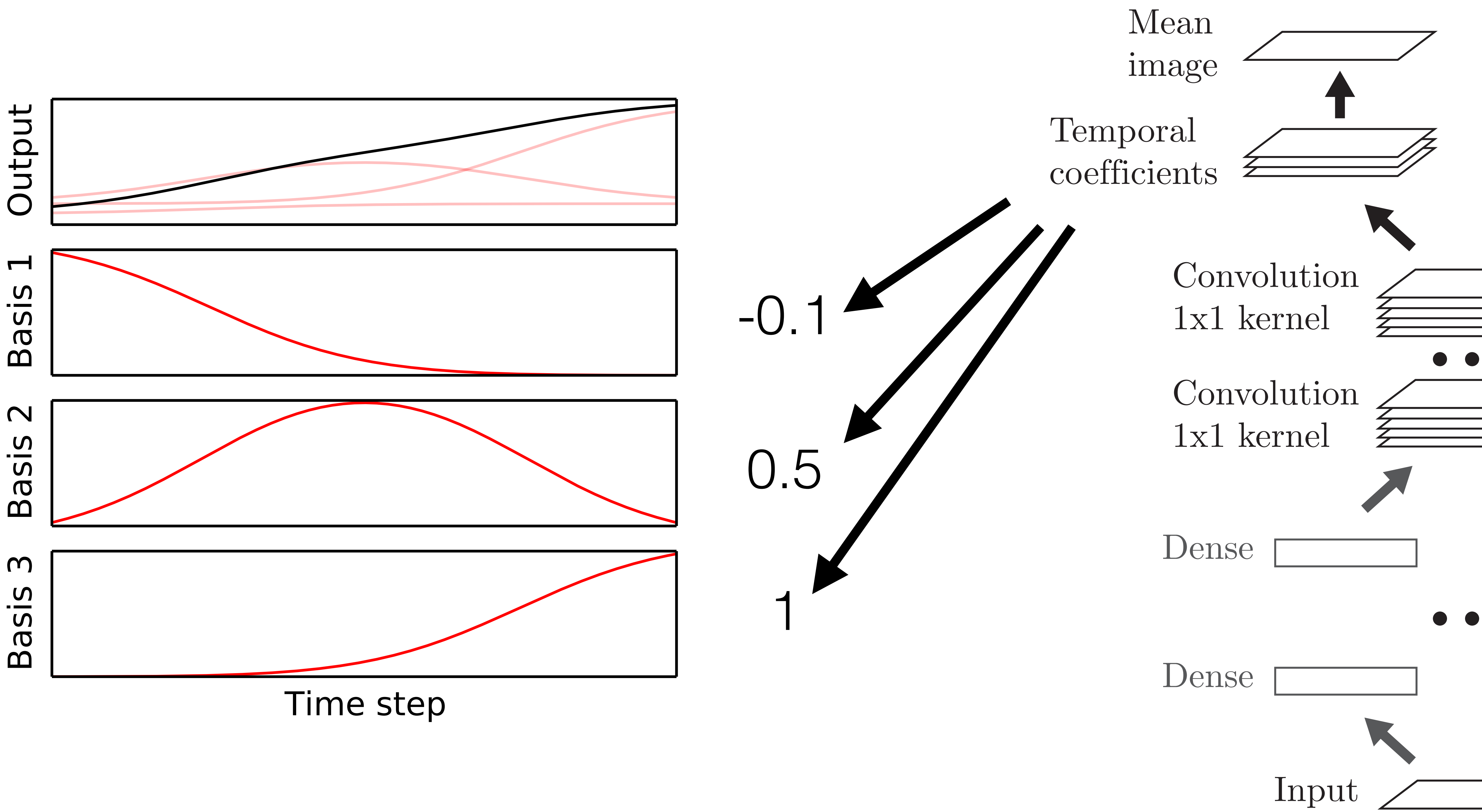
Time Dependence using Temporal Basis



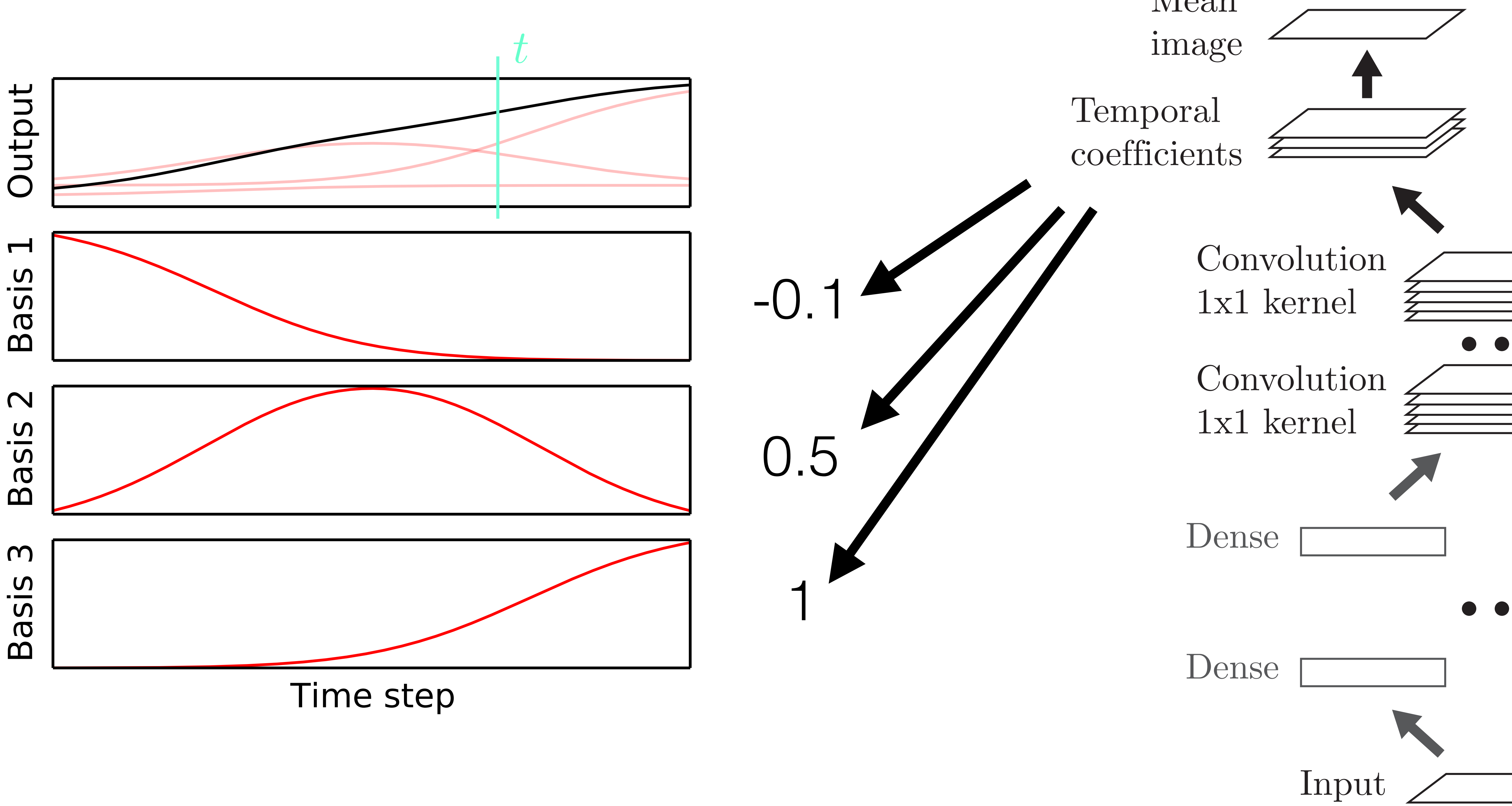
Time Dependence using Temporal Basis



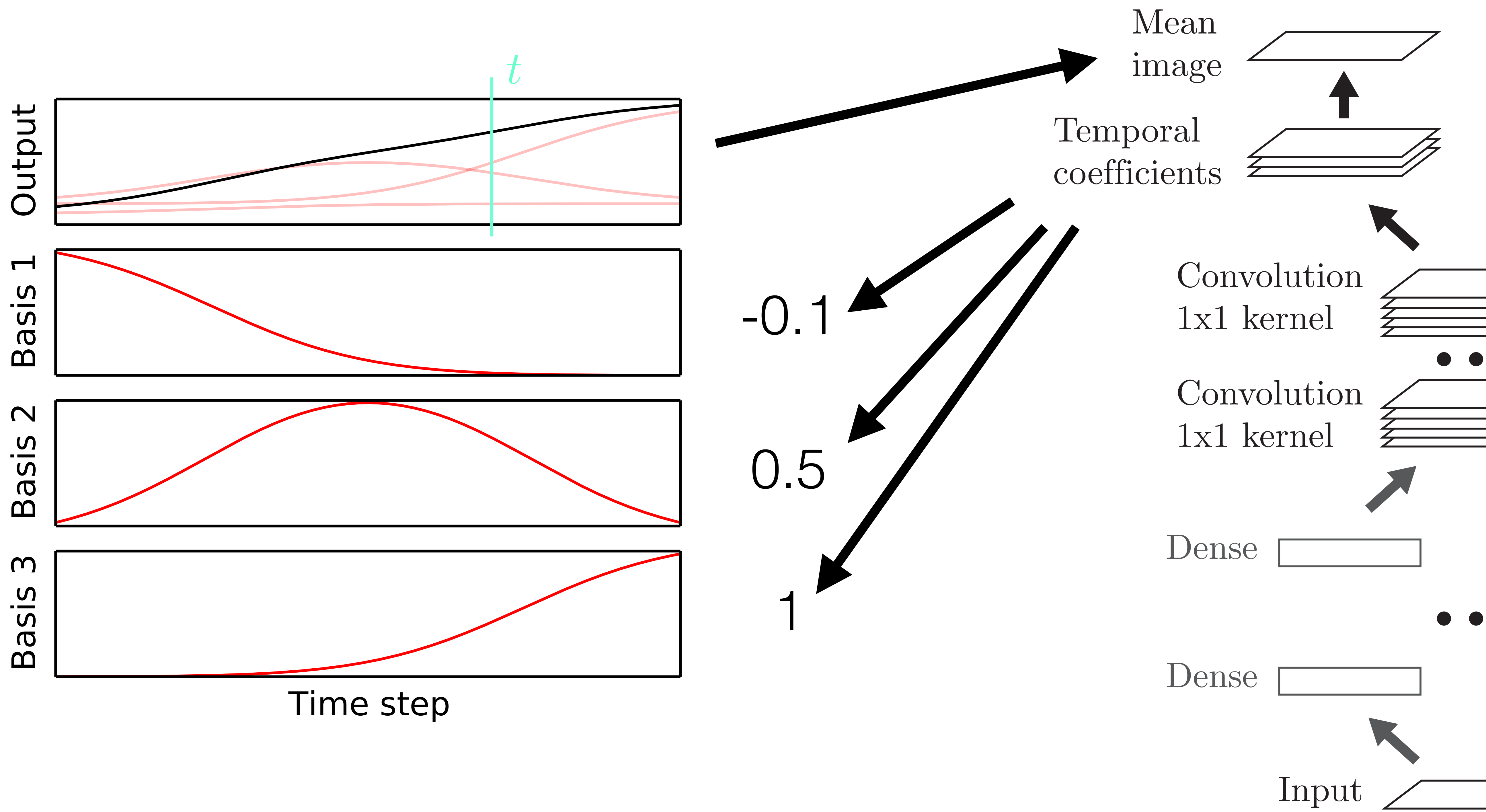
Time Dependence using Temporal Basis



Time Dependence using Temporal Basis



Time Dependence using Temporal Basis



Outline

- Motivation: The promise of deep unsupervised learning
- Physical intuition: Diffusion processes and time reversal
- Diffusion probabilistic model: Derivation and experimental results
  - Algorithm
  - Deep convolutional network: Universal function approximator
  - **Multiplying distributions: Inputation, denoising, computing posteriors**

Multiplying Distributions is Straightforward

Interested in $\tilde{p}\left(\mathbf{x}^{(0)}\right) \propto p\left(\mathbf{x}^{(0)}\right) r\left(\mathbf{x}^{(0)}\right)$

- Required to compute posterior distributions
  - Missing data (inpainting)
  - Corrupted data (denoising)

Multiplying Distributions is Straightforward

Interested in $\tilde{p}\left(\mathbf{x}^{(0)}\right) \propto p\left(\mathbf{x}^{(0)}\right) r\left(\mathbf{x}^{(0)}\right)$

- Required to compute posterior distributions
  - Missing data (inpainting)
  - Corrupted data (denoising)
- Difficult and expensive using competing techniques
  - e.g. variational autoencoders, GSNs, NADEs, most graphical models

Multiplying Distributions is Straightforward

Interested in $\tilde{p}\left(\mathbf{x}^{(0)}\right) \propto p\left(\mathbf{x}^{(0)}\right) r\left(\mathbf{x}^{(0)}\right)$

Multiplying Distributions is Straightforward

Interested in $\tilde{p}\left(\mathbf{x}^{(0)}\right) \propto p\left(\mathbf{x}^{(0)}\right) r\left(\mathbf{x}^{(0)}\right)$

Multiplying Distributions is Straightforward

Interested in $\tilde{p}\left(\mathbf{x}^{(0)}\right) \propto p\left(\mathbf{x}^{(0)}\right) \underline{r\left(\mathbf{x}^{(0)}\right)}$

Acts as small perturbation to diffusion process

Multiplying Distributions is Straightforward

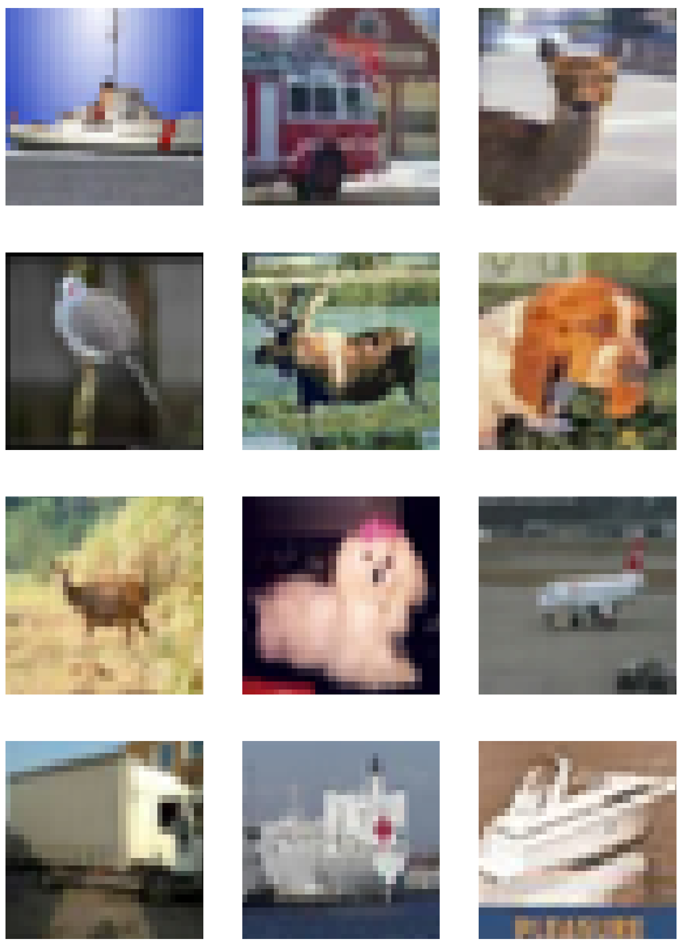
Interested in $\tilde{p}\left(\mathbf{x}^{(0)}\right) \propto p\left(\mathbf{x}^{(0)}\right) \underline{r\left(\mathbf{x}^{(0)}\right)}$

Acts as small perturbation to diffusion process

$$p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}^{(t-1)}; f_{\mu}\left(\mathbf{x}^{(t)}, t\right), f_{\Sigma}\left(\mathbf{x}^{(t)}, t\right)\right)$$

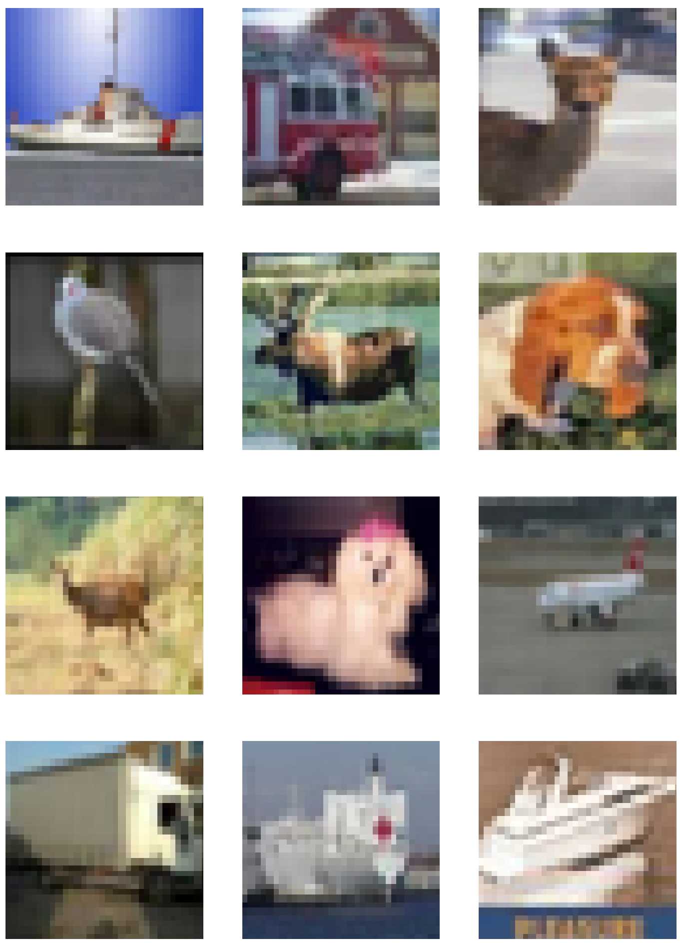
$$\tilde{p}\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right) \approx \mathcal{N}\left(x^{(t-1)}; \mathbf{f}_{\mu}\left(\mathbf{x}^{(t)}, t\right) + \mathbf{f}_{\Sigma}\left(\mathbf{x}^{(t)}, t\right) \frac{\partial \log r\left(\mathbf{x}^{(t-1)\prime}\right)}{\partial \mathbf{x}^{(t-1)\prime}}\Bigg|_{\mathbf{x}^{(t-1)\prime}=f_{\mu}\left(\mathbf{x}^{(t)}, t\right)}, \mathbf{f}_{\Sigma}\left(\mathbf{x}^{(t)}, t\right)\right)$$

Image Denoising by Sampling from Posterior

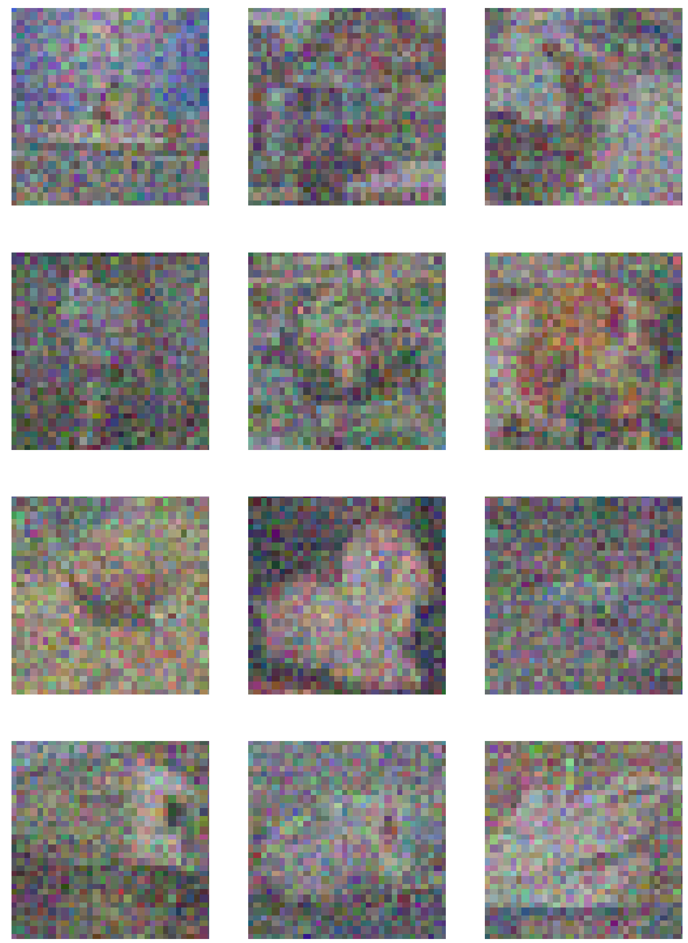


Holdout Data

Image Denoising by Sampling from Posterior

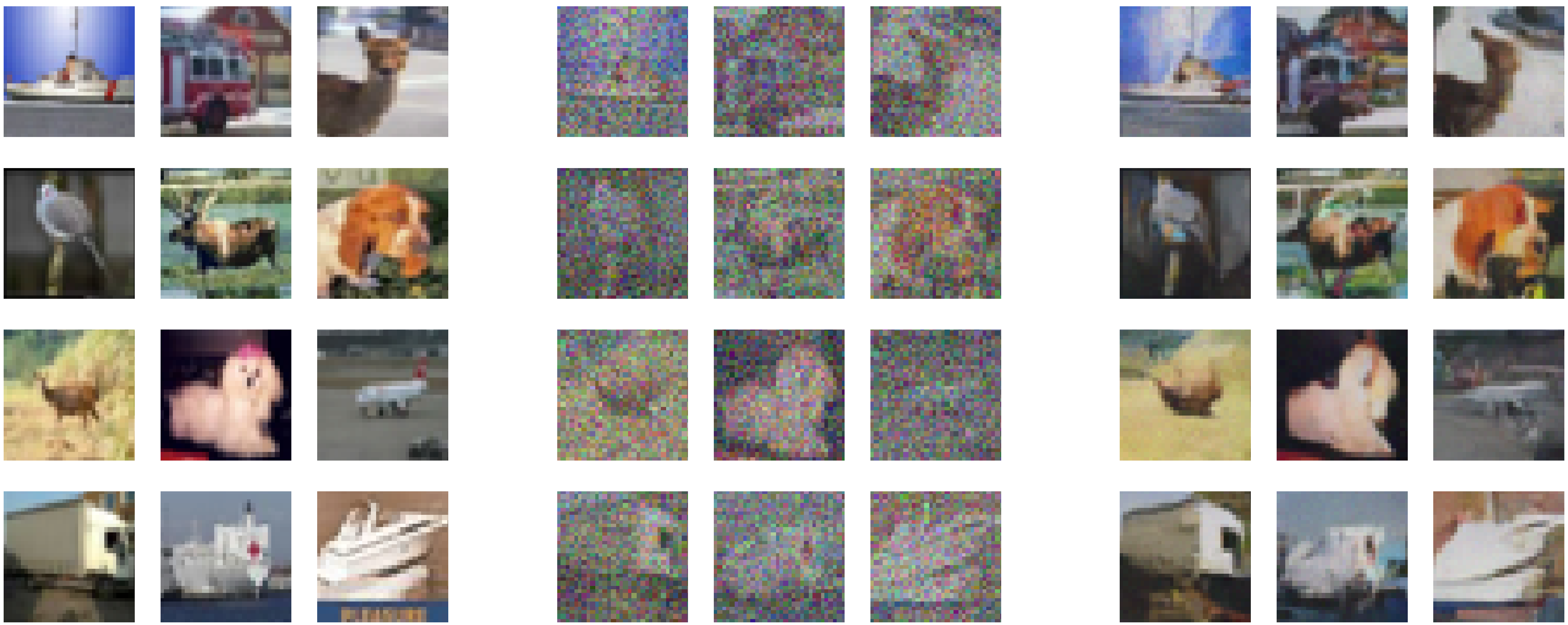


Holdout Data



Corrupted
(SNR = 1)

Image Denoising by Sampling from Posterior



Holdout Data

Corrupted (SNR = 1)

Denoised

Image Inpainting by Sampling from Posterior

- Training data [Lazebnik *et al*, 2005]

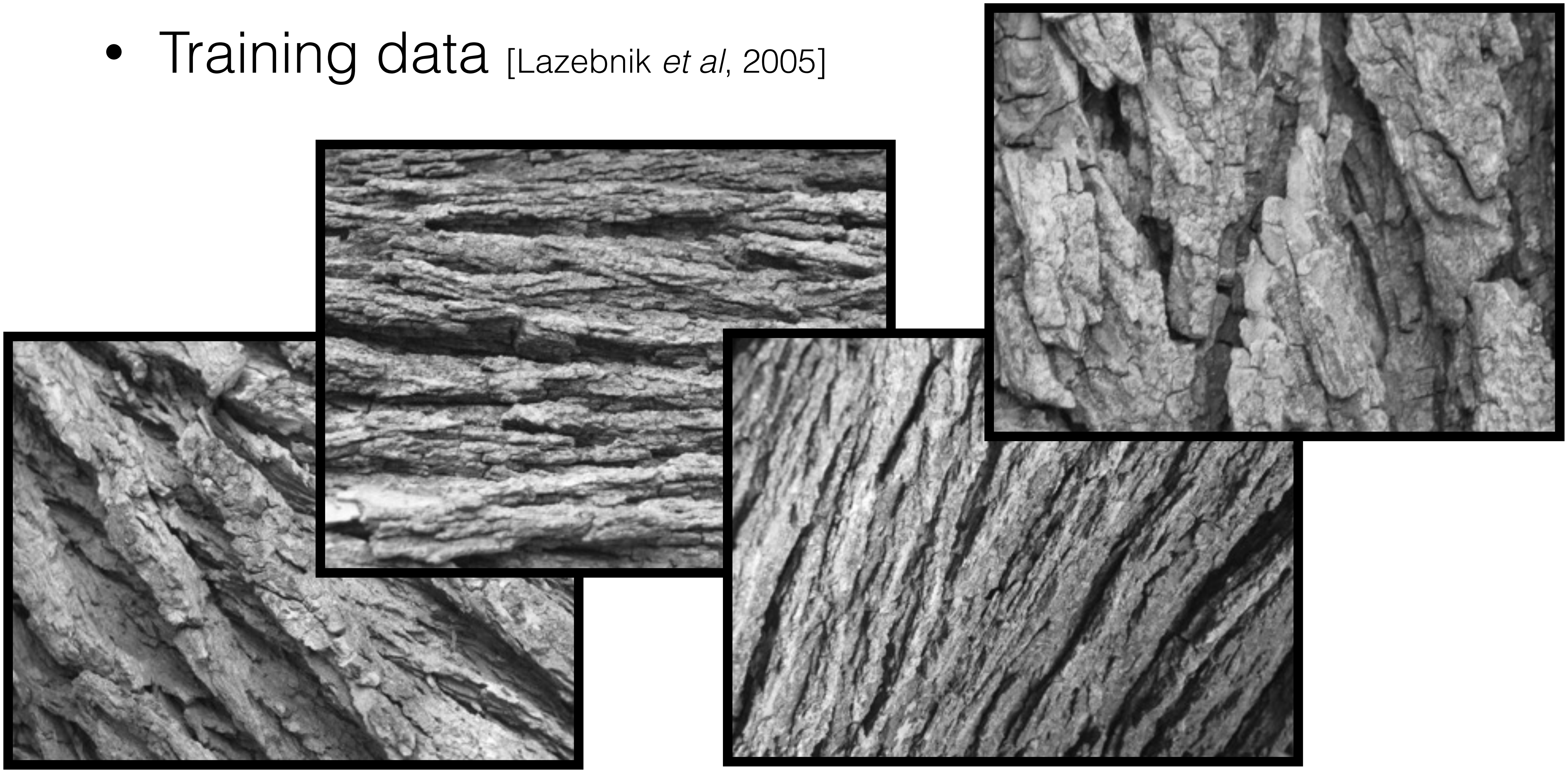
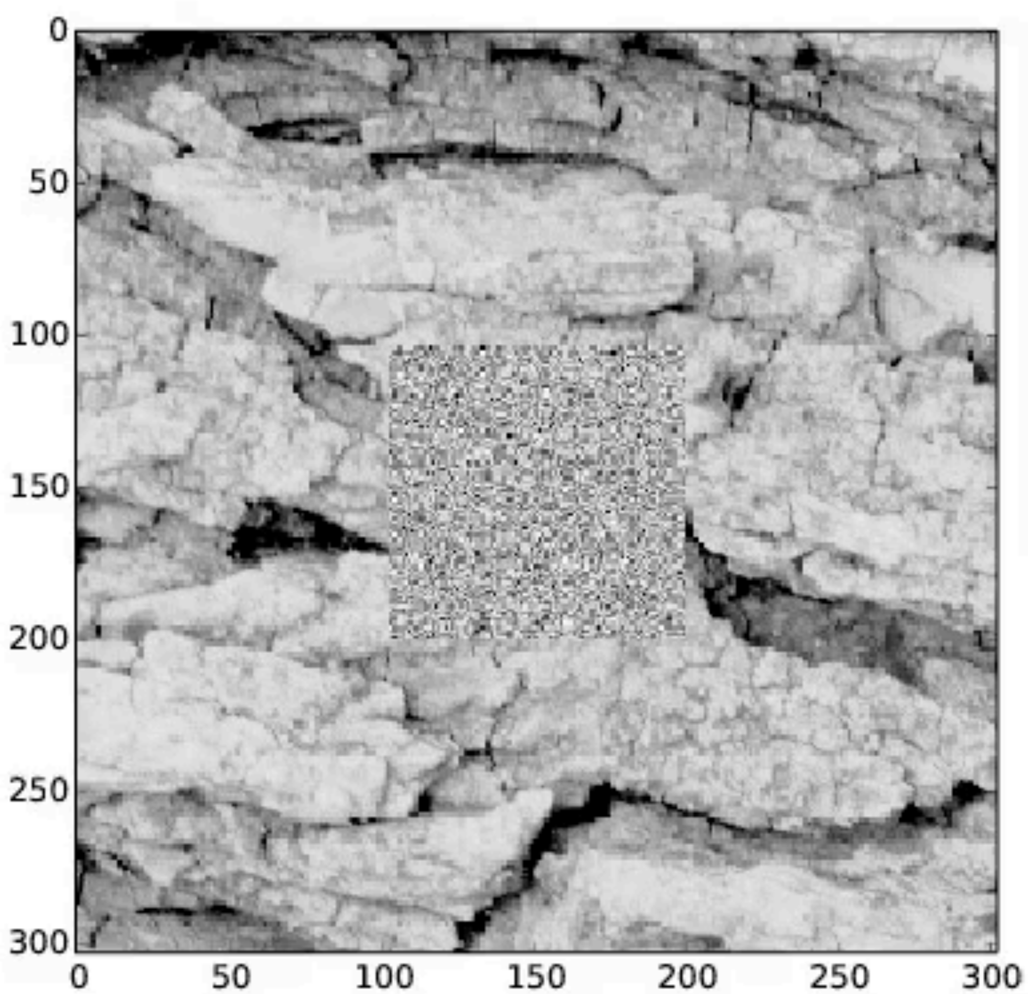
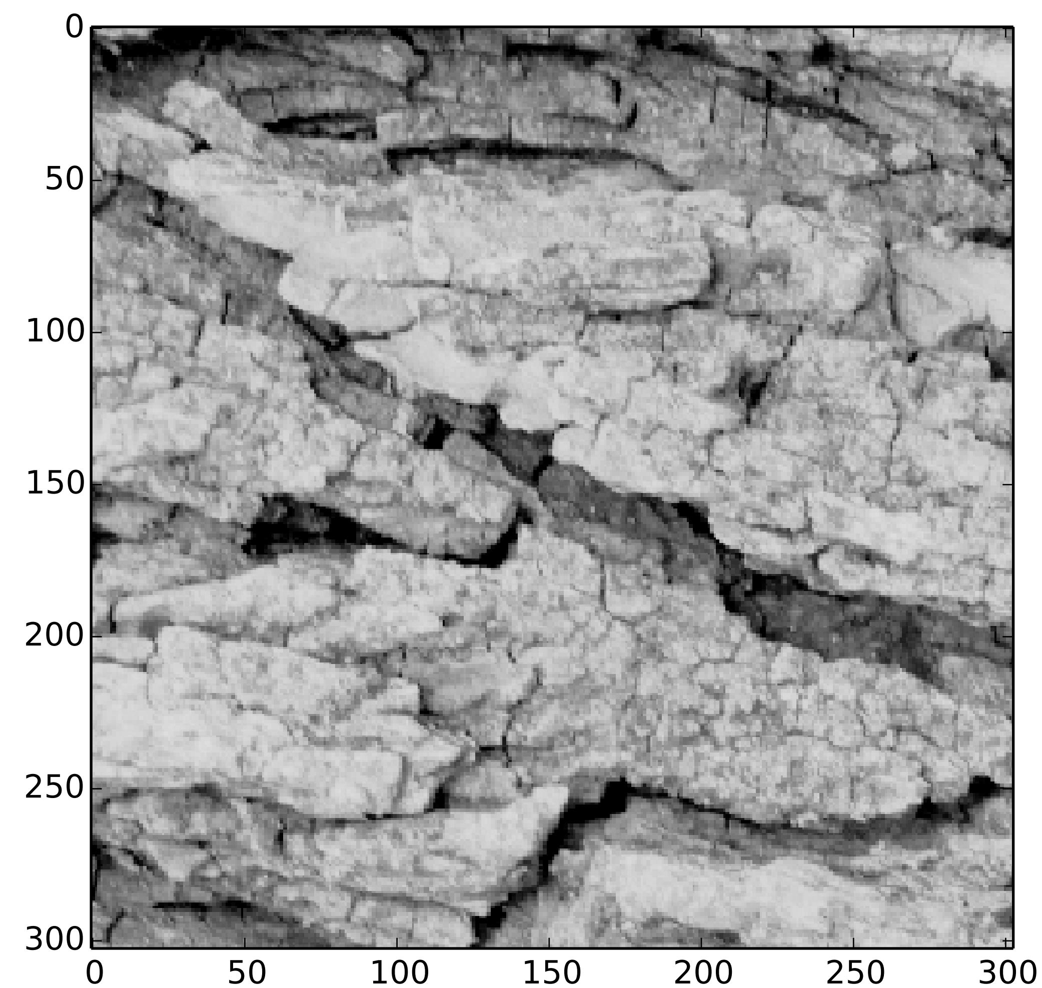


Image Inpainting by Sampling from Posterior

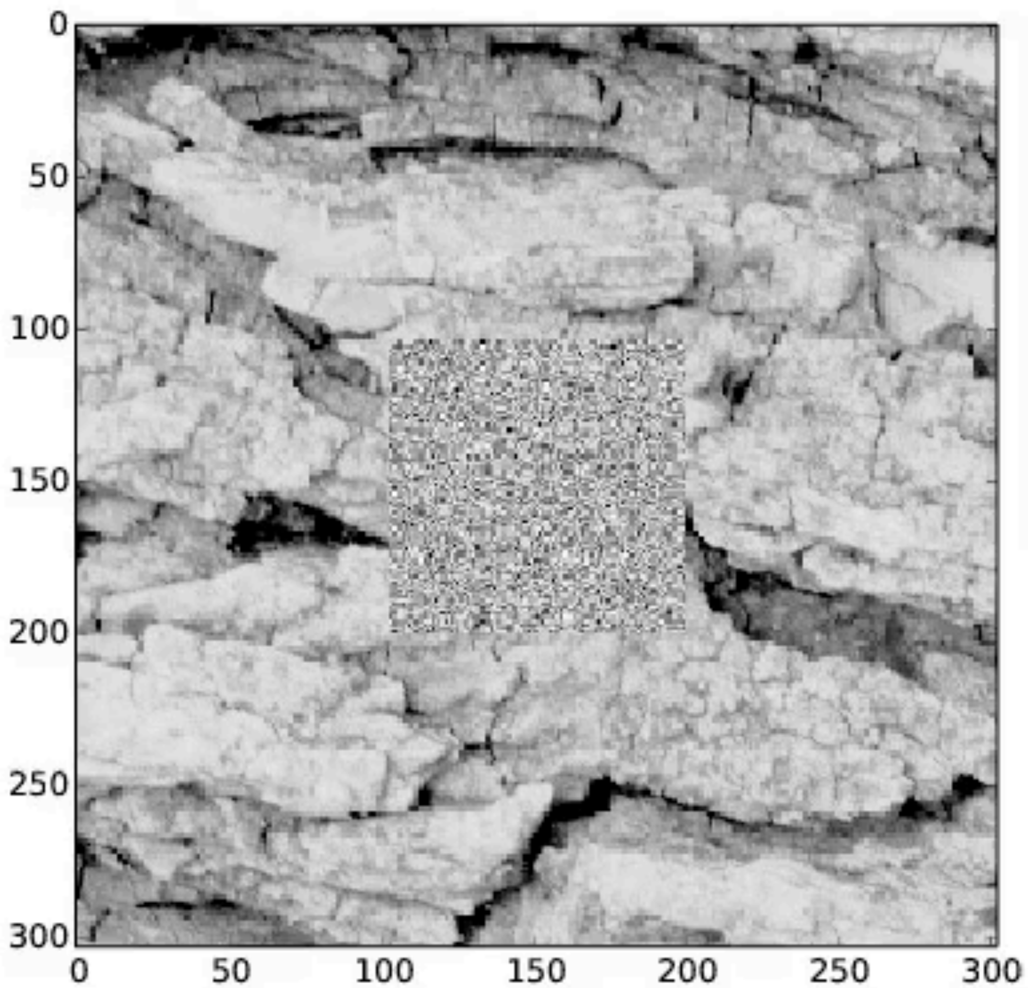


Inpainted image

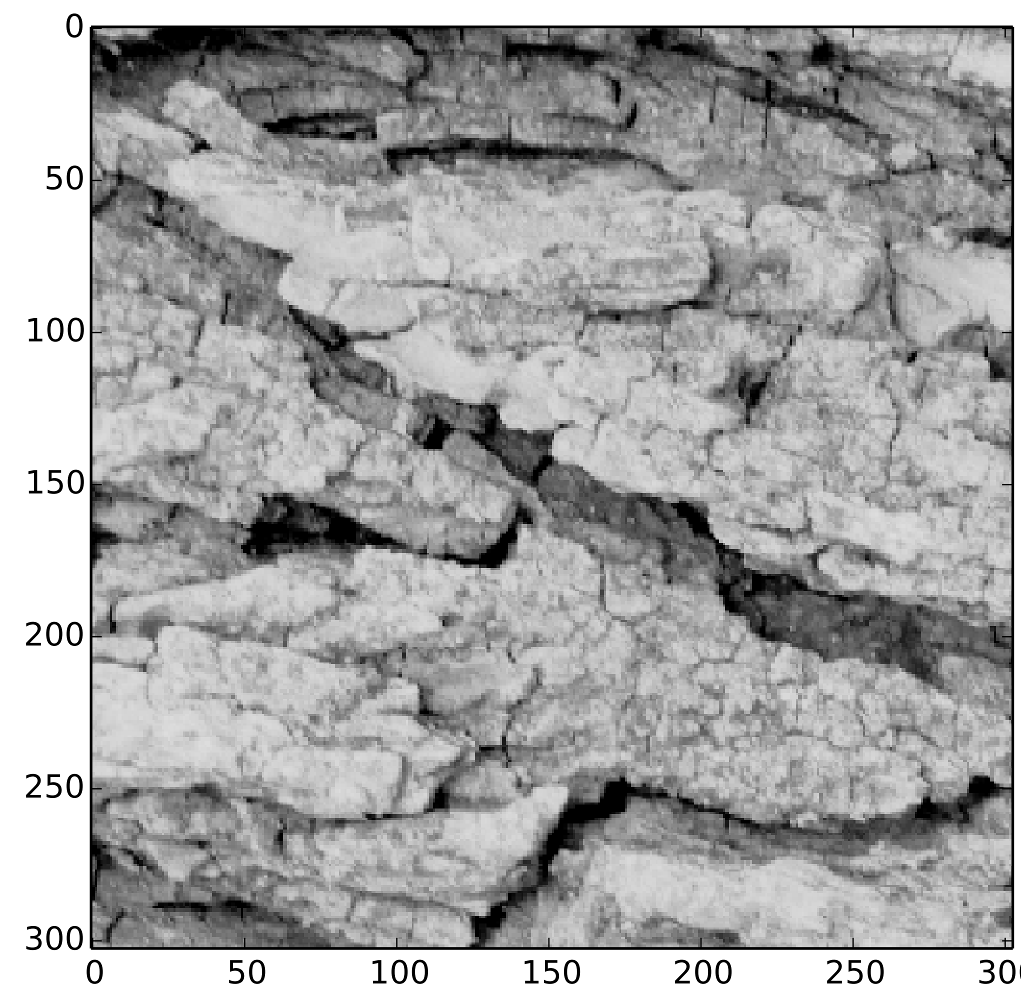


True image

Image Inpainting by Sampling from Posterior



Inpainted image



True image

Ongoing Work

Ongoing Work

- Binomial diffusion to neural spike trains

Ongoing Work

- Binomial diffusion to neural spike trains
- Full resolution color natural images

Ongoing Work

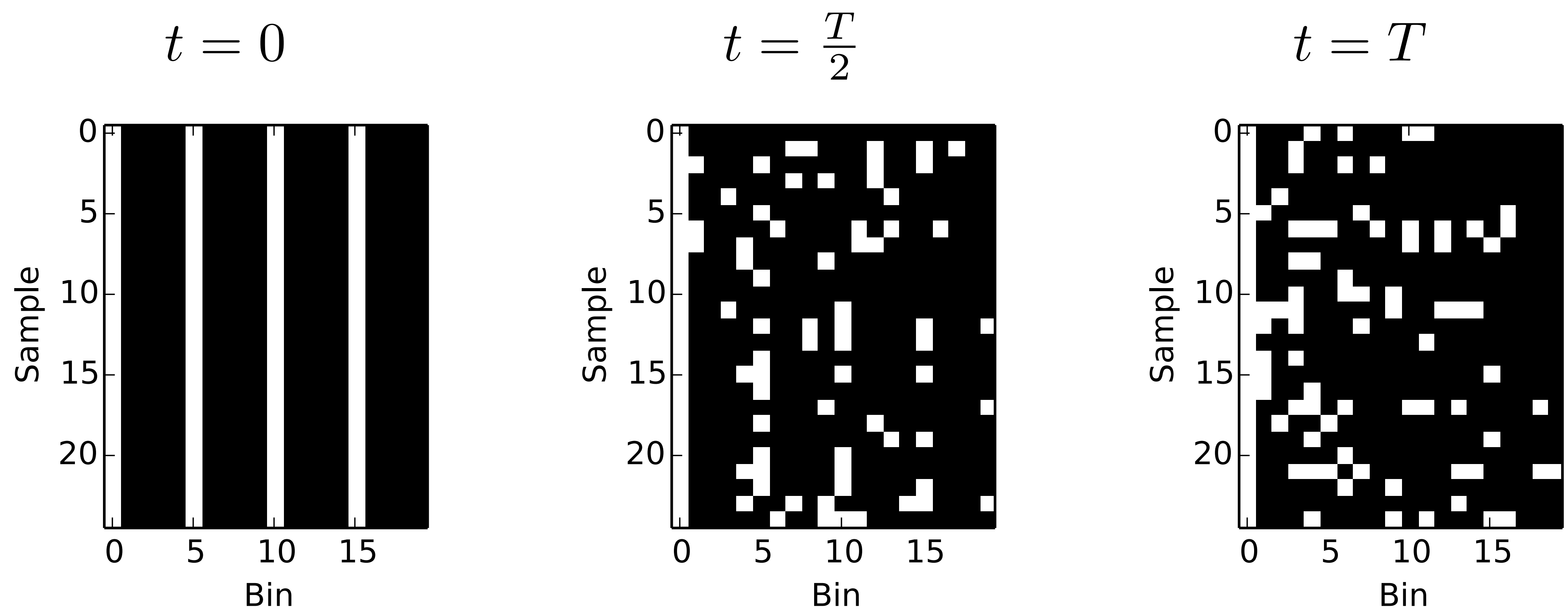
- Binomial diffusion to neural spike trains
- Full resolution color natural images
- Continuous time formulation

Ongoing Work

- Binomial diffusion to neural spike trains
- Full resolution color natural images
- Continuous time formulation
- Perturbation around energy based model

Toy Binary Sequence Learning

$$p\left(\mathbf{x}^{(0\cdots T)}\right)$$



Flexible and tractable method for deep unsupervised learning

Flexible and tractable method for deep unsupervised learning

- Flexible: Diffusion process for any (smooth) distribution

Flexible and tractable method for deep unsupervised learning

- Flexible: Diffusion process for any (smooth) distribution
    - Binary or continuous state space

Flexible and tractable method for deep unsupervised learning

- Flexible: Diffusion process for any (smooth) distribution
    - Binary or continuous state space
- Tractable: Training, exact sampling, inference, evaluation

Flexible and tractable method for deep unsupervised learning

- Flexible: Diffusion process for any (smooth) distribution
  - Binary or continuous state space
- Tractable: Training, exact sampling, inference, evaluation
- Deep networks with thousands of layers (/ time steps)

Flexible and tractable method for deep unsupervised learning

- Flexible: Diffusion process for any (smooth) distribution
  - Binary or continuous state space
- Tractable: Training, exact sampling, inference, evaluation
- Deep networks with thousands of layers (/ time steps)
- Easy to multiply distributions (e.g. for posterior)

Flexible and tractable method for deep unsupervised learning

- Flexible: Diffusion process for any (smooth) distribution
    - Binary or continuous state space
- Tractable: Training, exact sampling, inference, evaluation
- Deep networks with thousands of layers (/ time steps)
- Easy to multiply distributions (e.g. for posterior)
- Bounds on entropy production

Thanks!

Collaborators

Eric
Weiss

Niru
Maheswaranathan

Surya
Ganguli

Endless discussion

- The Ganguli Gang
- The Redwood Center for Theoretical Neuroscience

Setting Diffusion Rate

- Erase constant fraction of stimulus variance each step

$$\beta_t = \frac{1}{T - t + 1}$$

- Can also train β_t