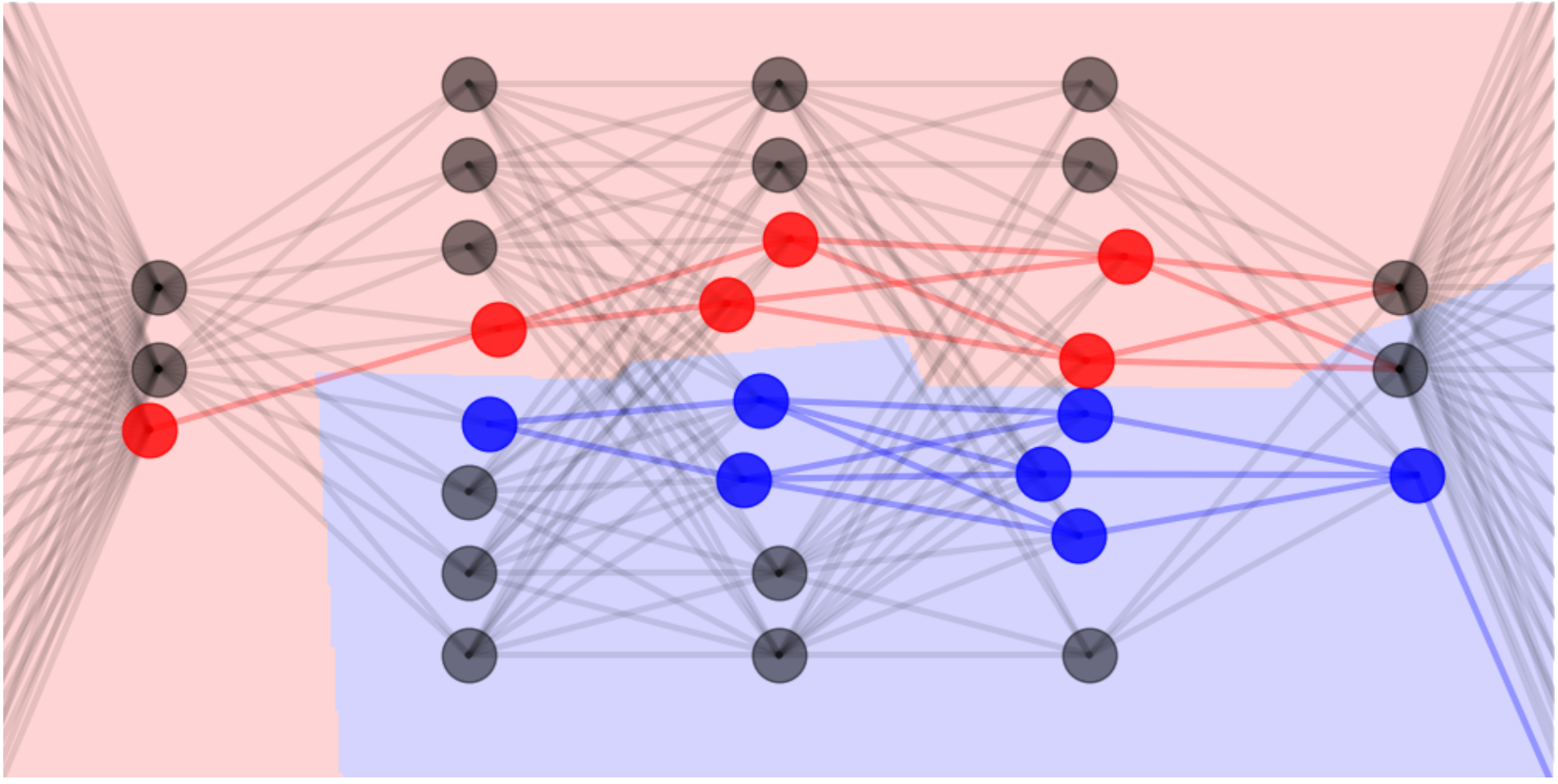
TU-Delft Machine and Deep Learning



Unsupervised

Lecture 8

Lecturer: Jan van Gemert

Unsupervised learning

DL Book: Chapters 14, 20

- Q: What is unsupervised learning?

Unsupervised learning

DL Book: Chapters 14, 20

- Q: What is unsupervised learning?
  A: No ground truth label.

- Q: Why is that good?

Unsupervised learning

DL Book: Chapters 14, 20

- Q: What is unsupervised learning?
  A: No ground truth label.
- Q: Why is that good?
  A: Labels are often the most expensive to obtain
- Q: What is it good for?

Unsupervised learning

DL Book: Chapters 14, 20

- Q: What is unsupervised learning?
  A: No ground truth label.

- Q: Why is that good?
  A: Labels are often the most expensive to obtain

- Q: What is it good for?
  A: Learn compression to store large datasets, pre-training for feature learning, density estimation, generating new data samples.

Unsupervised learning

DL Book: Chapters 14, 20

- Q: What is unsupervised learning?
  A: No ground truth label.

- Q: Why is that good?
  A: Labels are often the most expensive to obtain

- Q: What is it good for?
  A: Learn compression to store large datasets, pre-training for feature learning, density estimation, generating new data samples.

Topics: Auto encoders; generative models (VAE, GANs, diffusion); evaluation.

Autoencoder

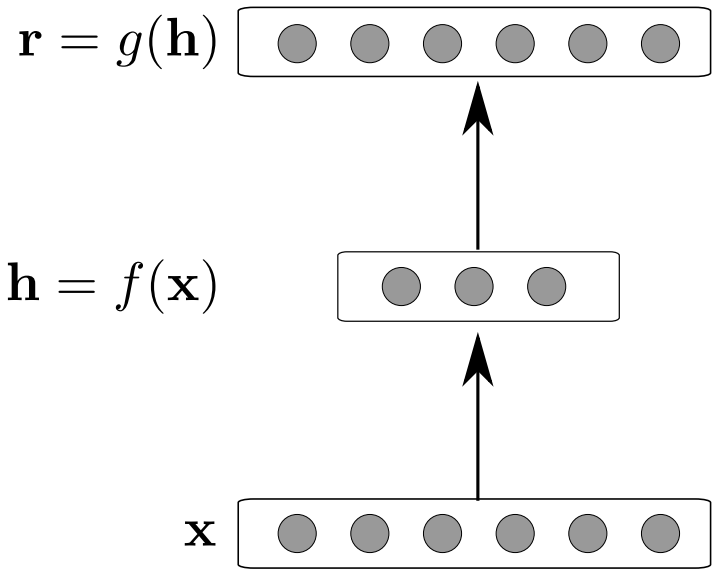
Book: Chapter 14; also: `https://www.jeremyjordan.me/autoencoders/`

Feed forward network to reproduce its input at the output layer; $g(f(\mathbf{x})) = \mathbf{x}$

Autoencoder

Book: Chapter 14; also: `https://www.jeremyjordan.me/autoencoders/`

Feed forward network to reproduce its input at the output layer; $g(f(\mathbf{x})) = \mathbf{x}$



Decoder:

$$\begin{aligned} \mathbf{r} &= g(\mathbf{h}) \\ &= \sigma(c + W_r \mathbf{h}) \end{aligned}$$

Encoder:

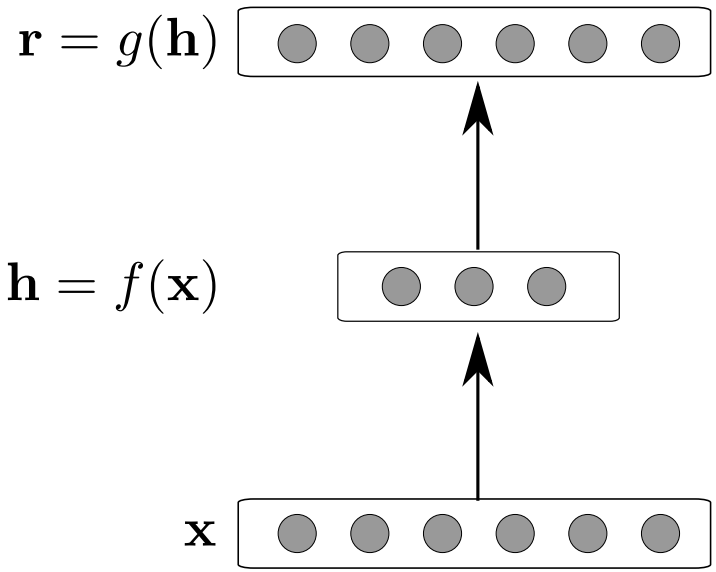
$$\begin{aligned} \mathbf{h} &= f(\mathbf{x}) \\ &= \sigma(b + W_h \mathbf{x}) \end{aligned}$$

Q: Explain what you see.

Autoencoder

Book: Chapter 14; also: `https://www.jeremyjordan.me/autoencoders/`

Feed forward network to reproduce its input at the output layer; $g(f(\mathbf{x})) = \mathbf{x}$



Decoder:

$$\begin{aligned} \mathbf{r} &= g(\mathbf{h}) \\ &= \sigma(c + W_r \mathbf{h}) \end{aligned}$$

Encoder:

$$\begin{aligned} \mathbf{h} &= f(\mathbf{x}) \\ &= \sigma(b + W_h \mathbf{x}) \end{aligned}$$

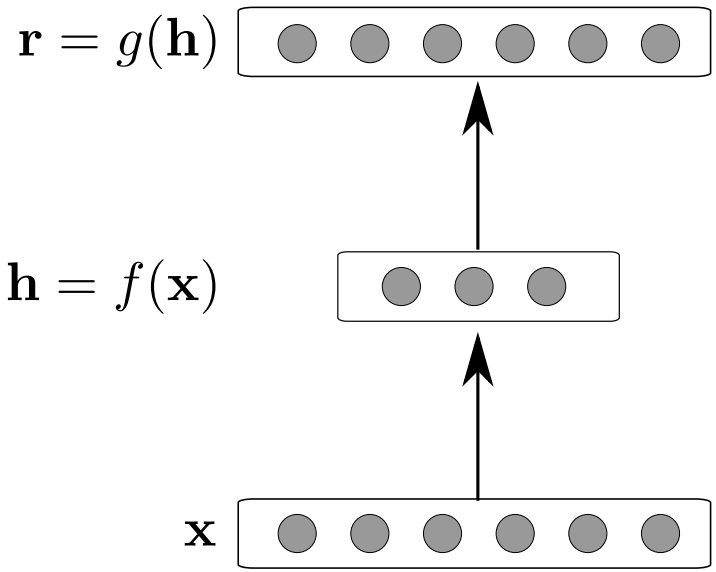
Q: Explain what you see.

Q: What loss function would you minimize to train $g(f(x)) = x$?

Autoencoder

Book: Chapter 14; also: `https://www.jeremyjordan.me/autoencoders/`

Feed forward network to reproduce its input at the output layer; $g(f(\mathbf{x})) = \mathbf{x}$



Decoder:

$$\begin{aligned} \mathbf{r} &= g(\mathbf{h}) \\ &= \sigma(c + W_r \mathbf{h}) \end{aligned}$$

Encoder:

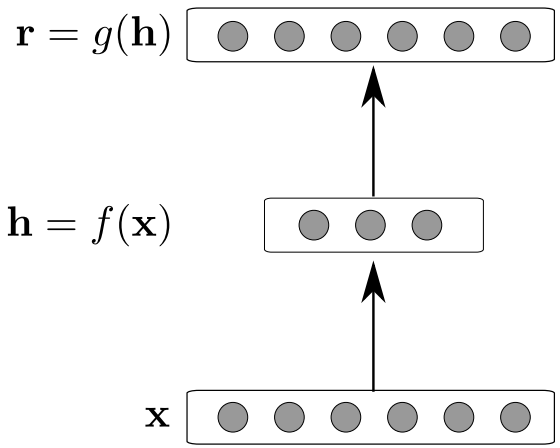
$$\begin{aligned} \mathbf{h} &= f(\mathbf{x}) \\ &= \sigma(b + W_h \mathbf{x}) \end{aligned}$$

Q: Explain what you see.

Q: What loss function would you minimize to train $g(f(x)) = x$?
A: $\sum_i \frac{1}{2}(g(f(x_i)) - x_i)^2$ (if x is continue)

Size of the hidden (bottleneck) layer

Book: Chapter 14

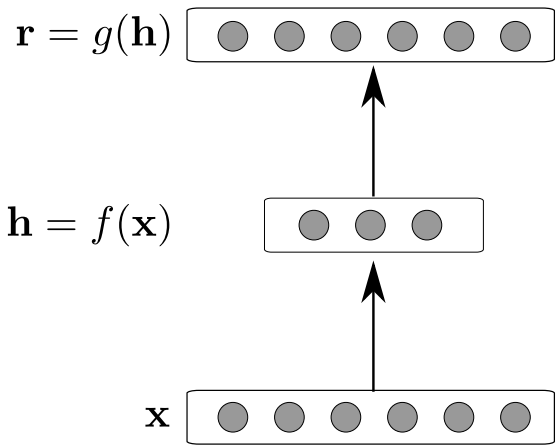


$h < x$, **Undercomplete** hidden layer:

- "Compress" the input
- Compresses well for training samples

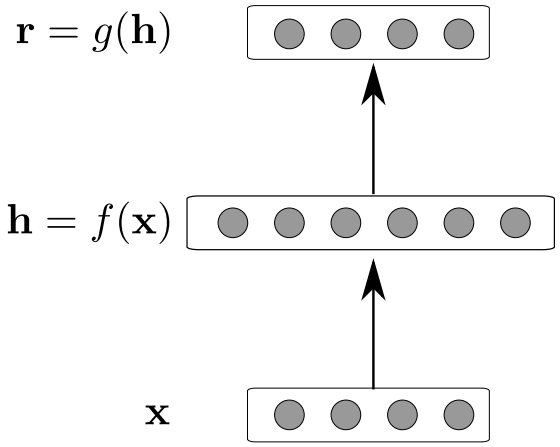
Size of the hidden (bottleneck) layer

Book: Chapter 14



$h < x$, **Undercomplete** hidden layer:

- "Compress" the input
- Compresses well for training samples

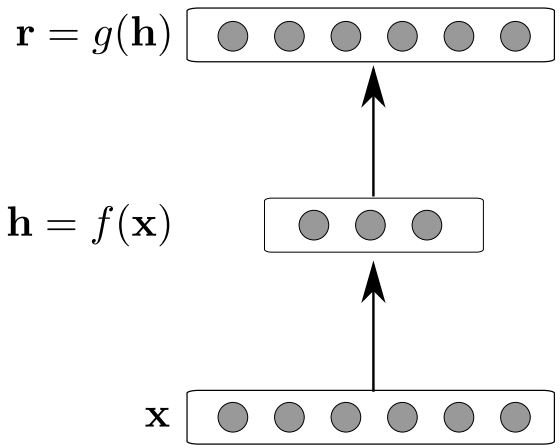


$h > x$, **Overcomplete** hidden layer

- No compression needed (could just copy input)
- Useful for representation learning

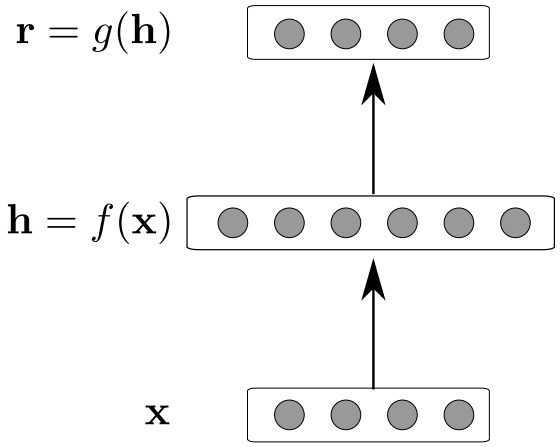
Size of the hidden (bottleneck) layer

Book: Chapter 14



$h < x$, **Undercomplete** hidden layer:

- "Compress" the input
- Compresses well for training samples



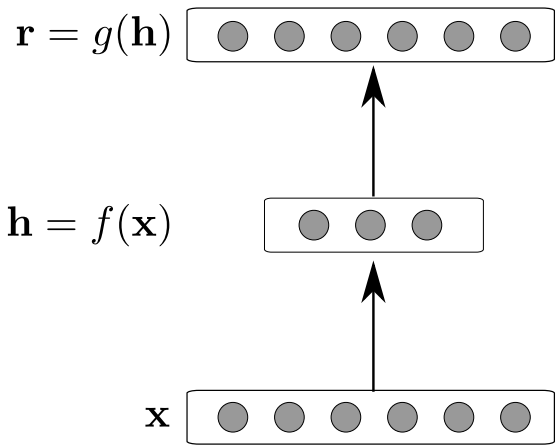
$h > x$, **Overcomplete** hidden layer

- No compression needed (could just copy input)
- Useful for representation learning

  Q: What lecture topic could prevent copying the input?

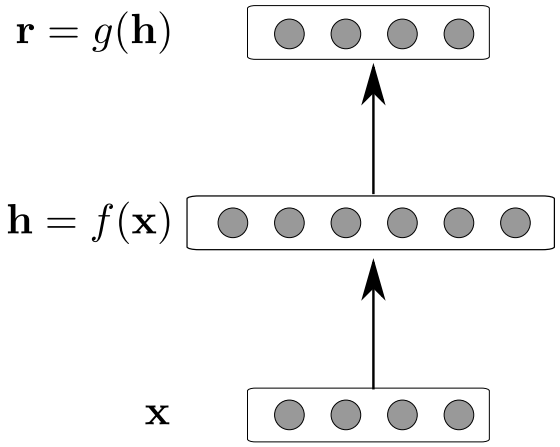
Size of the hidden (bottleneck) layer

Book: Chapter 14



$h < x$, **Undercomplete** hidden layer:

- "Compress" the input
- Compresses well for training samples



$h > x$, **Overcomplete** hidden layer

- No compression needed (could just copy input)
- Useful for representation learning

  Q: What lecture topic could prevent copying the input?
  A: Regularization

Overcomplete: Denoising autoencoder

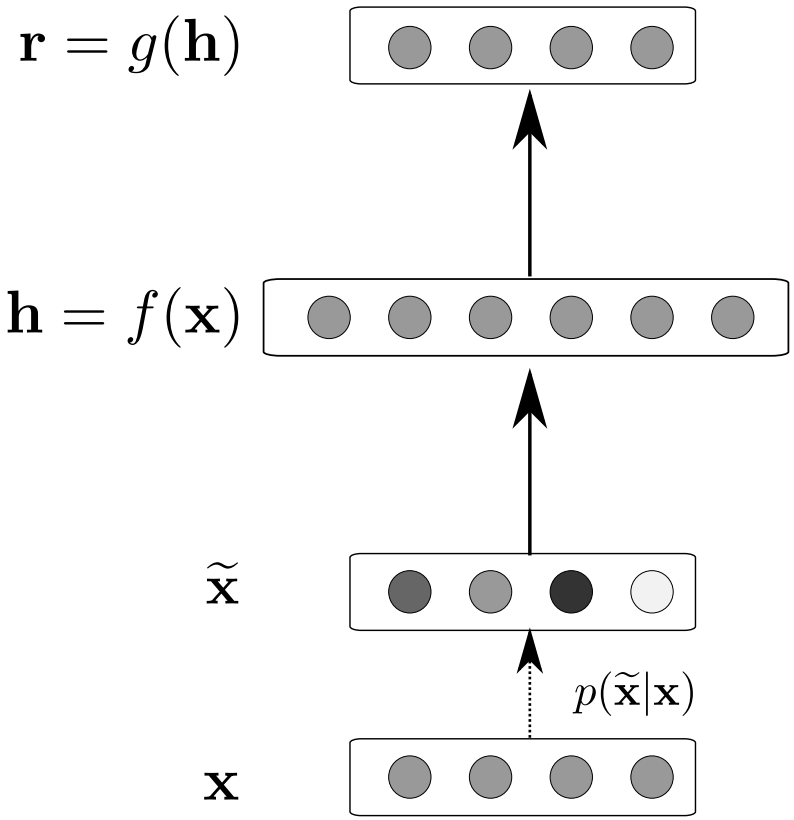
Book: 14.2.2, 14.5

Idea: Regularization by making reconstruction robust to noise.

Overcomplete: Denoising autoencoder

Book: 14.2.2, 14.5

Idea: Regularization by making reconstruction robust to noise.

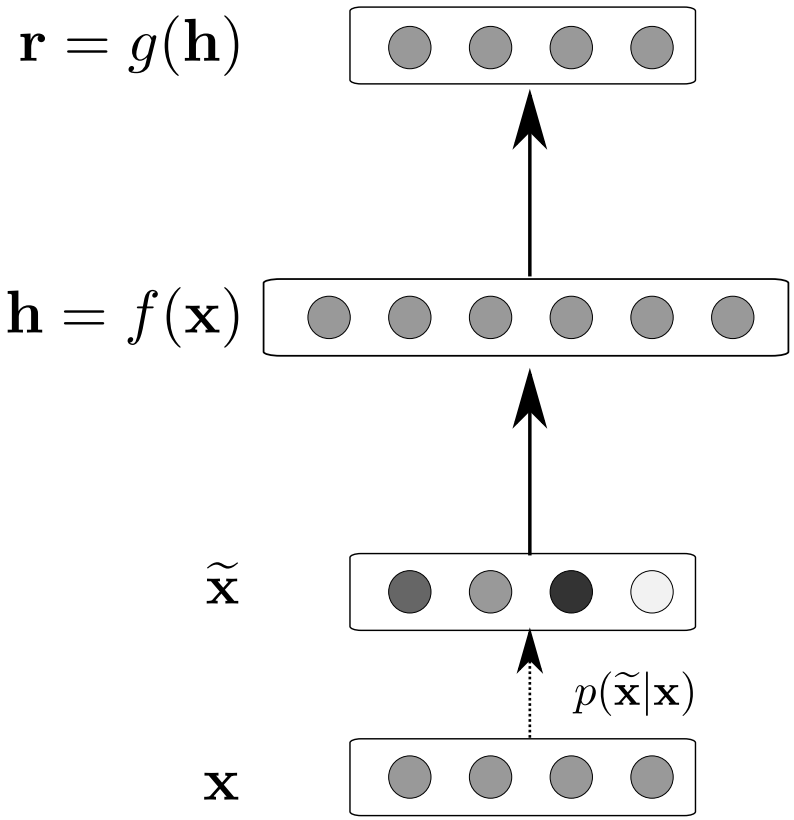


Generate noisy $\widetilde{\mathbf{x}}$ from a distribution $p(\widetilde{\mathbf{x}}|\mathbf{x})$

Overcomplete: Denoising autoencoder

Book: 14.2.2, 14.5

Idea: Regularization by making reconstruction robust to noise.



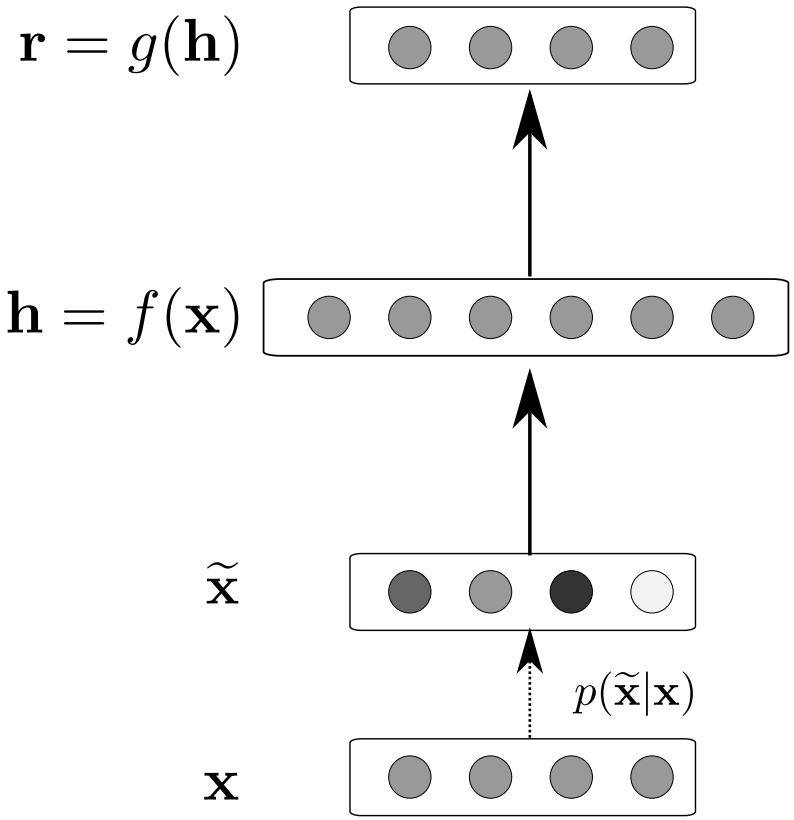
Generate noisy $\widetilde{\mathbf{x}}$ from a distribution $p(\widetilde{\mathbf{x}}|\mathbf{x})$

Q: How to make robust to noise?

Overcomplete: Denoising autoencoder

Book: 14.2.2, 14.5

Idea: Regularization by making reconstruction robust to noise.



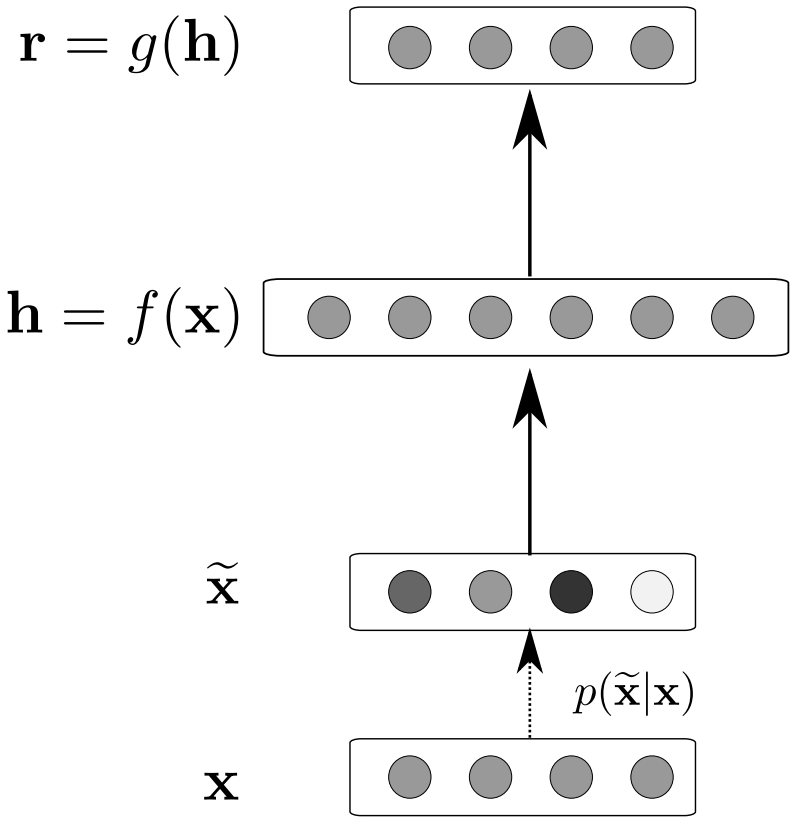
Generate noisy $\widetilde{\mathbf{x}}$ from a distribution $p(\widetilde{\mathbf{x}}|\mathbf{x})$

Q: How to make robust to noise?
A: Reconstruct the loss on $g(f(\widetilde{\mathbf{x}})) = \mathbf{x}$

Overcomplete: Denoising autoencoder

Book: 14.2.2, 14.5

Idea: Regularization by making reconstruction robust to noise.



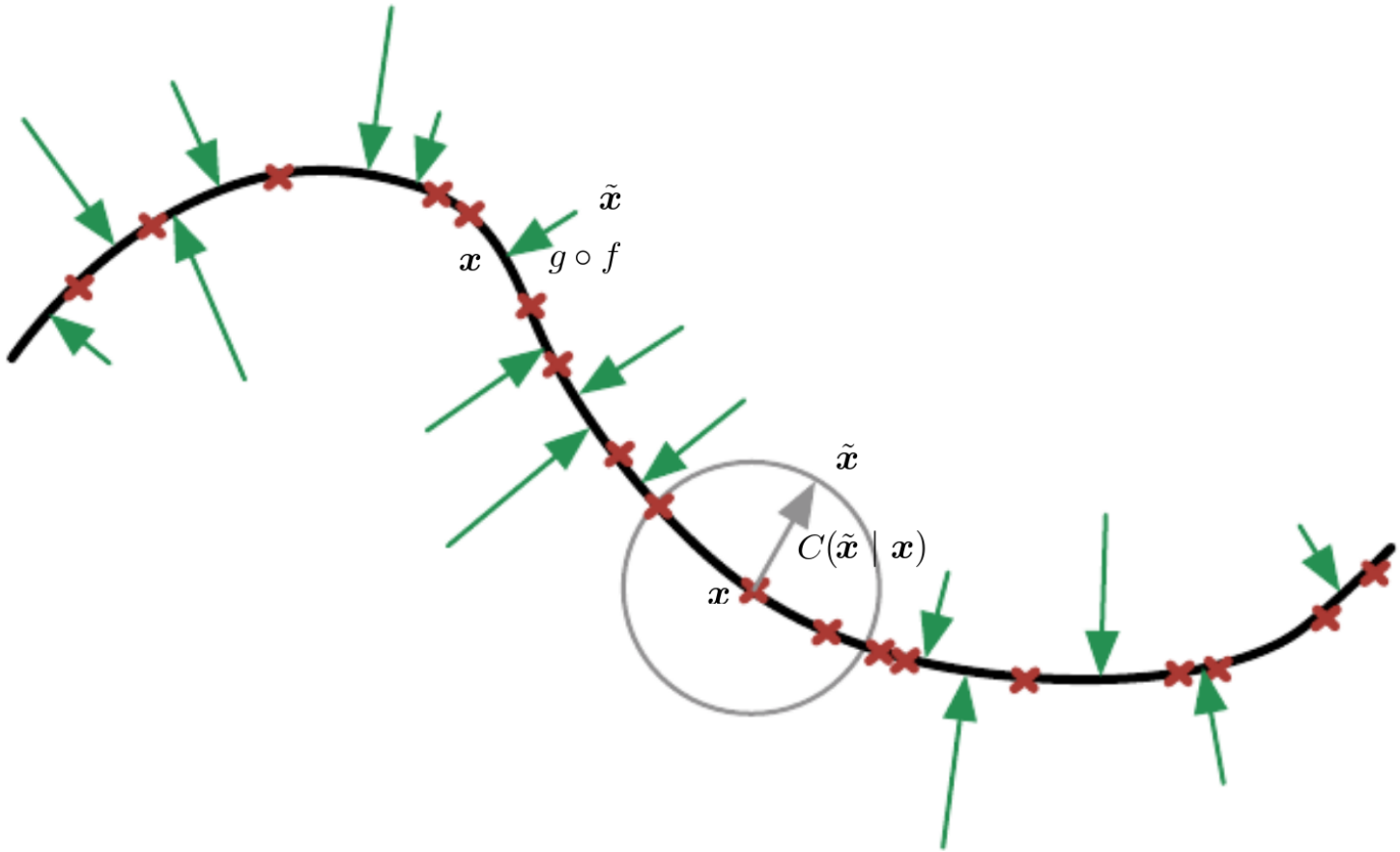
Generate noisy $\widetilde{\mathbf{x}}$ from a distribution $p(\widetilde{\mathbf{x}}|\mathbf{x})$

Q: How to make robust to noise?
A: Reconstruct the loss on $g(f(\widetilde{\mathbf{x}})) = \mathbf{x}$

Input a noisy sample $\widetilde{\mathbf{x}}$ and optimize to reconstruct the noise-free $\mathbf{x}$.

Denoising autoencoder

Fig 14.4 in book



True sample $\mathbf{x}$, noisy sample $\widetilde{\mathbf{x}}$, Noise process $C(\widetilde{\mathbf{x}}|\mathbf{x})$, wrt $p(\widetilde{\mathbf{x}}|\mathbf{x})$

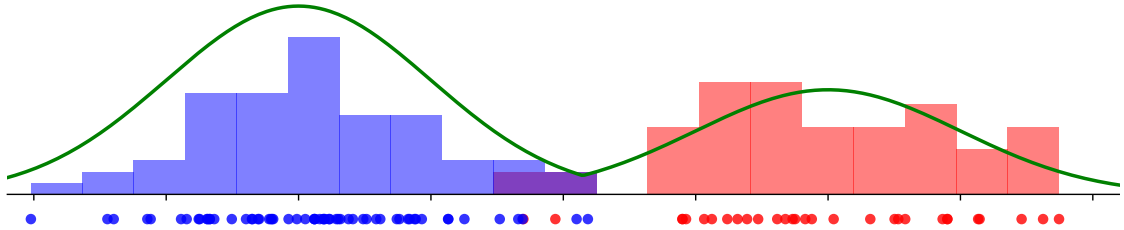
Questions?

Questions?

Let's talk about unsupervised generative modeling approaches (VAE, GANs, diffusion, etc.) for generating new data.

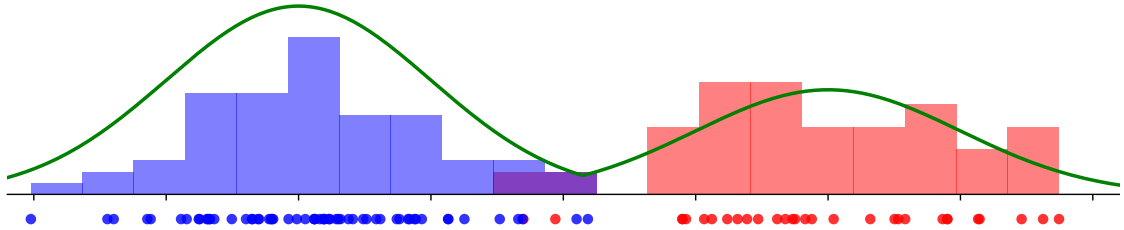
Generating new data

My phone images: empirical samples of the distribution of cats and dogs images.



Generating new data

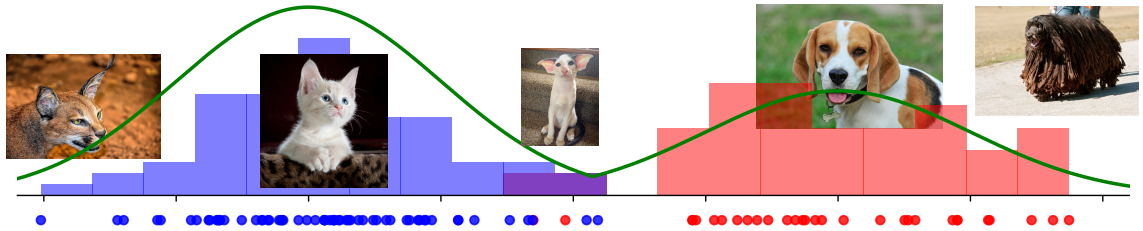
My phone images: empirical samples of the distribution of cats and dogs images.



Q: What are the blue/red dots?

Generating new data

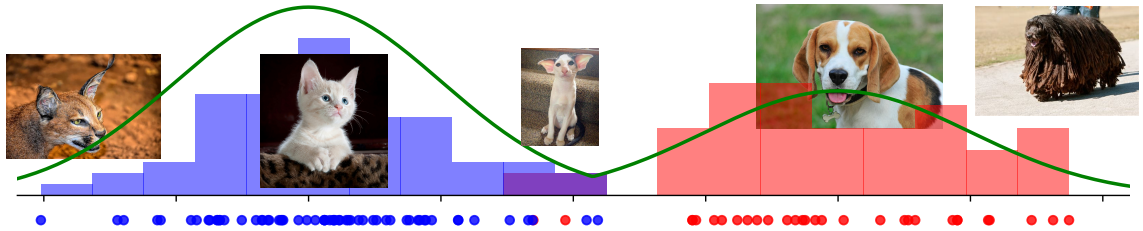
My phone images: empirical samples of the distribution of cats and dogs images.



Q: What are the blue/red dots? A: Cat/dog images (a distribution over pixels).

Generating new data

My phone images: empirical samples of the distribution of cats and dogs images.

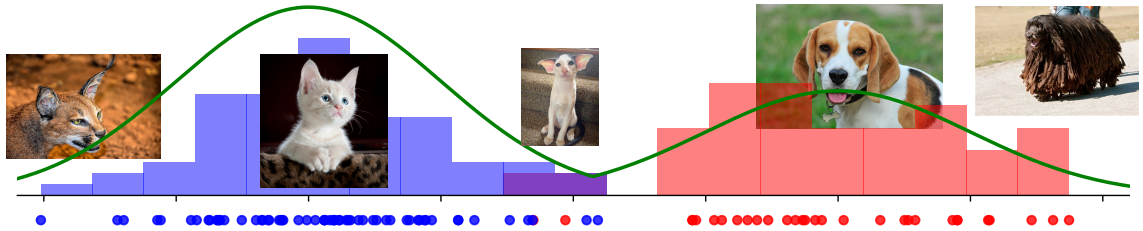


Q: What are the blue/red dots? A: Cat/dog images (a distribution over pixels).

Cannot directly sample from this distribution to generate new cat/dog images.

Generating new data

My phone images: empirical samples of the distribution of cats and dogs images.



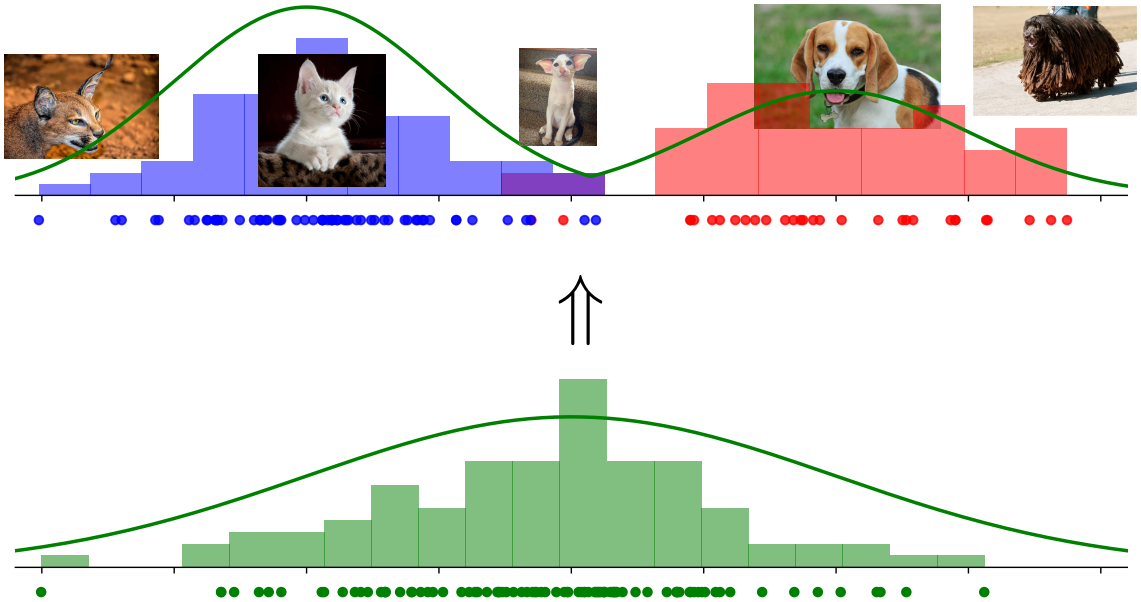
Q: What are the blue/red dots? A: Cat/dog images (a distribution over pixels).

Cannot directly sample from this distribution to generate new cat/dog images.

Key idea: Transform samples from a distribution where I *can* sample: $\mathcal{N}(0, I)$.

Generating new data

My phone images: empirical samples of the distribution of cats and dogs images.



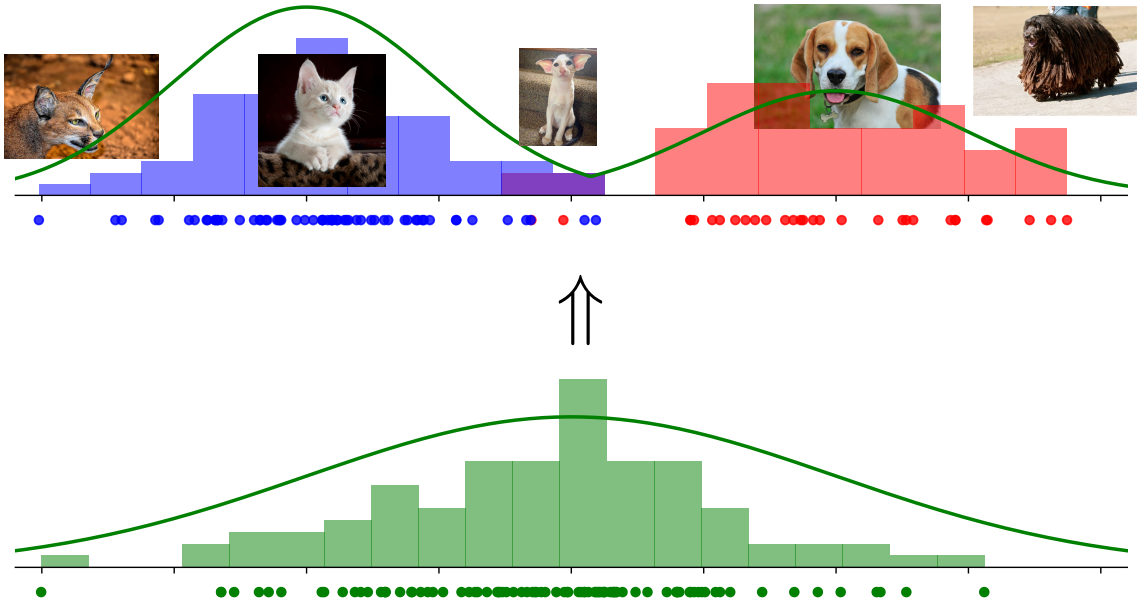
Q: What are the blue/red dots? A: Cat/dog images (a distribution over pixels).

Cannot directly sample from this distribution to generate new cat/dog images.

Key idea: Transform samples from a distribution where I *can* sample: $\mathcal{N}(0, I)$.

Generating new data

My phone images: empirical samples of the distribution of cats and dogs images.



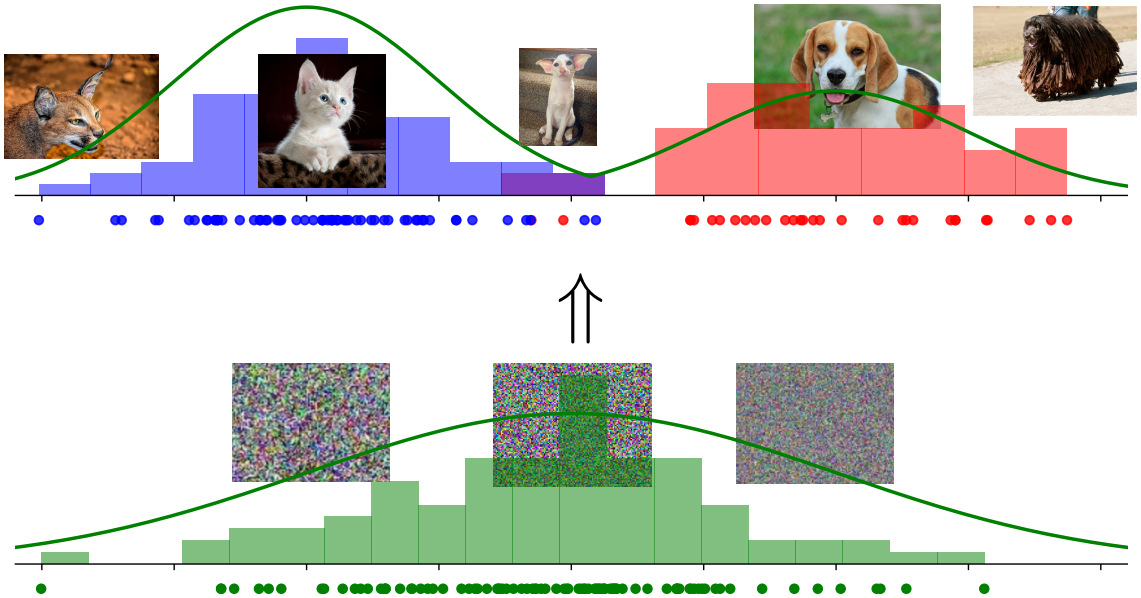
Q: What are the blue/red dots? A: Cat/dog images (a distribution over pixels).

Cannot directly sample from this distribution to generate new cat/dog images.

Key idea: Transform samples from a distribution where I *can* sample: $\mathcal{N}(0, I)$.
Q: What are the green dots?

Generating new data

My phone images: empirical samples of the distribution of cats and dogs images.



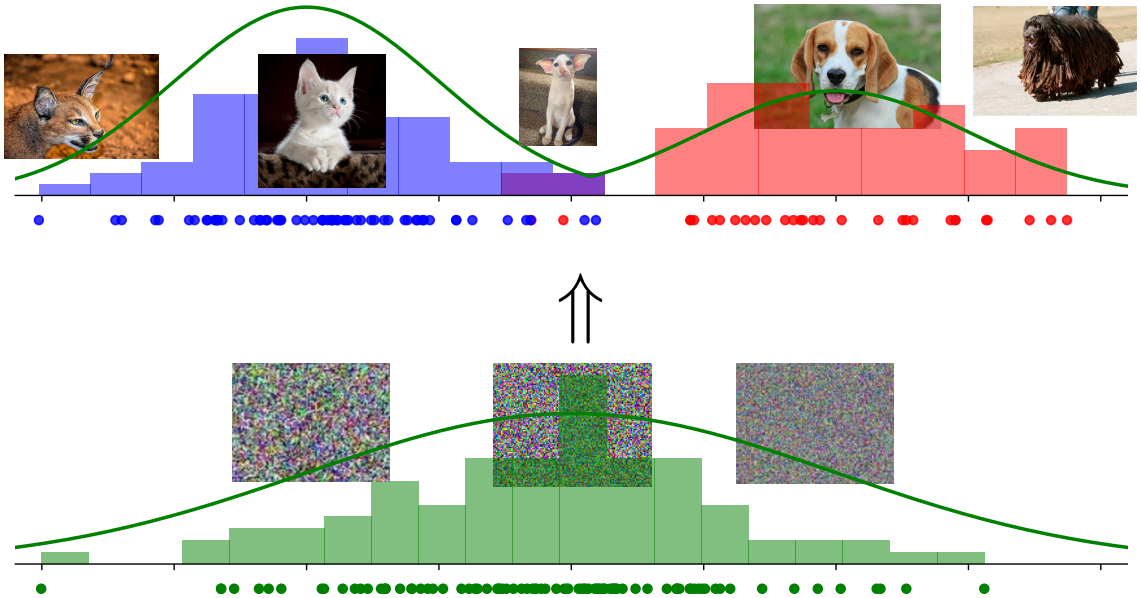
Q: What are the blue/red dots? A: Cat/dog images (a distribution over pixels).

Cannot directly sample from this distribution to generate new cat/dog images.

Key idea: Transform samples from a distribution where I *can* sample: $\mathcal{N}(0, I)$.
Q: What are the green dots? A: 'Noise images'; $\mathcal{N}(0, I)$.

Generating new data

My phone images: empirical samples of the distribution of cats and dogs images.



Q: What are the blue/red dots? A: Cat/dog images (a distribution over pixels).

Cannot directly sample from this distribution to generate new cat/dog images.

Key idea: Transform samples from a distribution where I *can* sample: $\mathcal{N}(0, I)$.
Q: What are the green dots? A: 'Noise images'; $\mathcal{N}(0, I)$.

Main idea for generative modeling approaches (VAE, GANs, diffusion, etc.)

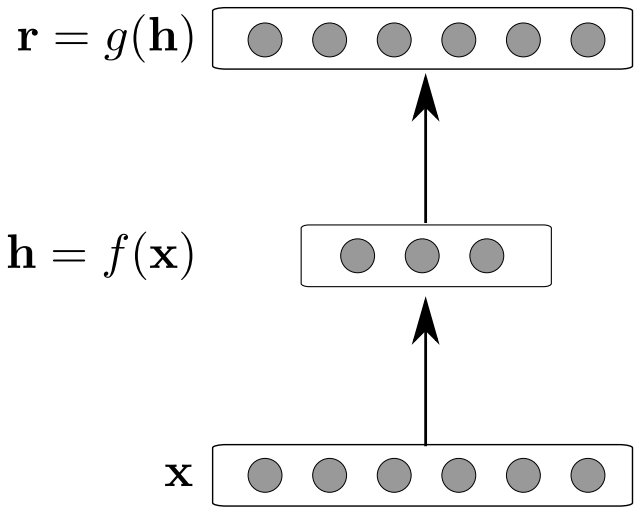
Questions?

Variational Autoencoder (VAE)

Book: 20.10.3, 20.9, Better: `https://www.jeremyjordan.me/variational-autoencoders/`

Q: How could you sample (generate samples) from an auto encoder?

Autoencoder:

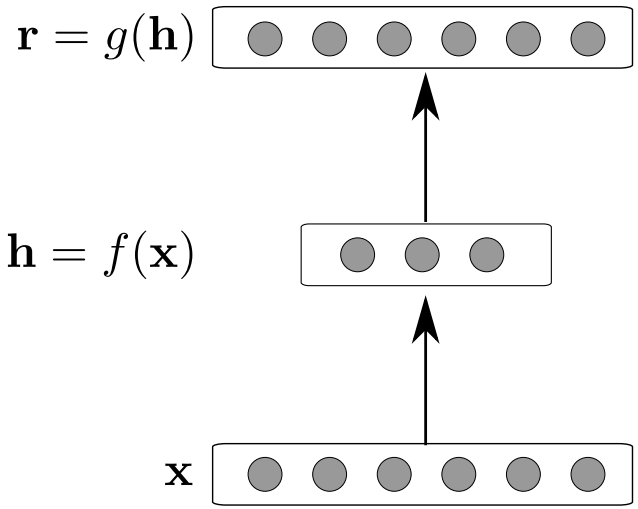


Variational Autoencoder (VAE)

Book: 20.10.3, 20.9, Better: `https://www.jeremyjordan.me/variational-autoencoders/`

Q: How could you sample (generate samples) from an auto encoder?
A: Cannot: undefined/unknown range in $\mathbf{h}$.

Autoencoder:

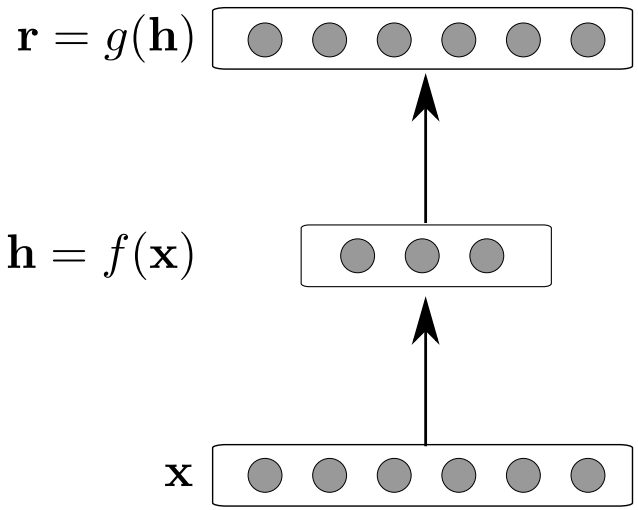


Variational Autoencoder (VAE)

Book: 20.10.3, 20.9, Better: `https://www.jeremyjordan.me/variational-autoencoders/`

Q: How could you sample (generate samples) from an auto encoder?
A: Cannot: undefined/unknown range in $\mathbf{h}$.
Idea: Make hidden layer a distribution which is easy to sample from: $\mathcal{N}(0, I)$.

Autoencoder:

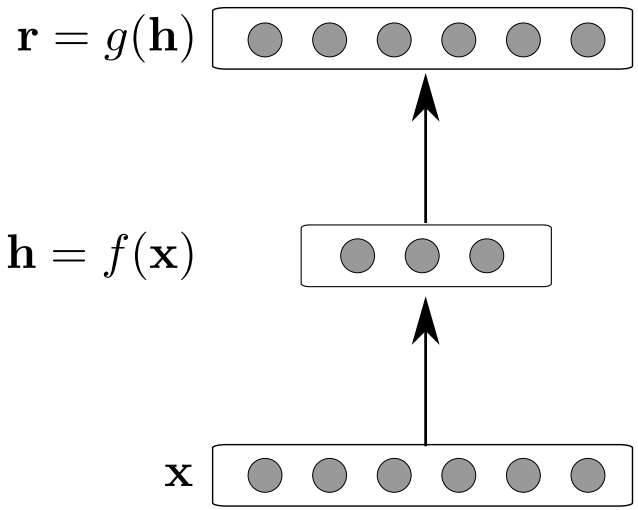


Variational Autoencoder (VAE)

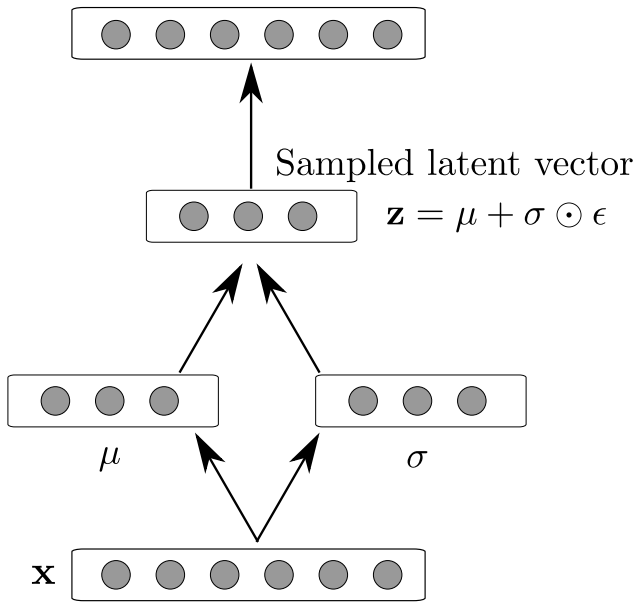
Book: 20.10.3, 20.9, Better: `https://www.jeremyjordan.me/variational-autoencoders/`

Q: How could you sample (generate samples) from an auto encoder?
A: Cannot: undefined/unknown range in $\mathbf{h}$.
Idea: Make hidden layer a distribution which is easy to sample from: $\mathcal{N}(0, I)$.

Autoencoder:



Variational autoencoder:

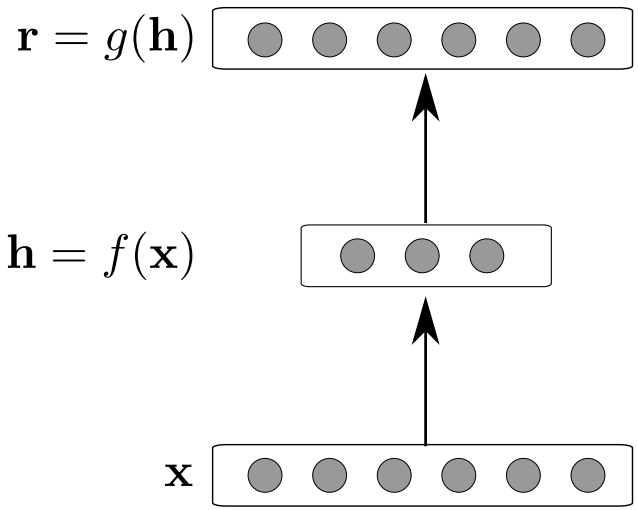


Variational Autoencoder (VAE)

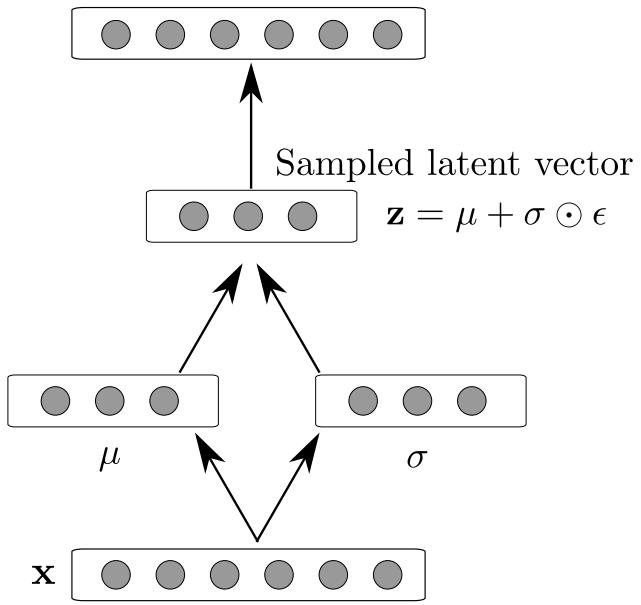
Book: 20.10.3, 20.9, Better: `https://www.jeremyjordan.me/variational-autoencoders/`

Q: How could you sample (generate samples) from an auto encoder?
A: Cannot: undefined/unknown range in $\mathbf{h}$.
Idea: Make hidden layer a distribution which is easy to sample from: $\mathcal{N}(0, I)$.

Autoencoder:



Variational autoencoder:



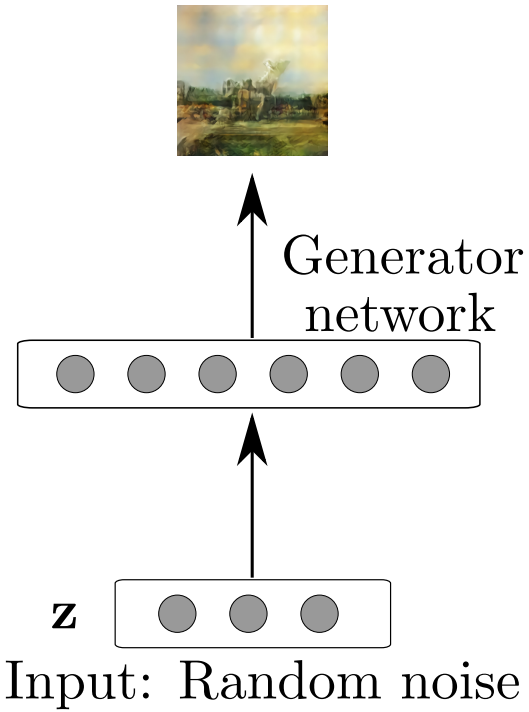
Reparameterization trick: Compute sampler gradients, draw $\epsilon \sim \mathcal{N}(0, 1)$. (Book: 20.9).

GANs: Generative Adversarial Networks

Book: Chapter 20.10.4

Introduction paper: Generative Adversarial Networks; `https://arxiv.org/abs/2203.00667`

Input white noise samples $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ to a network; let it learn the transformation from noise to training distribution.



Training GANs: Two-player adversarial game

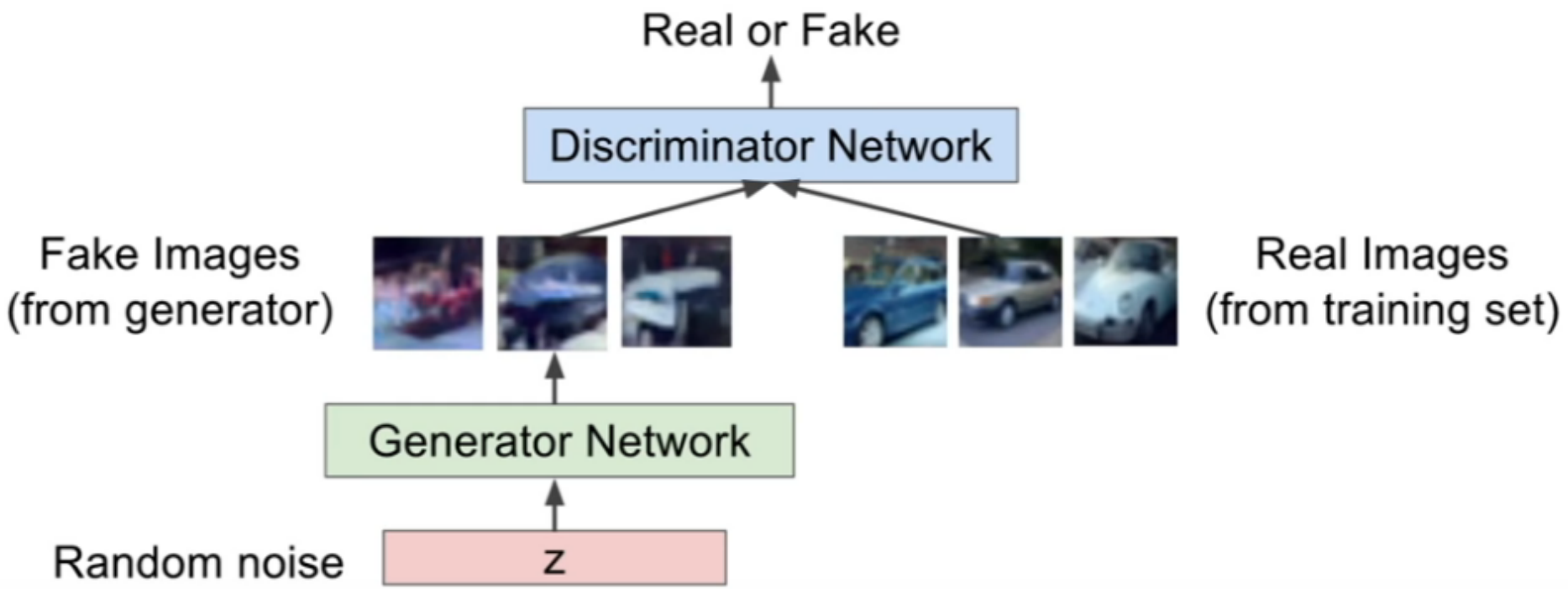
Generator network:
Try to fool the discriminator by generating real looking images.

Discriminator network:
Try to discriminate between real and fake images.

Training GANs: Two-player adversarial game

Generator network:
Try to fool the discriminator by generating real looking images.

Discriminator network:
Try to discriminate between real and fake images.



Training GANS: Two player game

Generator network:
Try to fool the discriminator by generating real looking images.

Discriminator network:
Try to discriminate between real and fake images (between 0 and 1).

Training GANS: Two player game

Generator network:
Try to fool the discriminator by generating real looking images.

Discriminator network:
Try to discriminate between real and fake images (between 0 and 1).

Objective:

$$\arg\min_{\theta_G}\max_{\theta_D}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\log D_{\theta_D}(\mathbf{x})+\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\log(1-D_{\theta_D}(G_{\theta_G}(\mathbf{z})))$$

Q: What do the terms mean? Where is the discriminator?

Training GANS: Two player game

Generator network:
Try to fool the discriminator by generating real looking images.

Discriminator network:
Try to discriminate between real and fake images (between 0 and 1).

Objective:

$$\arg\min_{\theta_G}\max_{\theta_D} \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}} \underbrace{\log D_{\theta_D}(\mathbf{x})}_{\text{D for real data } \mathbf{x}} + \mathbb{E}_{z\sim p(\mathbf{z})} \log(1 - \underbrace{D_{\theta_D}(G_{\theta_G}(\mathbf{z}))}_{\text{D for fake data}})$$

Q: What do the terms mean? Where is the discriminator?

Training GANS: Two player game

Generator network:
Try to fool the discriminator by generating real looking images.

Discriminator network:
Try to discriminate between real and fake images (between 0 and 1).

Objective:

$$\arg\min_{\theta_G}\max_{\theta_D} \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}} \underbrace{\log D_{\theta_D}(\mathbf{x})}_{\text{D for real data } \mathbf{x}} + \mathbb{E}_{z\sim p(\mathbf{z})} \log(1 - \underbrace{D_{\theta_D}(G_{\theta_G}(\mathbf{z}))}_{\text{D for fake data}})$$

Q: What do the terms mean? Where is the discriminator?
Q: What values for which terms does each network aim to obtain?

Training GANS: Two player game

Generator network:
Try to fool the discriminator by generating real looking images.

Discriminator network:
Try to discriminate between real and fake images (between 0 and 1).

Objective:

$$\arg\min_{\theta_G}\max_{\theta_D} \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}} \underbrace{\log D_{\theta_D}(\mathbf{x})}_{\text{D for real data } \mathbf{x}} + \mathbb{E}_{z\sim p(\mathbf{z})} \log(1 - \underbrace{D_{\theta_D}(G_{\theta_G}(\mathbf{z}))}_{\text{D for fake data}})$$

Q: What do the terms mean? Where is the discriminator?
Q: What values for which terms does each network aim to obtain?
A: Discriminator D_{θ_D} aims to **maximize** so that $D_{\theta_D}(\mathbf{x})$ is close to 1 for real data and $D_{\theta_D}(G_{\theta_G}(\mathbf{x}))$ is 0 for fakes.

A: Generator G_{θ_G} aims to **minimize** so that $D_{\theta_D}(G_{\theta_G}(\mathbf{z}))$ is close to 1 (fooling the discriminator)

Questions?

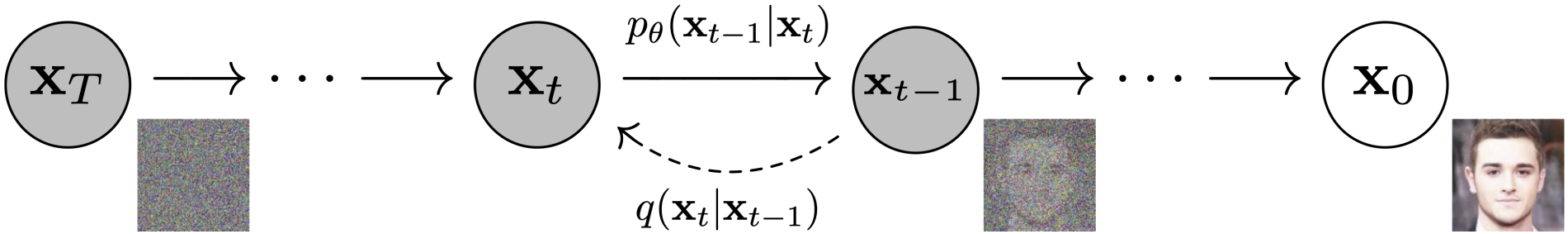
Diffusion/Score-matching https://yang-song.net/blog/2021/score/

In $t \in (0, \ldots, T)$ steps, turning an image $\mathbf{x}_0$ into noise $\mathbf{x}_T$:

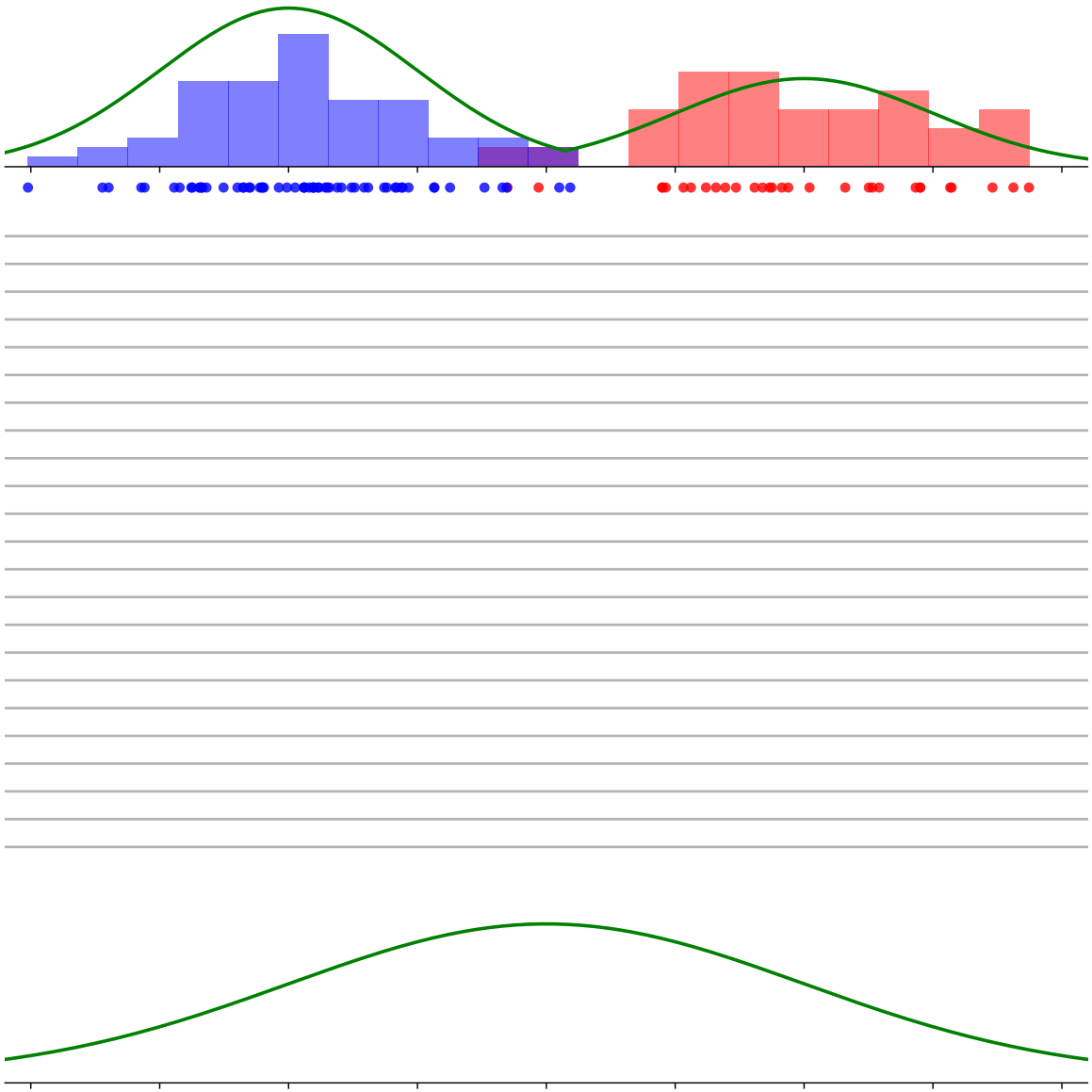
Diffusion/Score-matching https://yang-song.net/blog/2021/score/

In $t \in (0, \ldots, T)$ steps, turning an image $\mathbf{x}_0$ into noise $\mathbf{x}_T$:

Step-by-step turning the noise $\mathbf{x}_T$ back into an image $\mathbf{x}_0$:

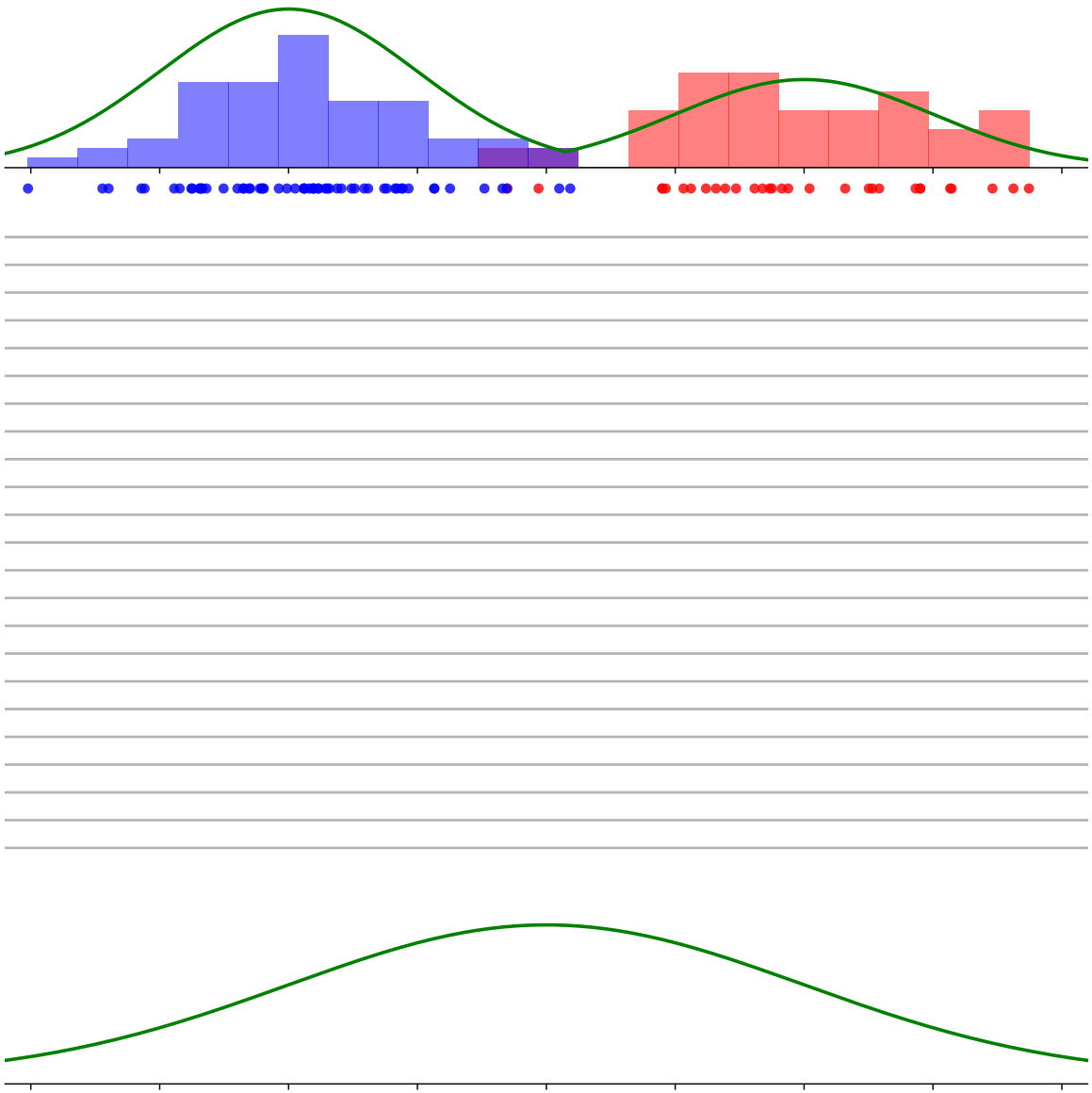


Diffusion/Score-matching https://yang-song.net/blog/2021/score/



Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

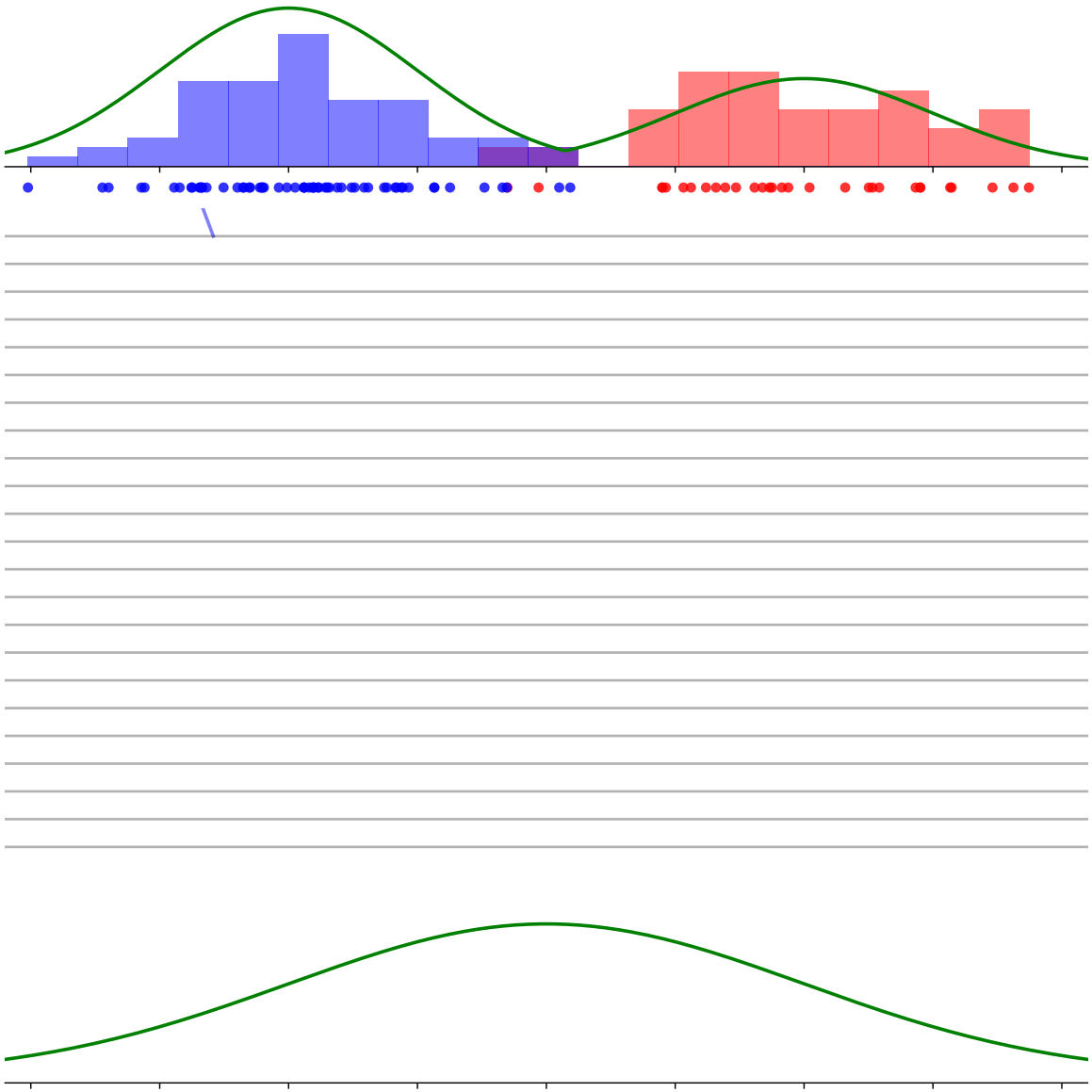
Diffusion/Score-matching https://yang-song.net/blog/2021/score/



Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Diffusion/Score-matching https://yang-song.net/blog/2021/score/

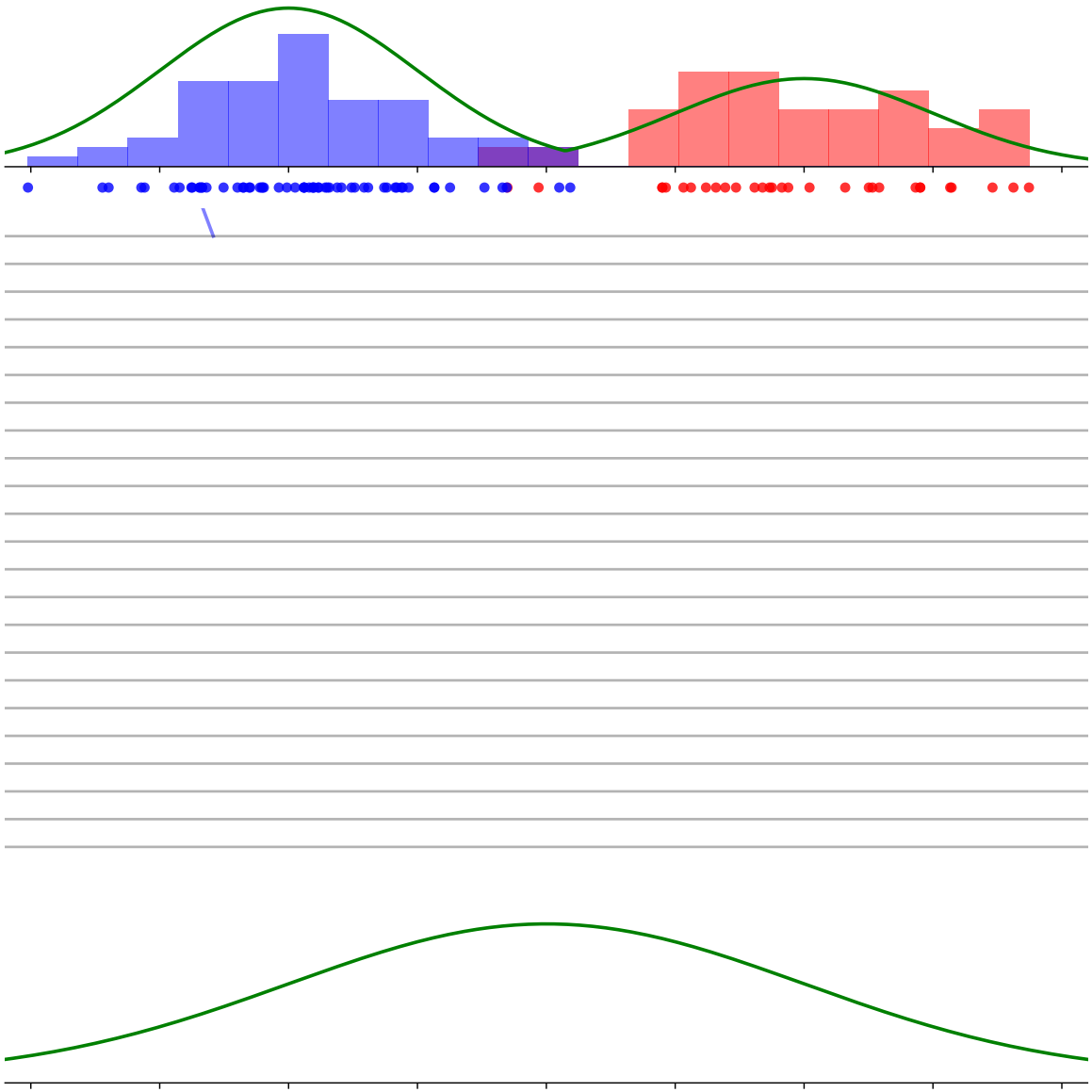


Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Q: Step from $\mathbf{x}$ to $\mu = 0$?

Diffusion/Score-matching https://yang-song.net/blog/2021/score/

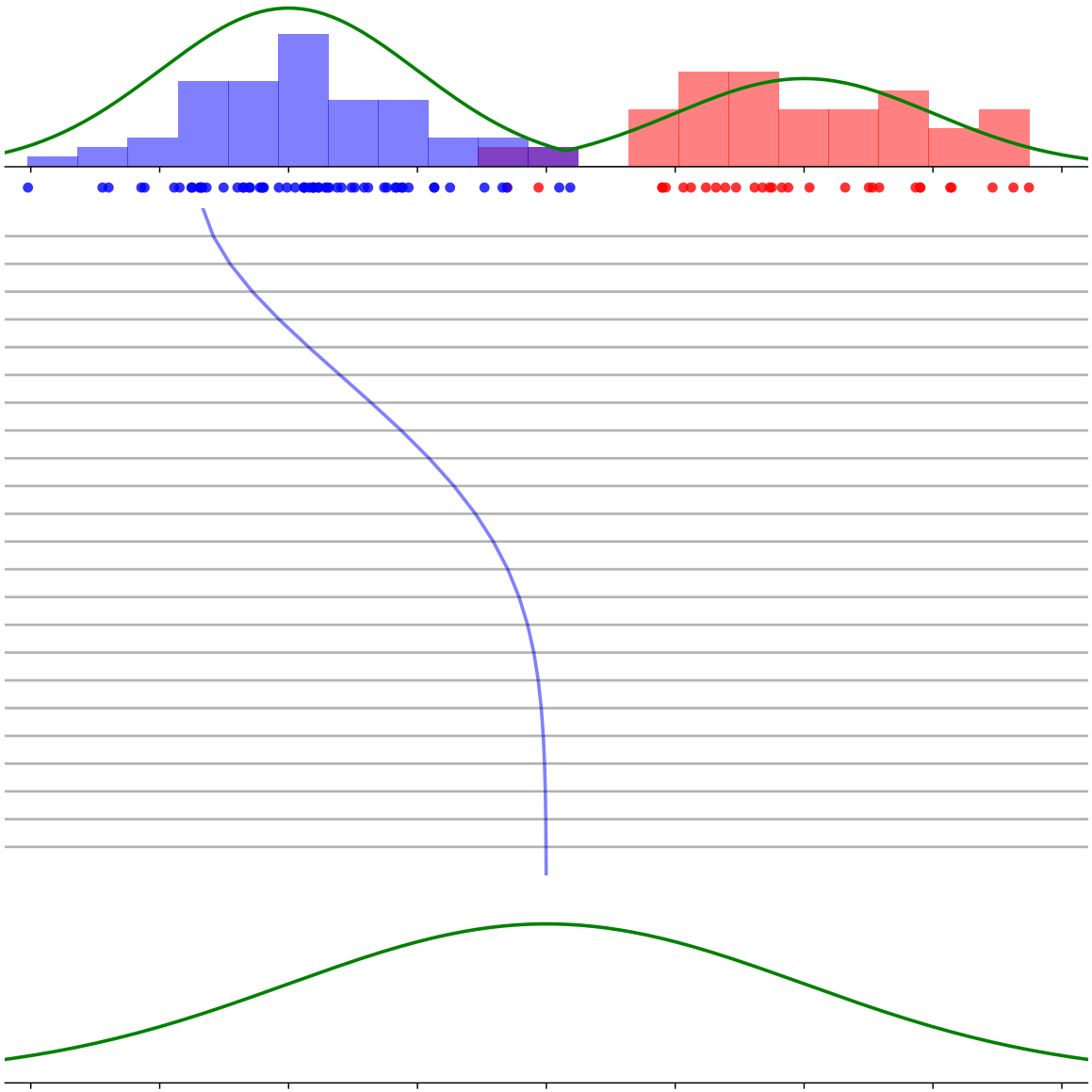


Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Q: Step from $\mathbf{x}$ to $\mu = 0$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t(1 - \beta); \quad 0 \leq \beta \leq 1$
Schedule: $\beta_{t-1} \leq \beta_t \leq \beta_{t+1}$

Diffusion/Score-matching https://yang-song.net/blog/2021/score/

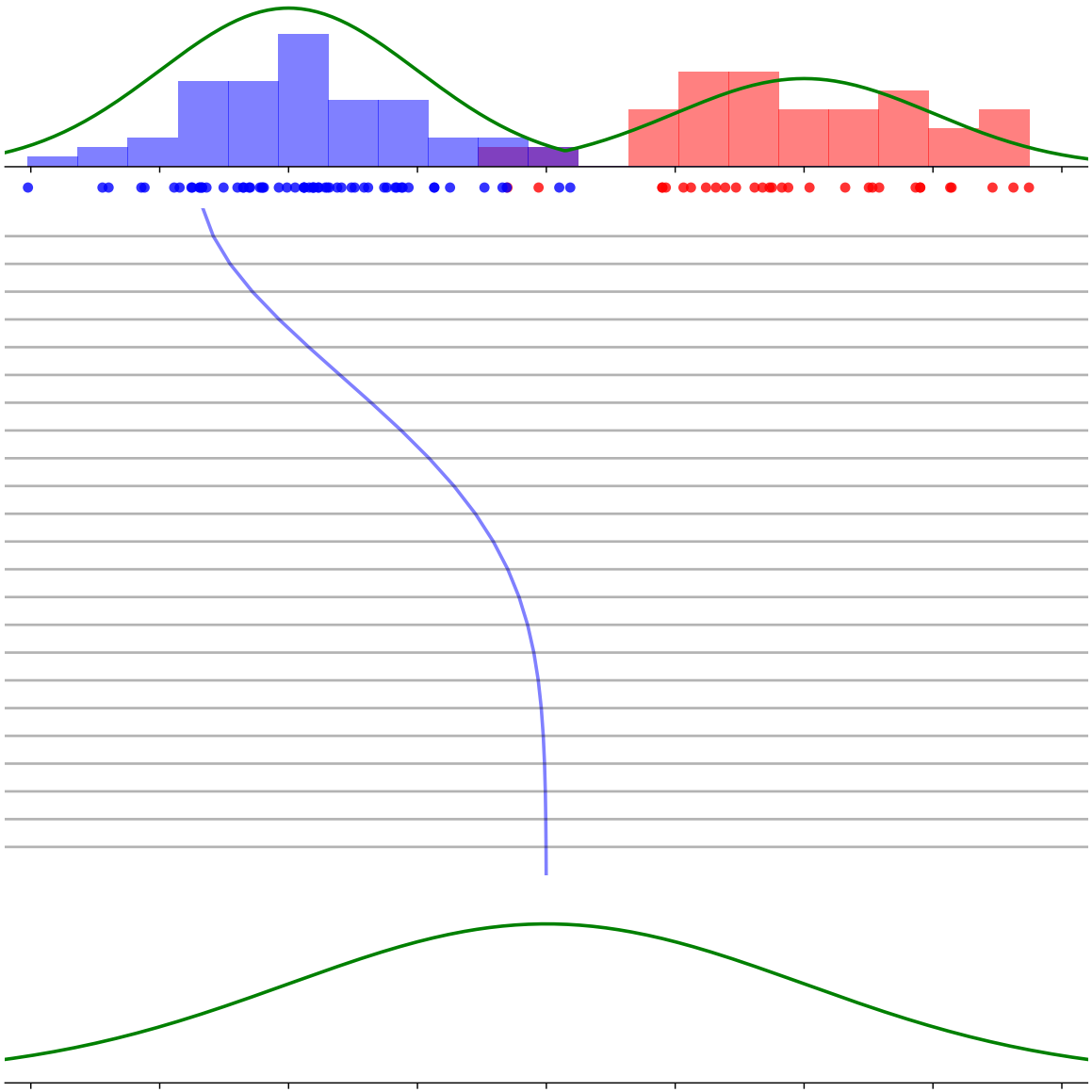


Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Q: Step from $\mathbf{x}$ to $\mu = 0$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t(1 - \beta); \quad 0 \leq \beta \leq 1$
Schedule: $\beta_{t-1} \leq \beta_t \leq \beta_{t+1}$

Diffusion/Score-matching https://yang-song.net/blog/2021/score/

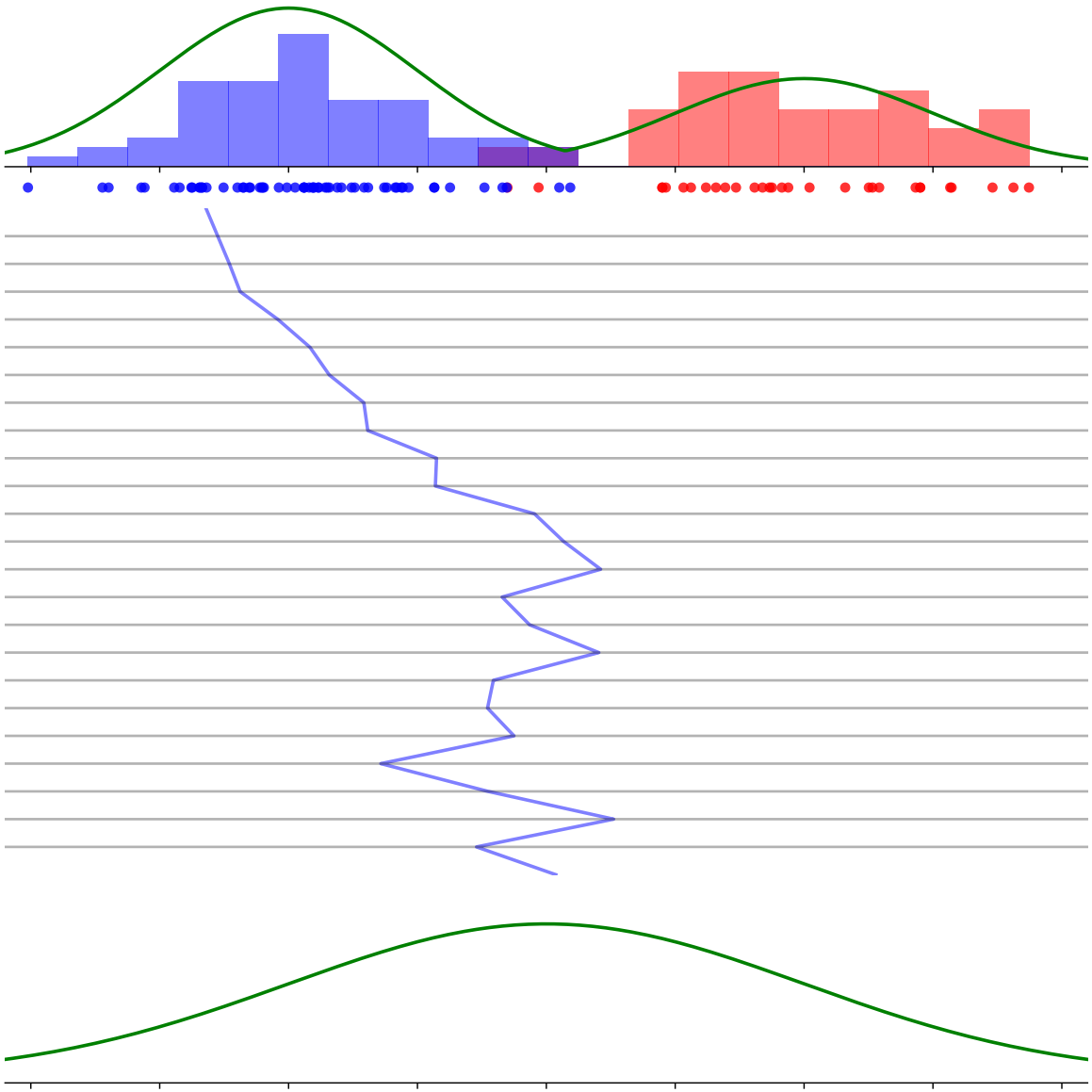


Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Q: Step from $\mathbf{x}$ to $\mu = 0$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t(1 - \beta); \quad 0 \leq \beta \leq 1$
Schedule: $\beta_{t-1} \leq \beta_t \leq \beta_{t+1}$
Q: Step from $\mathbf{x}$ to $\sigma = 1$?

Diffusion/Score-matching https://yang-song.net/blog/2021/score/



Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Q: Step from $\mathbf{x}$ to $\mu = 0$?

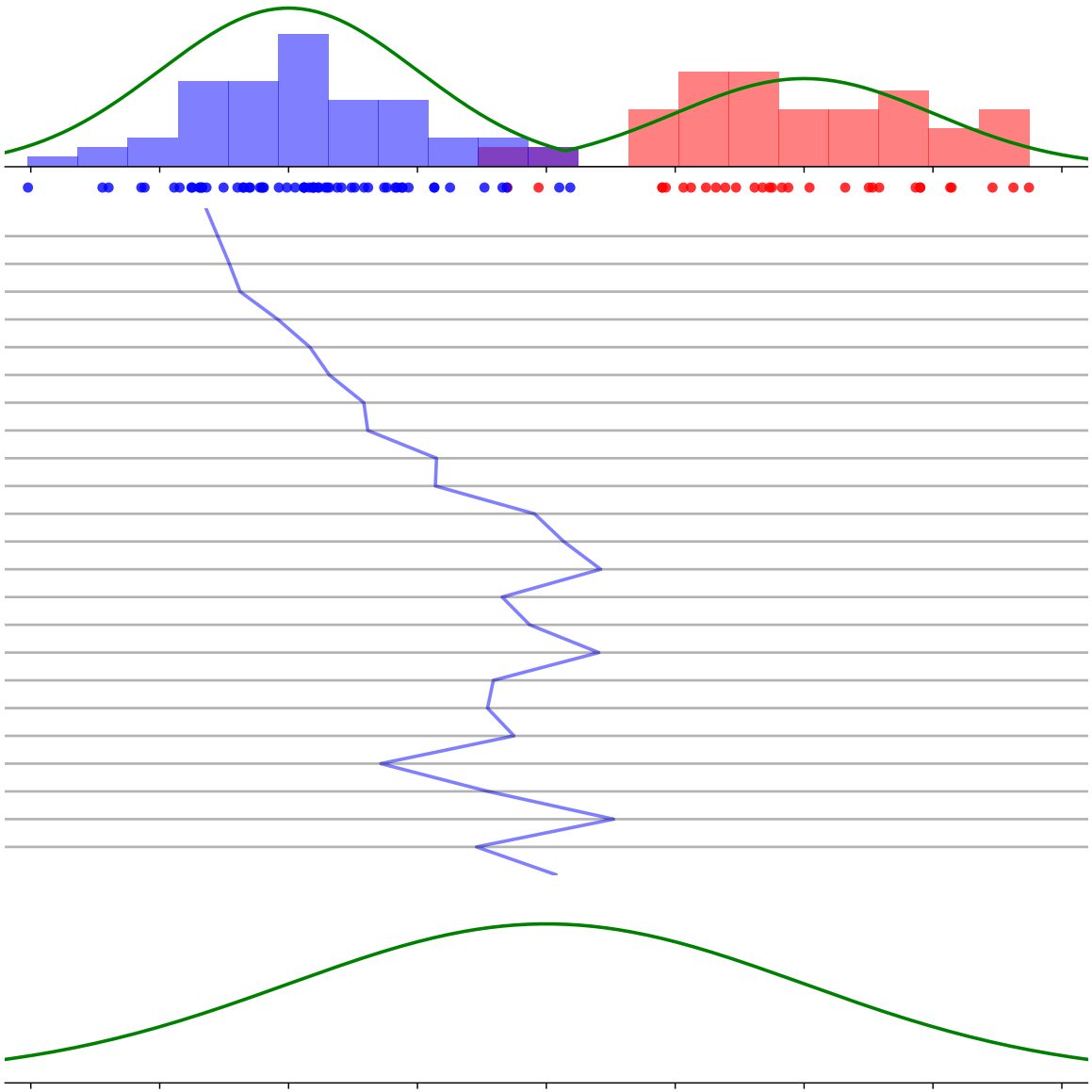
A: $\mathbf{x}_{t+1} = \mathbf{x}_t(1 - \beta); \quad 0 \leq \beta \leq 1$

Schedule: $\beta_{t-1} \leq \beta_t \leq \beta_{t+1}$

Q: Step from $\mathbf{x}$ to $\sigma = 1$?

A: $\mathbf{x}_{t+1} = \mathbf{x}_t + \beta\epsilon; \quad \epsilon \sim \mathcal{N}(0, 1)$

Diffusion/Score-matching https://yang-song.net/blog/2021/score/



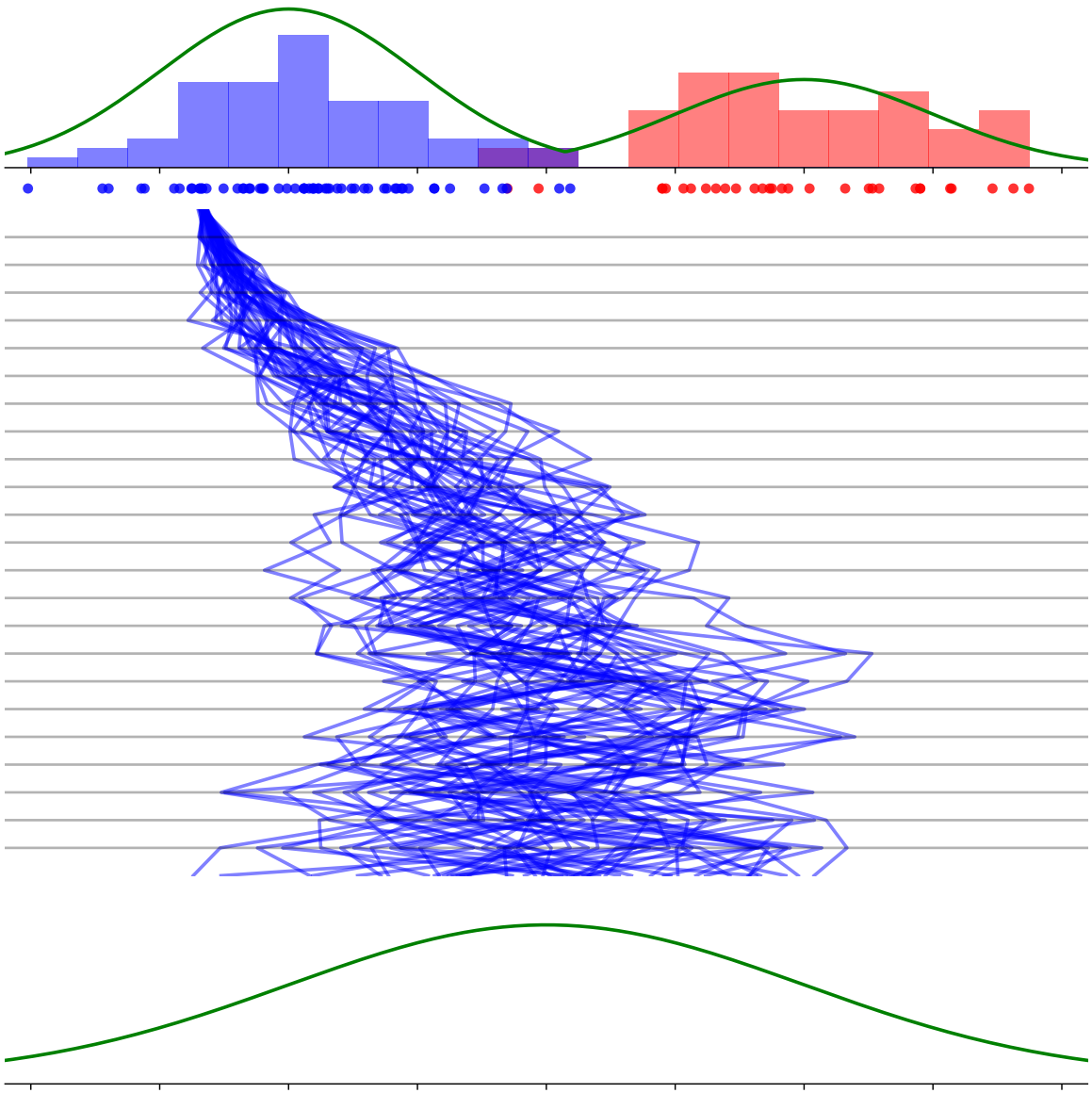
Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Q: Step from $\mathbf{x}$ to $\mu = 0$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t(1 - \beta); \quad 0 \leq \beta \leq 1$
Schedule: $\beta_{t-1} \leq \beta_t \leq \beta_{t+1}$
Q: Step from $\mathbf{x}$ to $\sigma = 1$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t + \beta\epsilon; \quad \epsilon \sim \mathcal{N}(0, 1)$

Q: Is this deterministic?

Diffusion/Score-matching https://yang-song.net/blog/2021/score/



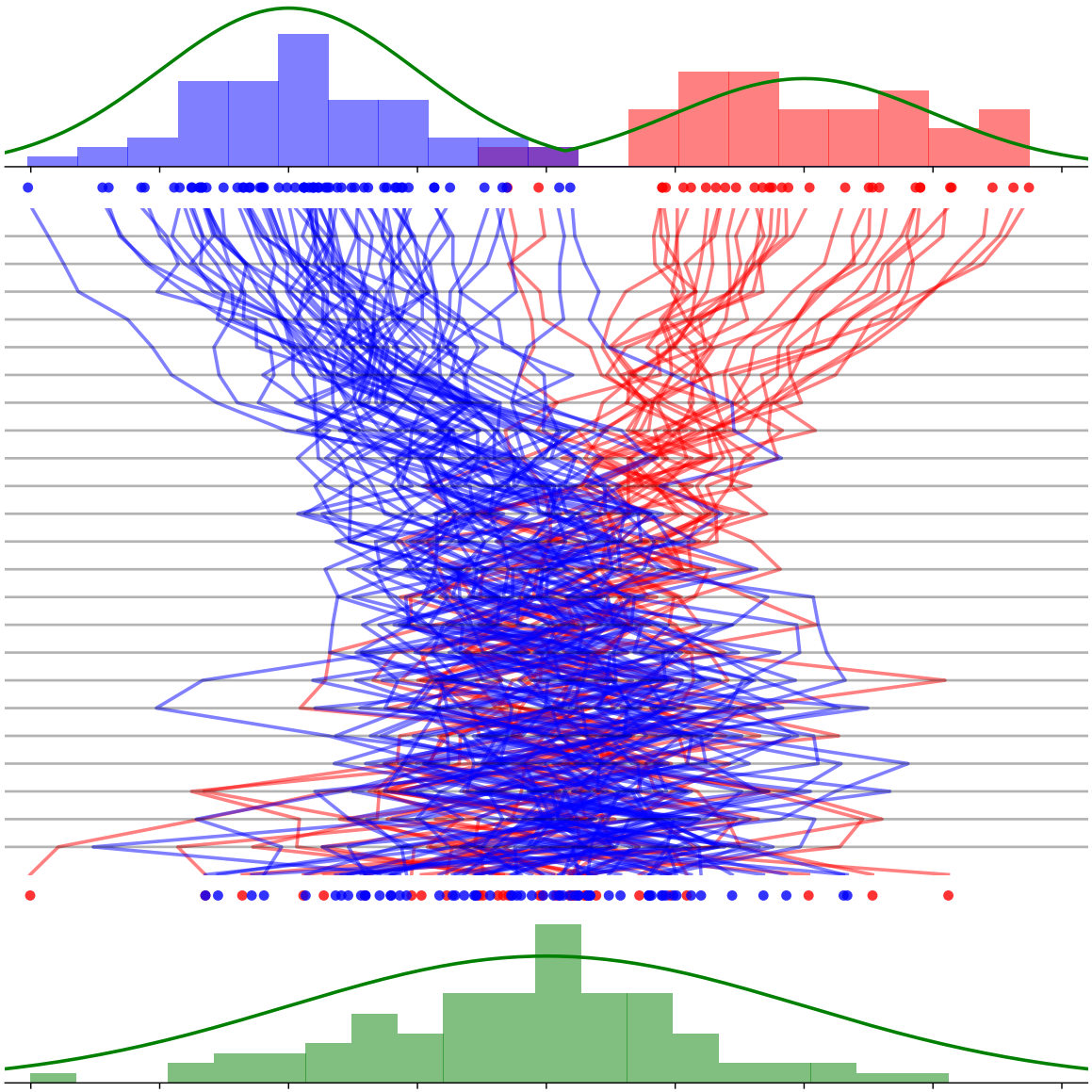
Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Q: Step from $\mathbf{x}$ to $\mu = 0$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t(1 - \beta); \quad 0 \leq \beta \leq 1$
Schedule: $\beta_{t-1} \leq \beta_t \leq \beta_{t+1}$
Q: Step from $\mathbf{x}$ to $\sigma = 1$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t + \beta\epsilon; \quad \epsilon \sim \mathcal{N}(0, 1)$

Q: Is this deterministic? A: No.

Diffusion/Score-matching https://yang-song.net/blog/2021/score/



Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

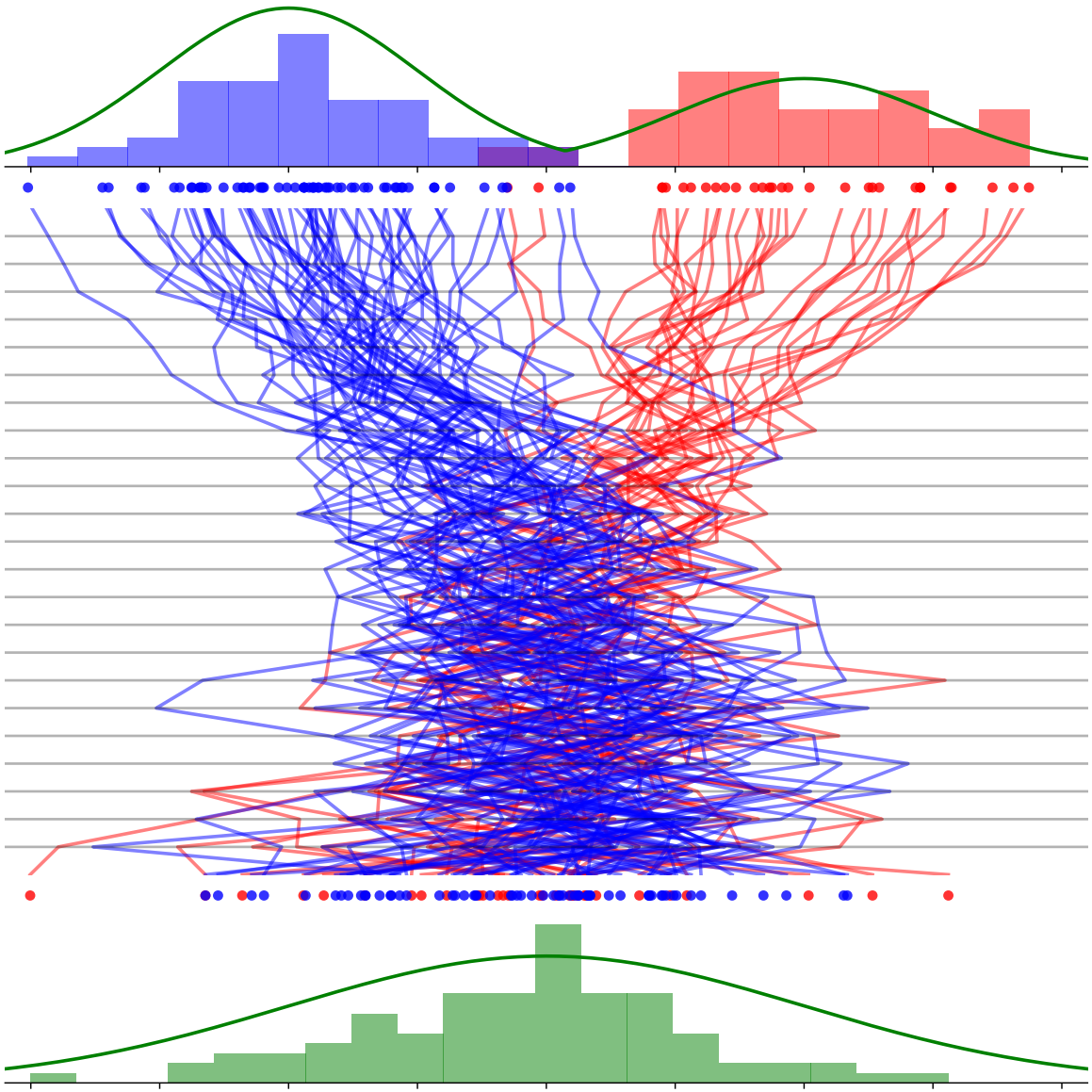
Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Q: Step from $\mathbf{x}$ to $\mu = 0$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t(1 - \beta); \quad 0 \leq \beta \leq 1$
Schedule: $\beta_{t-1} \leq \beta_t \leq \beta_{t+1}$
Q: Step from $\mathbf{x}$ to $\sigma = 1$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t + \beta\epsilon; \quad \epsilon \sim \mathcal{N}(0, 1)$

Q: Is this deterministic? A: No.

Repeatedly turn all data to noise; learn to predict the step back.

Diffusion/Score-matching https://yang-song.net/blog/2021/score/



Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

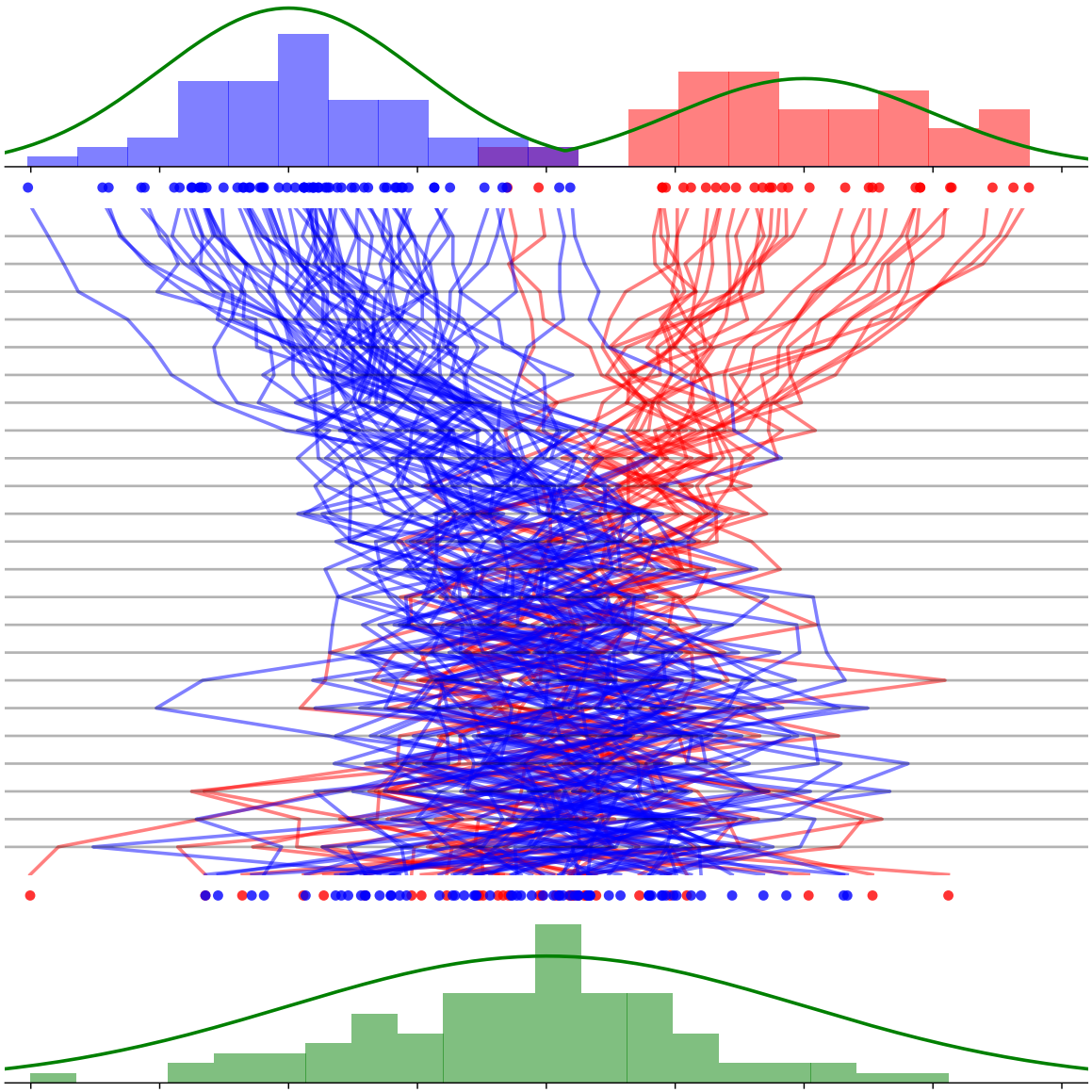
Q: Step from $\mathbf{x}$ to $\mu = 0$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t(1 - \beta); \quad 0 \leq \beta \leq 1$
Schedule: $\beta_{t-1} \leq \beta_t \leq \beta_{t+1}$
Q: Step from $\mathbf{x}$ to $\sigma = 1$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t + \beta\epsilon; \quad \epsilon \sim \mathcal{N}(0, 1)$

Q: Is this deterministic? A: No.

Repeatedly turn all data to noise; learn to predict the step back.

Q: What type of network?

Diffusion/Score-matching https://yang-song.net/blog/2021/score/



Sample from $\mathcal{N}(\mu = 0, \sigma = 1)$; update step-by-step using $\nabla_{\mathbf{x}} p(\mathbf{x})$

Take steps from $\mathbf{x}$ to $\mathcal{N}(0, 1)$. Each step is $-\nabla_{\mathbf{x}} p(\mathbf{x})$. Learn how to take a step back: $\nabla_{\mathbf{x}} p(\mathbf{x})$

Q: Step from $\mathbf{x}$ to $\mu = 0$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t(1 - \beta); \quad 0 \leq \beta \leq 1$
Schedule: $\beta_{t-1} \leq \beta_t \leq \beta_{t+1}$
Q: Step from $\mathbf{x}$ to $\sigma = 1$?
A: $\mathbf{x}_{t+1} = \mathbf{x}_t + \beta\epsilon; \quad \epsilon \sim \mathcal{N}(0, 1)$

Q: Is this deterministic? A: No.

Repeatedly turn all data to noise; learn to predict the step back.

Q: What type of network?
A: Denoising auto encoder.

Questions?

How to evaluate generative models?

Book: Chapter 20.14

- Q: Can't we just evaluate likelihoods?

How to evaluate generative models?

Book: Chapter 20.14

- Q: Can't we just evaluate likelihoods?
  A: Higher likelihoods is not necessarily visually better than a lower one.

- Q: Can't we let external evaluators evaluate model quality?

How to evaluate generative models?

Book: Chapter 20.14

- Q: Can't we just evaluate likelihoods?
  A: Higher likelihoods is not necessarily visually better than a lower one.
- Q: Can't we let external evaluators evaluate model quality?
  A: Just copying the training set will do very well
- Q: Can't we use the unsupervised features to evaluate another task (e.g., classification accuracy)?

How to evaluate generative models?

Book: Chapter 20.14

- Q: Can't we just evaluate likelihoods?
  A: Higher likelihoods is not necessarily visually better than a lower one.
- Q: Can't we let external evaluators evaluate model quality?
  A: Just copying the training set will do very well
- Q: Can't we use the unsupervised features to evaluate another task (e.g., classification accuracy)?
  A: Yes, but that does limit the use of generative models

How to evaluate generative models is an active research topic.

Questions?

Questions?

Topics:

- Auto encoders
- Sampling from unknown distribution by transforming a known distribution
- Variational autoencoder (VAE); GANs, Diffusion/Score-matching
- Evaluation