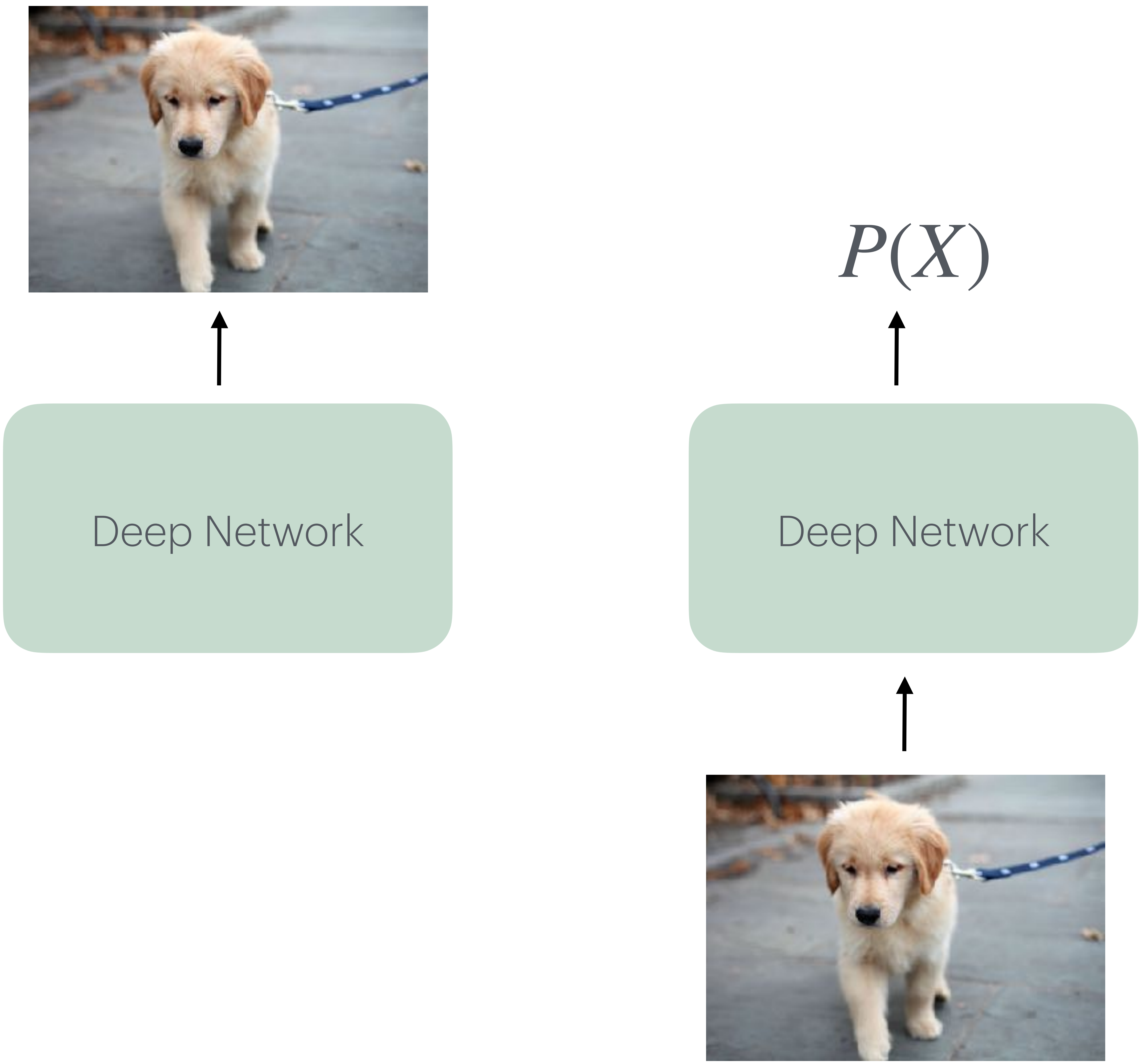
Diffusion

Philipp Krähenbühl, UT Austin

Generative models

- Two tasks of a generative model $P(X)$
  - Sampling: $x \sim P(X)$
  - Density estimation: $P(X = x)$

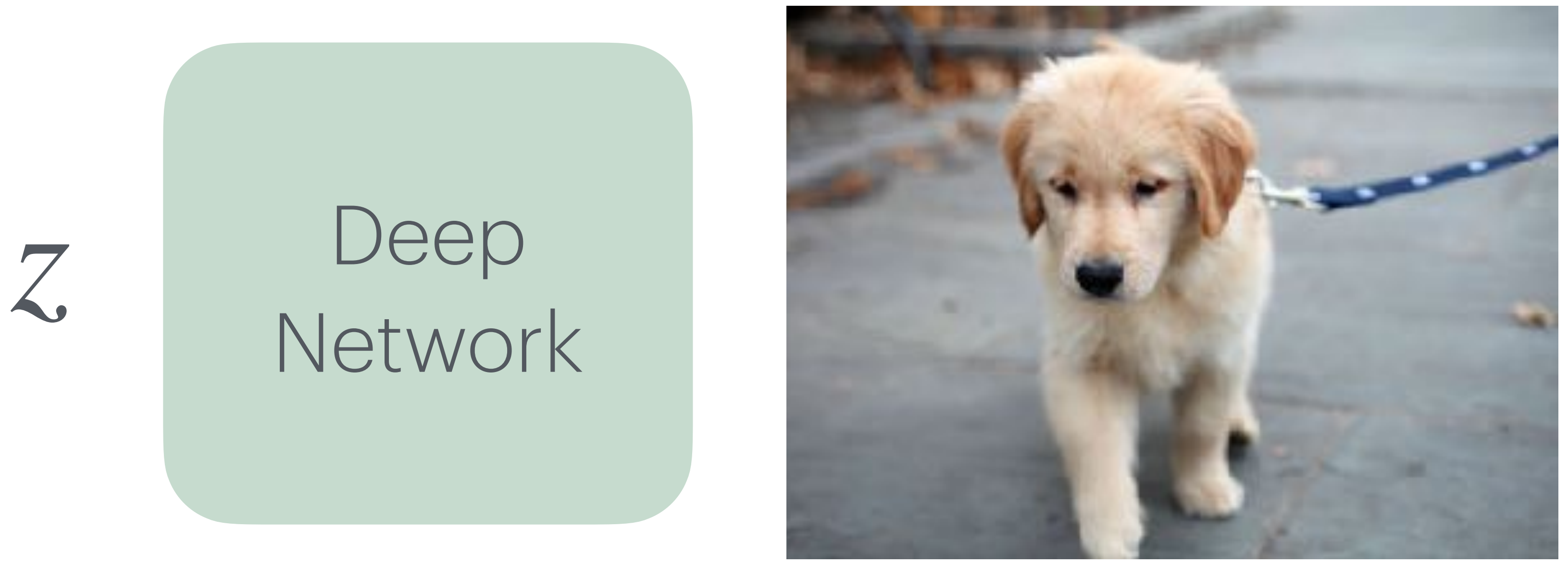


Generative models

Two kinds of models

Sampling based $x \sim P(X)$

- Sample $z \sim P(Z)$
- Learn transformation
  - $P(x|z)$ or $f: z \rightarrow x$

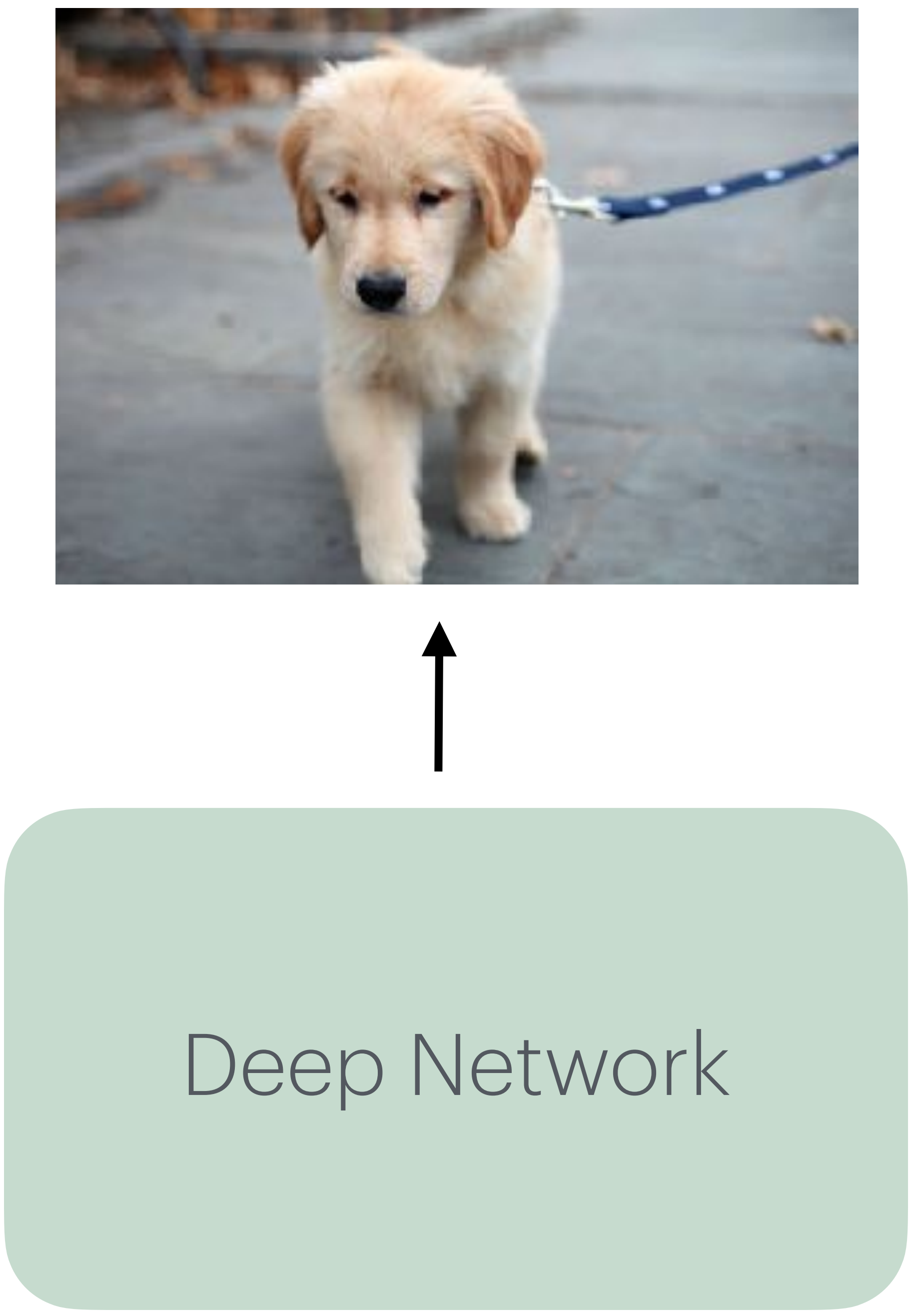


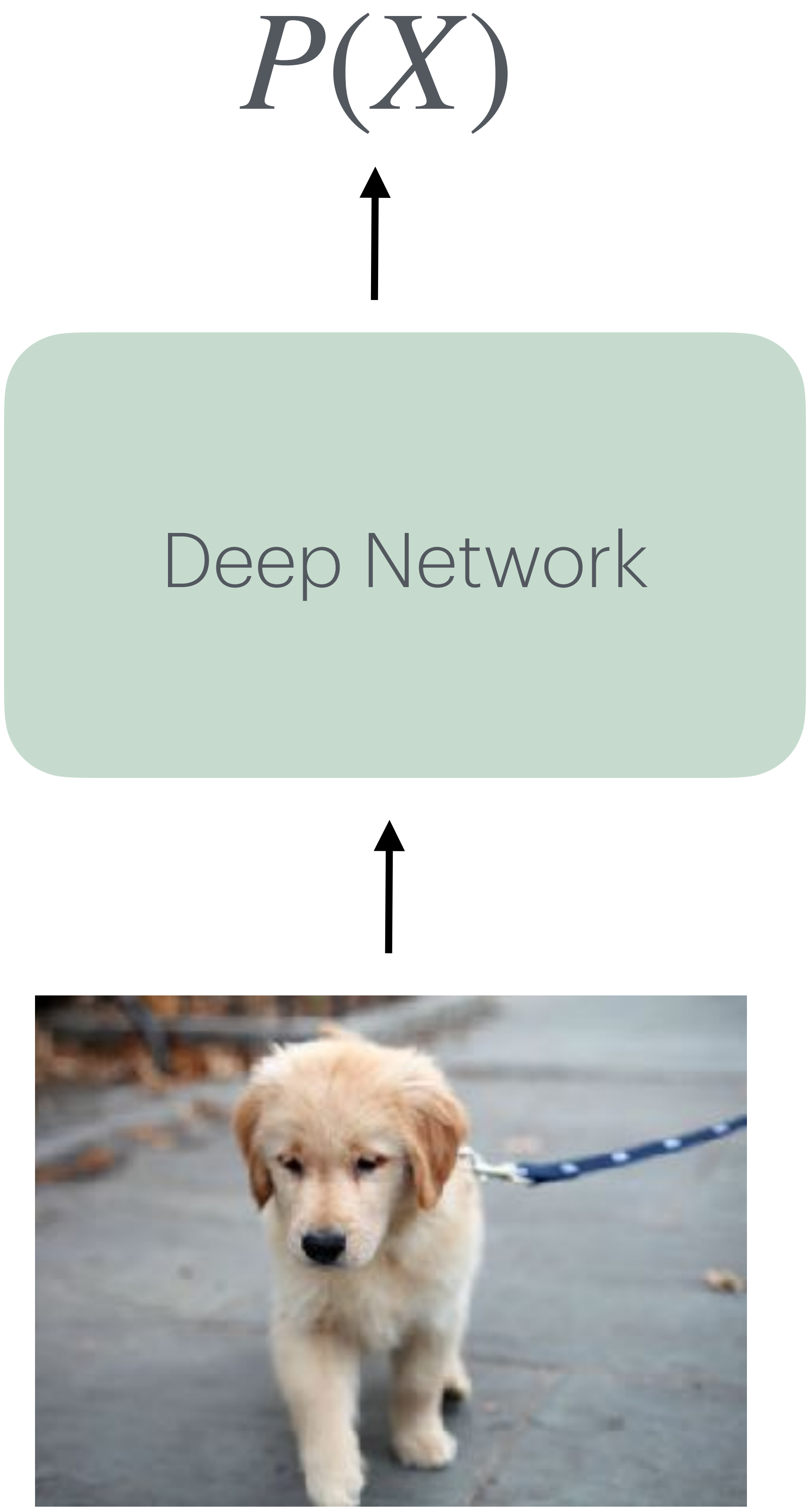
Density estimation based $P(X)$

- Learn special form of $P(X)$
- Model specific sampling / generation

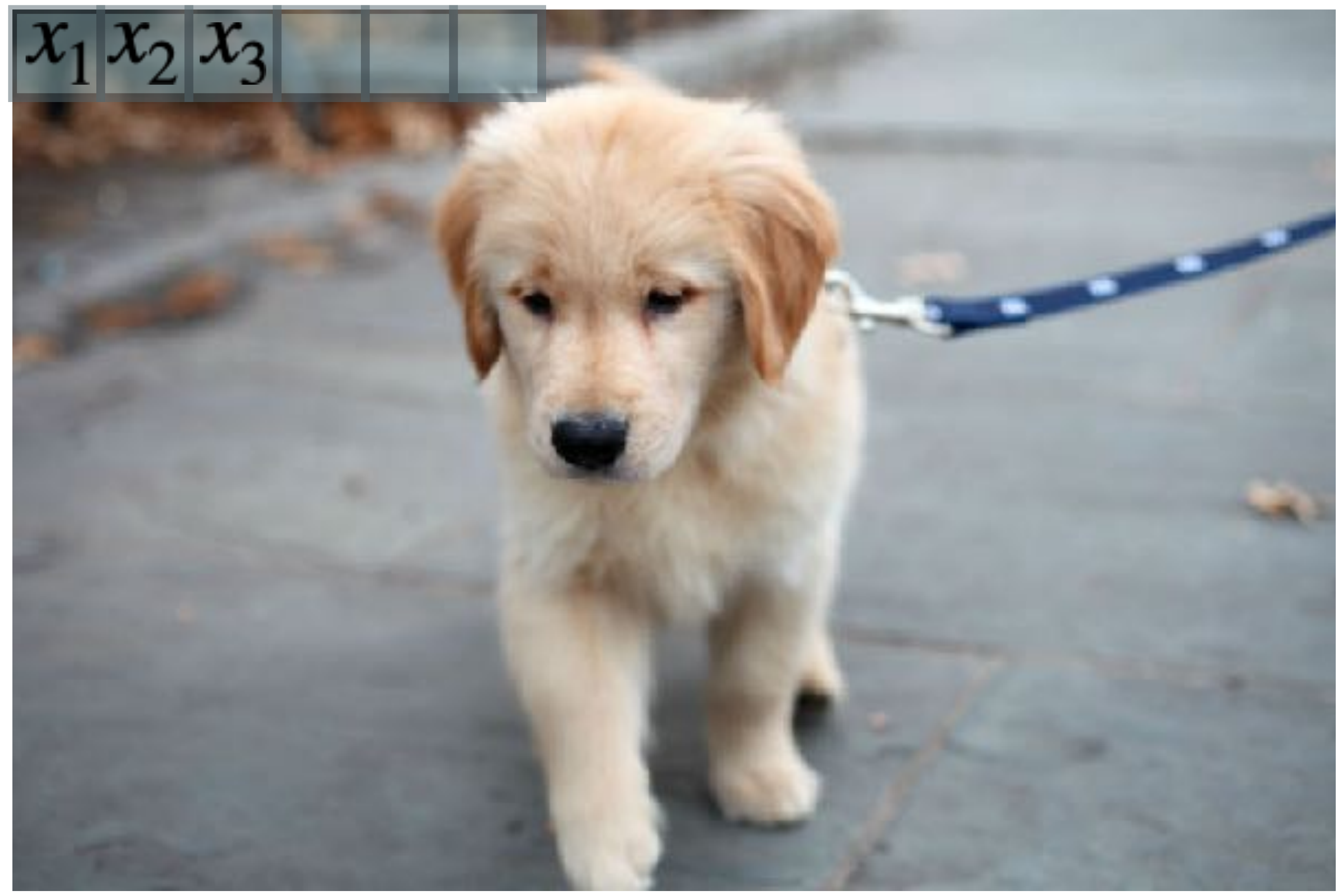
Generative modeling is hard

- Density estimation $P(X = x)$
  - How to ensure $\sum_x P(x) = 1$ for all x
  - Impossible to compute (in general)





Auto-regressive models



$P(x) = P(x_1)P(x_2 | x_1)P(x_3 | x_1, x_2)P(x_4 | x_1 \dots x_3)\dots$

- $P(x_i | x_1 \dots x_{i-1}) = \text{softmax}(f(x_1 \dots x_{i-1}))$
- Basis of most LLM models
- Easy estimation of $P(x)$
- Easy sampling
  $x_1 \sim P(X_1); x_2 \sim P(X_2 | x_1)$
  - Slow sampling

[1] WaveNet: A Generative Model for Raw Audio. Aaron van den Oord, et al. 2016
[2] Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer. Songwei Ge, et al. 2022

Auto-regressive models

Issues

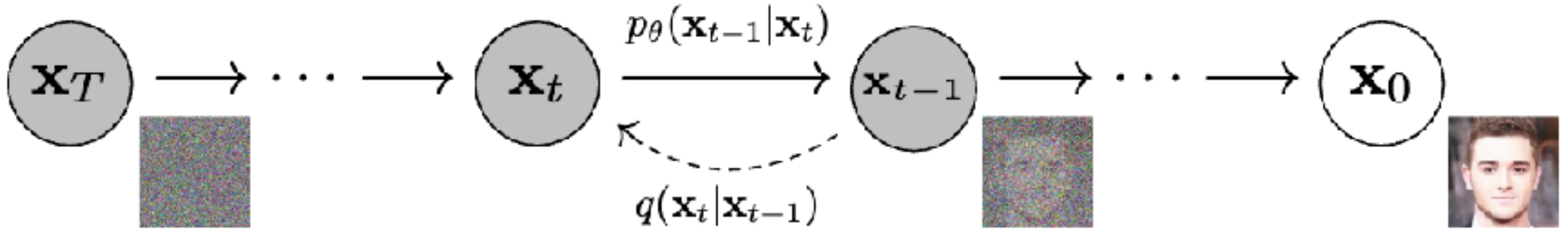
$$P(x) = P(x_1)P(x_2 | x_1)P(x_3 | x_1, x_2)P(x_4 | x_1 \ldots x_3)\ldots$$

- Difficult learning problem for long sequences (requires good model)
- Solution: Tokenization/Vector-Quantization (next class)
  - More complex x_i
  - Shorter sequence

[1] WaveNet: A Generative Model for Raw Audio. Aaron van den Oord, et al. 2016
[2] Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer. Songwei Ge, et al. 2022

Diffusion Process

Forward process

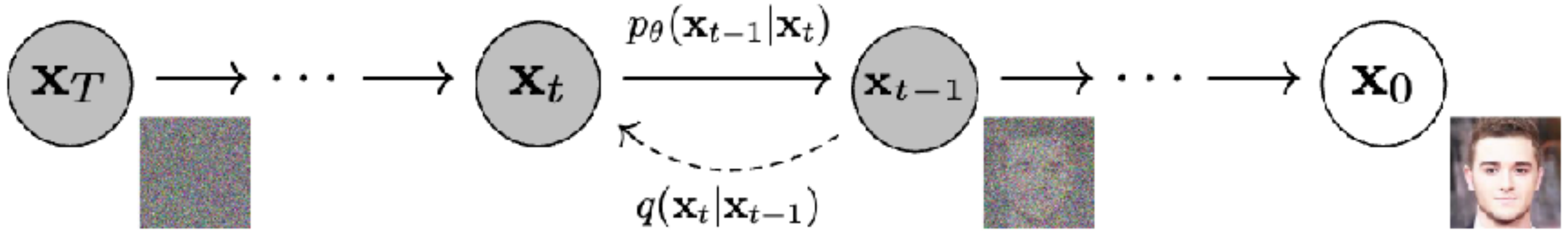


- Make an image noisy
  - Start with an image x_0
  - Add noise $q(x_t | x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t} x_t, \beta_t I)$
  - β_t increases linearly with t
- $$q(x_t | x_0) = \int \prod_{i=1}^{t} q(x_i | x_{i-1}) dx_{1\ldots t-1}$$
  $$= \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t) I)$$
  where $\bar{\alpha}_t = \prod_{i=1}^{t} (1-\beta_i)$

[3] Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

Diffusion Process

Reverse process



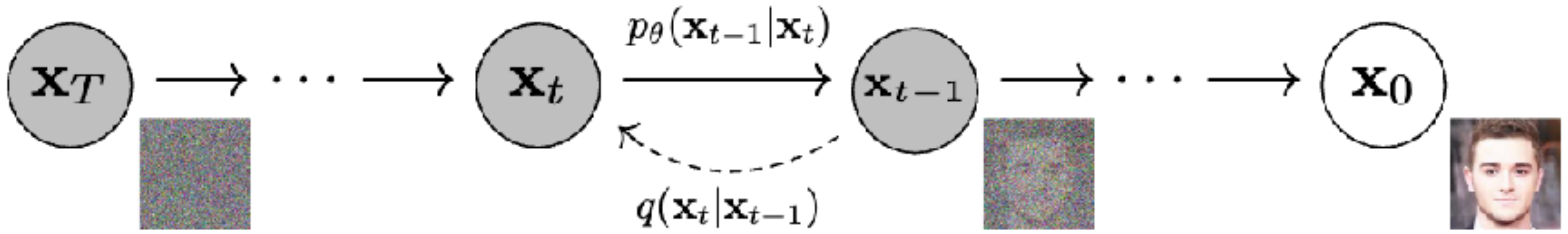
- Learn to predict image progressively
  - Start $P(x_T) = \mathcal{N}(0,I)$
  - Denoise $P(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$
- Reverse process

$$P(x_{0...T}) = P(x_T)\prod_{t=1}^{T} P(x_{t-1} \mid x_t)$$

- $P(x_0) = \int P(x_{0...T})dx_{1...T}$

[3] Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

Diffusion Process



- Maximize Evidence lower bound (ELBO)

$$\log P(x_0) \geq E_q \left[\log \frac{P(x_{0\ldots T})}{q(x_{1\ldots T} | x_0)} \right]$$

- (Lot's of math later)
- Relatively simply training and sampling algorithms
    - $\epsilon(x_t, t)$ is a noise-prediction network

Algorithm 1 Training

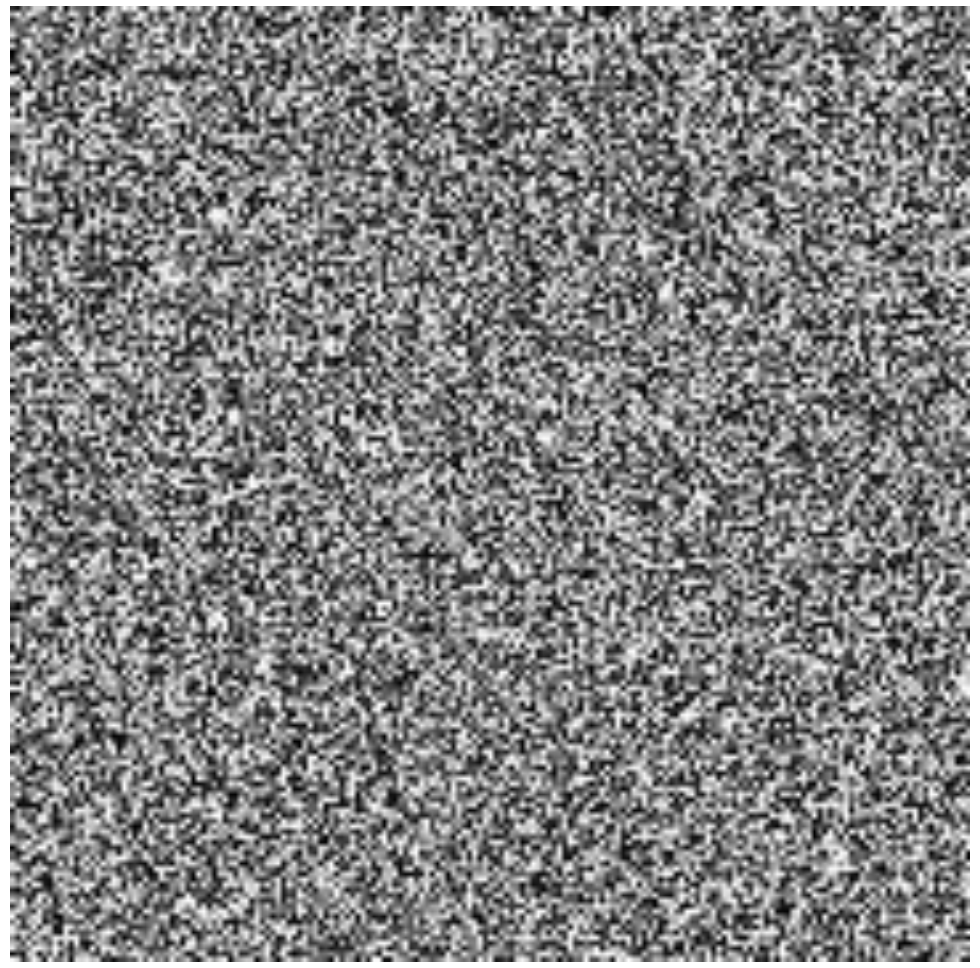
repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

[3] Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

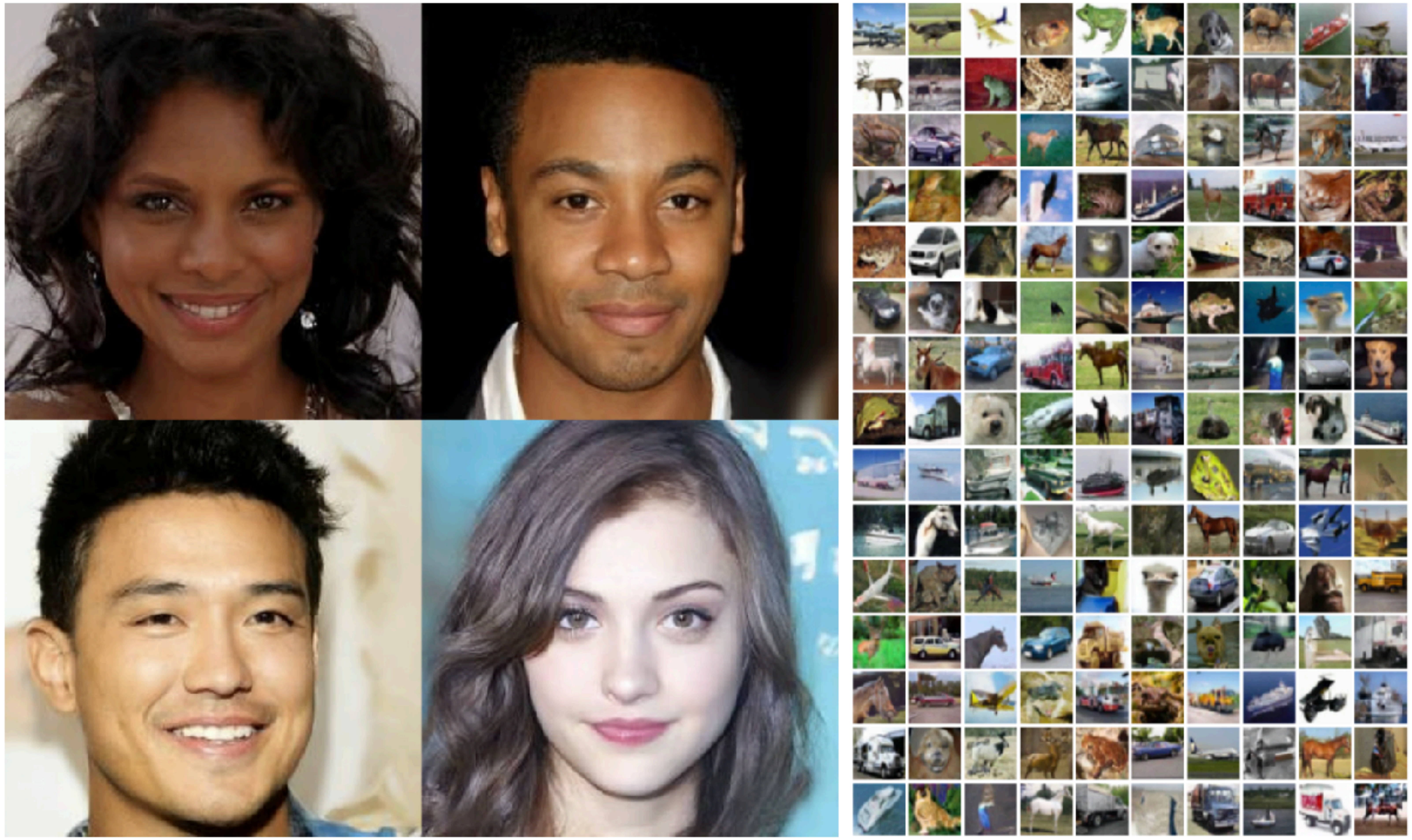
Diffusion Model



U-Net

[3] Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

Diffusion - Results



[3] Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.

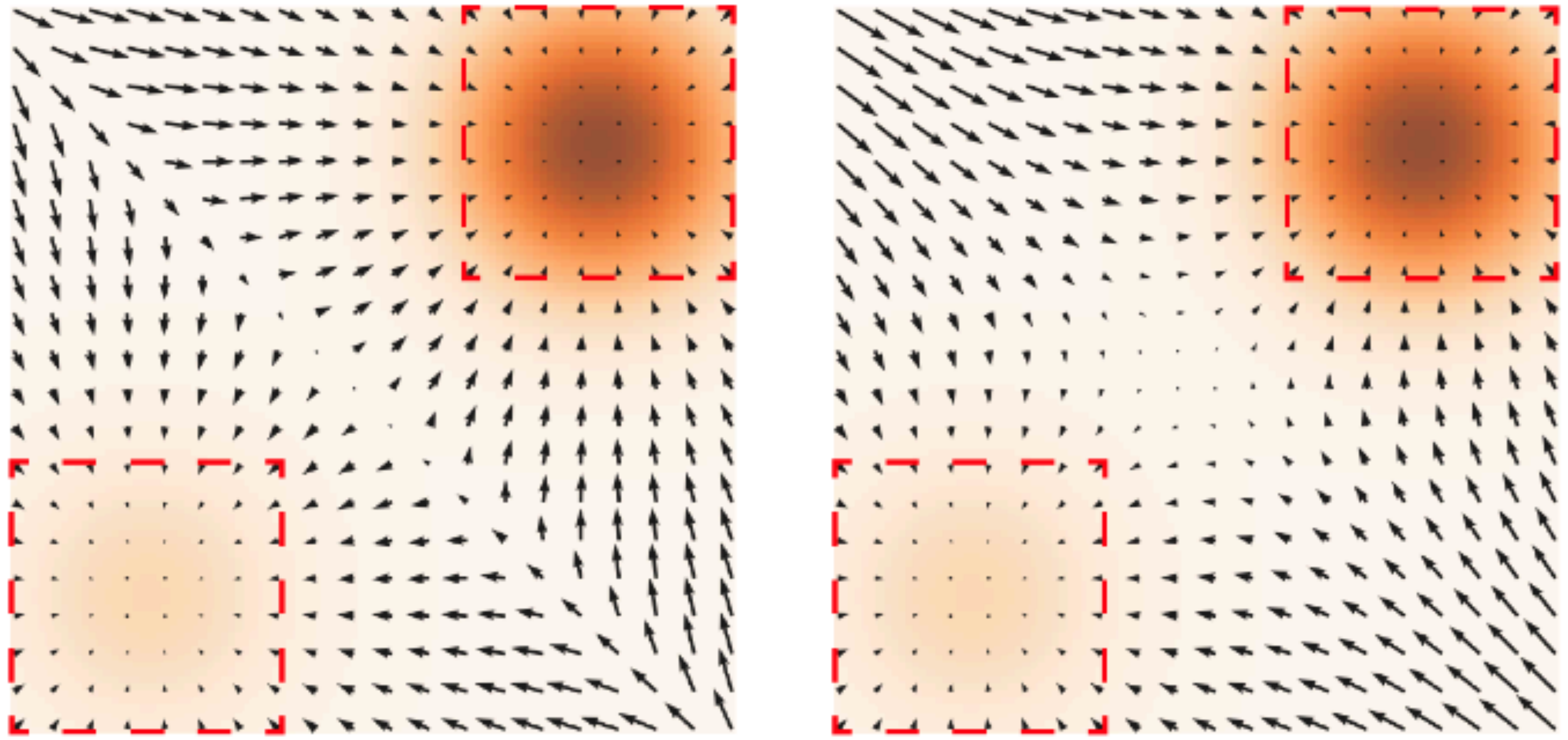
Score-based models

- $P(x)$ / $\log P(x)$
  - is hard to learn
- $\nabla \log P(x)$ (score function)
  - is easier to learn/estimate

$$E_{x\sim P}\left[\left|s(x) - \nabla P(x)\right|^2\right] =$$

- $$E_{x\sim P}\left[\mathrm{tr}(\nabla_x s(x)) + \frac{1}{2}\left|s(x)\right|^2\right] + \mathrm{const}$$

Data scores Estimated scores



Algorithm 1 Annealed Langevin dynamics.

```
Require: {σ_i}_{i=1}^L, ε, T.
  Initialize x̃_0
  for i ← 1 to L do
      α_i ← ε · σ_i^2 / σ_L^2          ▷ α_i is the step size.
      for t ← 1 to T do
          Draw z_t ~ N(0, I)
          x̃_t ← x̃_{t−1} + (α_i / 2) s_θ(x̃_{t−1}, σ_i) + √α_i z_t
      end for
      x̃_0 ← x̃_T
  end for
  return x̃_T
```

[4] Generative Modeling by Estimating Gradients of the Data Distribution. Yang Song, et al. 2019.

Guided Diffusion

- Learn variance $\Sigma(x_t)$
- Better architecture
  - Deeper, more attention heads, attention on multiple blocks, …
- Classifier guidance (conditioning)

[5] Diffusion Models Beat GANs on Image Synthesis. Prafulla Dhariwal, et al. 2021.

Diffusion

- Very good image quality
- Not easily controllable
- Computationally quite expensive
  - Multiple sampling steps
  - Fairly high resolution inputs and outputs required (original image size)

References

- [1] WaveNet: A Generative Model for Raw Audio. Aaron van den Oord, et al. 2016
- [2] Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer. Songwei Ge, et al. 2022
- [3] Denoising Diffusion Probabilistic Models. Jonathan Ho, et al. 2020.
- [4] Generative Modeling by Estimating Gradients of the Data Distribution. Yang Song, et al. 2019.
- [5] Diffusion Models Beat GANs on Image Synthesis. Prafulla Dhariwal, et al. 2021.