Text Diffusion Models

Agenda

- Problems of autoregressive text generation
- Diffusion models: reminder
- Text diffusion models:
    - Discrete diffusion
    - Continuous diffusion

Autoregressive text generation

Generate one token at a time

The next token is __

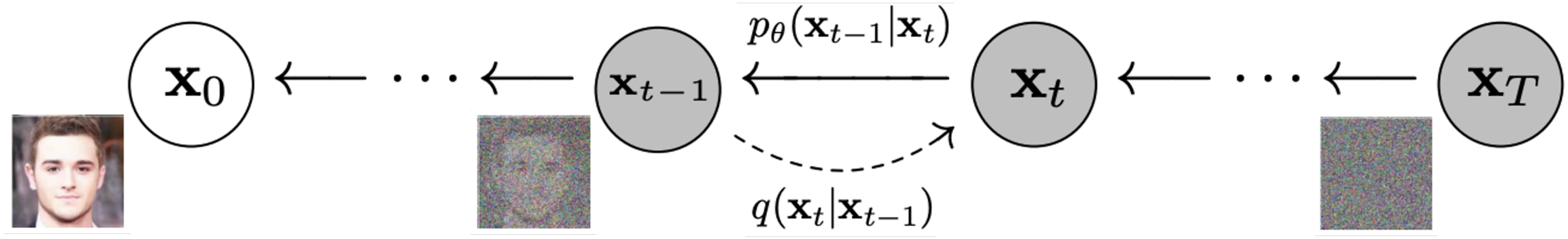
Disadvantages:

- Can't correct previously generated tokens
- Can't think a several tokens ahead
- Need to choose a sampling method

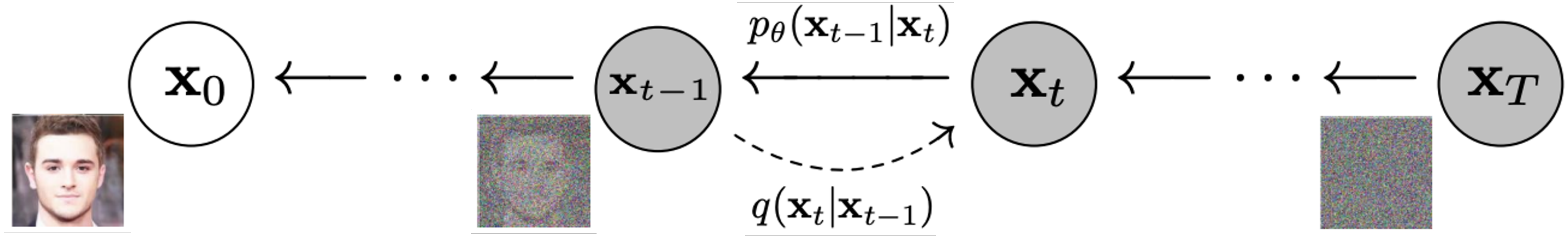
Diffusion models

Diffusion models were originally made for image generation

Idea: gradually add noise to an object during the forward diffusion process
Learn a model to denoise objects x_t at each noise level



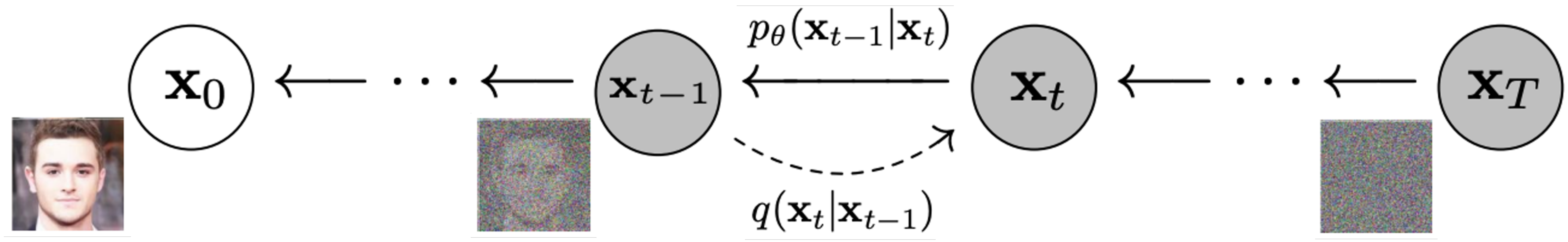
Diffusion models



$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t \mid \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t) I)$$

$$\alpha_t \in [0, 1]$$

Diffusion models



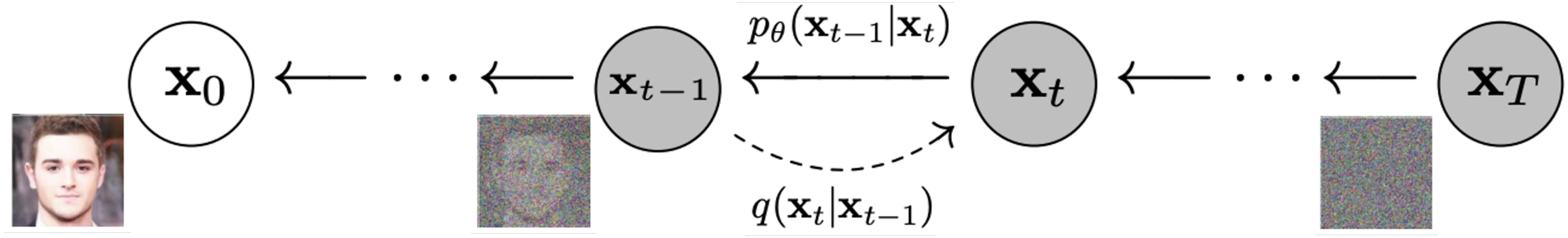
$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t \mid \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t)I)$$

$$q(x_t \mid x_0) = \mathcal{N}(x_t \mid \sqrt{\bar{\alpha}_t} x_{t-1}, (1-\bar{\alpha}_t)I)$$

$$\alpha_t \in [0, 1]$$

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

Diffusion models



$$q(x_t \,|\, x_{t-1}) = \mathcal{N}(x_t \,|\, \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t)I)$$

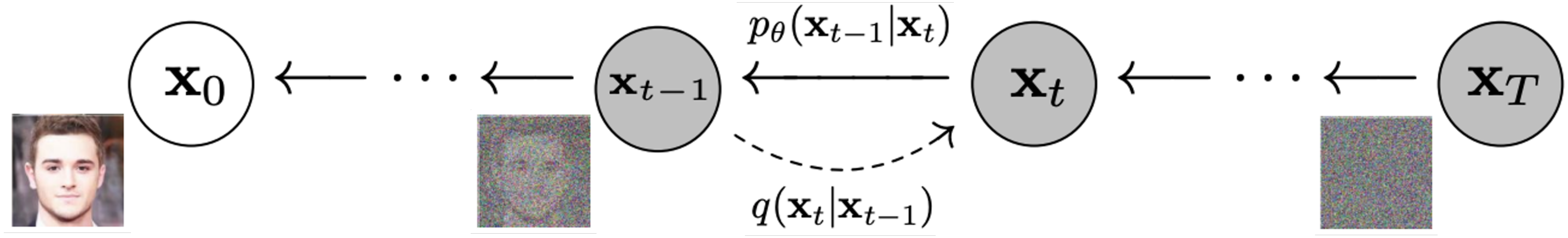
$$\alpha_t \in [0, 1]$$

$$q(x_t \,|\, x_0) = \mathcal{N}(x_t \,|\, \sqrt{\bar{\alpha}_t} x_{t-1}, (1-\bar{\alpha}_t)I)$$

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

$$p(x_{t-1} \,|\, x_t, x_0) \propto q(x_t \,|\, x_{t-1}) q(x_{t-1} \,|\, x_0) = \mathcal{N}(x_{t-1} \,|\, \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I)$$

Diffusion models



$$q(x_t \,|\, x_{t-1}) = \mathcal{N}(x_t \,|\, \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t)I) \qquad \alpha_t \in [0,1]$$

$$q(x_t \,|\, x_0) = \mathcal{N}(x_t \,|\, \sqrt{\bar{\alpha}_t} x_{t-1}, (1-\bar{\alpha}_t)I) \qquad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

$$p(x_{t-1} \,|\, x_t, x_0) \propto q(x_t \,|\, x_{t-1}) q(x_{t-1} \,|\, x_0) = \mathcal{N}(x_{t-1} \,|\, \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I)$$

$$\tilde{\beta}_t = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \qquad \tilde{\mu}_t(x_t, x_0) = \frac{1}{\bar{\alpha}_t}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_t\right)$$

DM training and sampling

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 $\mathbf{x}_0$

Is is possible to predict $\tilde{\mu}_t$ directly or x_0

However, for images prediction of ε_t works better

Why is it hard to apply DMs to text data?

Texts are **discrete**!

It is not obvious how to add noise to texts

Why is it hard to apply DMs to text data?

Texts are **discrete**!

It is not obvious how to add noise to texts

Two approaches:

- **Discrete diffusion** – destroy information by replacing one tokens with others
- **Continuous diffusion** – map text into continuous space and perform diffusion there

Both approaches are actively developing.
Time will show, which one wins

Discrete Diffusion

Idea: Introduce stochastic matrix Q_t that sets token change probabilities

$$Q_t[i, j] = p(x_t = j \,|\, x_{t-1} = i)$$

Then we the forward process becomes

$$q(x_t \,|\, x_{t-1}) = \text{Cat}(x_t \,|\, p = x_{t-1} Q_t)$$

x_t here is a one-hot vector

Examples of Q_t

Uniform: interpolation between data and uniform distributions

$$Q_t = (1 - \beta_t)I + \beta_t \frac{1}{|V|} 11^T$$

T = 0	The great brown fox hopped over the lazy dog.
T = 10	The vast black fox hopping over the lazy cat.
T = 20	Their vast tripped this jumping upon walked organizations.
T = 25	Bunk scamper tripped this Sanchez walked organizations.

Examples of Q_t

Absorbing: all tokens degrade to the [MASK] (m) token

$$[Q_t]_{ij} = \begin{cases} 1, & i = j = m \\ 1 - \beta_t, & i = j \neq m \\ \beta_t, & i = m, j = m \end{cases}$$

T = 0	The great brown fox hopped over the lazy dog.
T = 10	The great [MASK] fox hopped over [MASK] lazy dog.
T = 20	The [MASK][MASK] [MASK] ship over [MASK] lazy the.
T = 25	[MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK]

Discrete Diffusion training

To get the loss function we need to maximize the likelihood $p(x_{t-1} | x_t, x_0)$

$$p(x_{t-1} | x_t, x_0) \propto q(x_t | x_{t-1}) q(x_{t-1} | x_0)$$

Discrete Diffusion training

To get the loss function we need to maximize the likelihood $p(x_{t-1} | x_t, x_0)$

$$p(x_{t-1} | x_t, x_0) \propto q(x_t | x_{t-1}) q(x_{t-1} | x_0)$$

or to maximize the variational lower bound (VLB), which is the same

$$\log p_\theta(x_0) = \log \int q(x_{1:T} | x_0) \frac{p_\theta(x_{0:T})}{q(x_{1:T} | x_0)} dx_{1:T} \geq \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log p_\theta(x_{0:T}) - \log q(x_{1:T} | x_0)\right]$$

Discrete Diffusion training

To get the loss function we need to maximize the likelihood $p(x_{t-1} | x_t, x_0)$

$$p(x_{t-1} | x_t, x_0) \propto q(x_t | x_{t-1}) q(x_{t-1} | x_0)$$

or to maximize the variational lower bound (VLB), which is the same

$$\log p_\theta(x_0) = \log \int q(x_{1:T} | x_0) \frac{p_\theta(x_{0:T})}{q(x_{1:T} | x_0)} dx_{1:T} \geq \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log p_\theta(x_{0:T}) - \log q(x_{1:T} | x_0) \right]$$

However, in practice often a simple cross entropy loss is used

$$L_\theta = - \mathbb{E}_{q(x_t|x_0)} \left[\log p_\theta \left(x_0 | x_t \right) \right]$$

Concrete Score Matching

Let $\mathcal{N}(x)$ be a neighbourhood of x, $\mathcal{N}(x) = \{x_{n_1}, \ldots, x_{n_k}\}$

Then *concrete score* is

$$c_p(x; \mathcal{N}) = \left[\frac{p(x_{n_1})}{p(x)}, \ldots, \frac{p(x_{n_k})}{p(x)}\right] - 1$$

https://arxiv.org/pdf/2310.16834

Concrete Score Matching

Let $\mathcal{N}(x)$ be a neighbourhood of x, $\mathcal{N}(x) = \{x_{n_1}, \ldots, x_{n_k}\}$

Then *concrete score* is

$$c_p(x; \mathcal{N}) = \left[\frac{p(x_{n_1})}{p(x)}, \ldots, \frac{p(x_{n_k})}{p(x)}\right] - 1$$

Proposition: For $x \in \mathbb{R}^d$ and $\delta > 0$ let $\mathcal{N}_\delta = \{x + \delta \mathbf{e}_i\}_{i=1}^d$. Then we have

$$\lim_{\delta \to 0} \frac{c_p\left(x; \mathcal{N}_\delta\right)}{\delta} = \nabla_x \log p(x)$$

Proof:

$$\lim_{\delta \to 0} \left\{ \frac{p(x + \delta \mathbf{e}_i) - p(x)}{\delta \cdot p(x)} \right\}_{i=1}^d = \frac{1}{p(x)} \nabla_x p(x)$$

https://arxiv.org/pdf/2310.16834

Concrete Score Matching

Turns out that is it much better to predict the ratio of probability densities.

$$s_\theta(x,t) \approx \left[\frac{p_t(y)}{p_t(x)}\right]_{x \neq y}$$

For optimization we can use, for example, MSE

$$L_{\text{CSM}} = \frac{1}{2}\mathbb{E}_{x\sim p_t}\left[\sum_{y\neq x}^{|V|}\left(s_\theta(x_t,t)_y - \frac{p_t(y)}{p_t(x)}\right)^2\right]$$

https://arxiv.org/pdf/2310.16834

Concrete Score Matching

Size	Model	LAMBADA	WikiText2	PTB	WikiText103	1BW
Small	GPT-2	**45.04**	42.43	138.43	41.60	**75.20**
	SEDD Absorb	$\leq$50.92	$\leq$**41.84**	$\leq$**114.24**	$\leq$**40.62**	$\leq$79.29
	SEDD Uniform	$\leq$65.40	$\leq$50.27	$\leq$140.12	$\leq$49.60	$\leq$101.37
	D3PM	$\leq$93.47	$\leq$77.28	$\leq$200.82	$\leq$75.16	$\leq$138.92
	PLAID	$\leq$57.28	$\leq$51.80	$\leq$142.60	$\leq$50.86	$\leq$91.12
Medium	GPT-2	**35.66**	31.80	123.14	31.39	**55.72**
	SEDD Absorb	$\leq$42.77	$\leq$**31.04**	$\leq$**87.12**	$\leq$**29.98**	$\leq$61.19
	SEDD Uniform	$\leq$51.28	$\leq$38.93	$\leq$102.28	$\leq$36.81	$\leq$79.12

Zero-shot unconditional perplexity (↓) on a variety of datasets

https://arxiv.org/pdf/2310.16834