Diffusion Models for Discrete Data

Aaron Lou

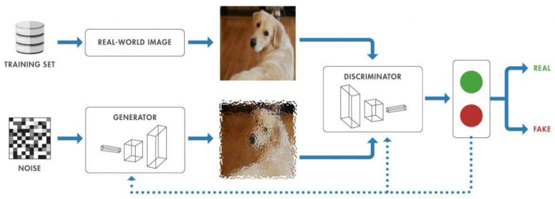
Introduction

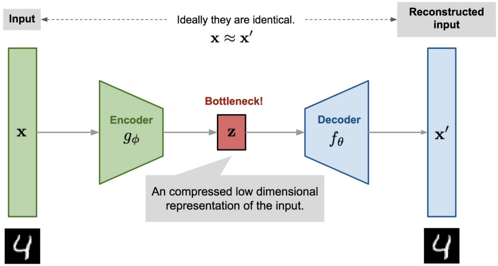
Dataset $\{x_1, x_2, \ldots, x_n\} \sim p_{\text{data}}$

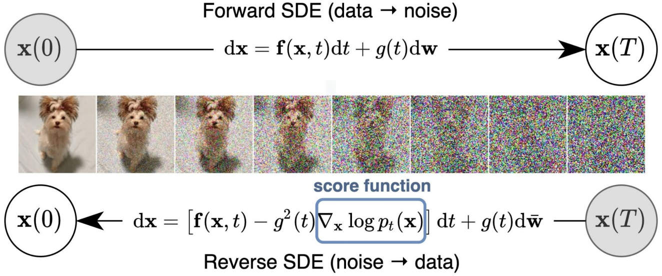
Learn $p_\theta \approx p_{\mathrm{d}ata}$

Generate samples using p_θ

Continuous vs Discrete Data



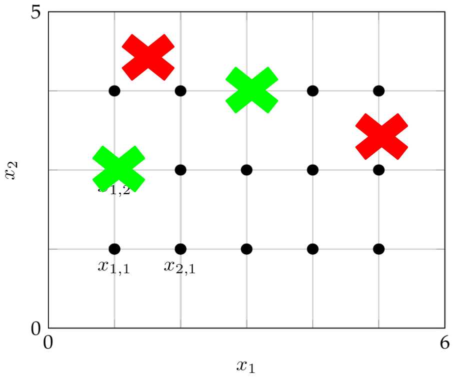

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
$\mathbf{x}$
Encoder g_ϕ
Bottleneck!
$\mathbf{z}$
Decoder f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.


Forward SDE (data → noise)
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$
$\mathbf{x}(T)$
score function
$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_\mathbf{x} \log p_t(\mathbf{x})}]\,\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$
Reverse SDE (noise → data)

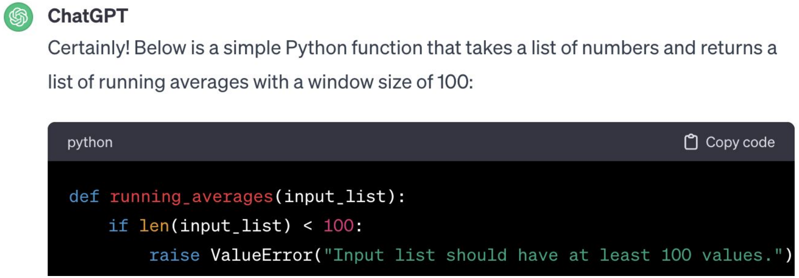
$$\mathcal{X} = \mathbb{R}^d$$

$$\mathcal{X} = \{1, \ldots, N\}^d$$

$$\mathbf{x} = x^1 \ldots x^d$$



Why Discrete Data?



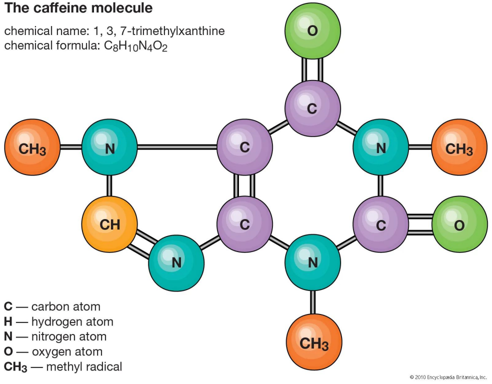
ChatGPT

Certainly! Below is a simple Python function that takes a list of numbers and returns a list of running averages with a window size of 100:

```python
def running_averages(input_list):
    if len(input_list) < 100:
        raise ValueError("Input list should have at least 100 values.")
```

Language Model “Pretaining”: fitting a discrete probabilistic model to data.

Why Discrete Data?

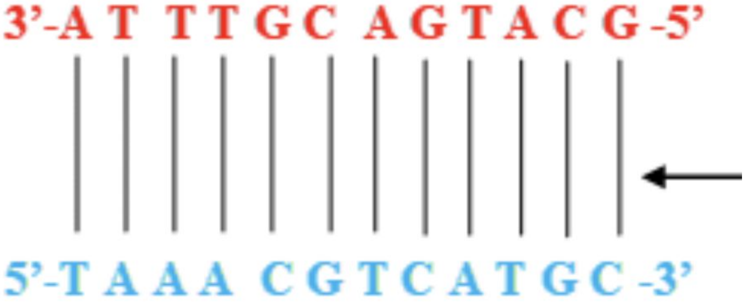


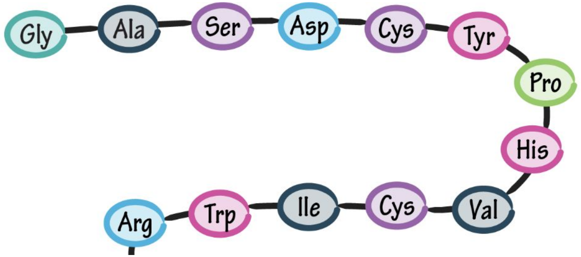
The caffeine molecule

chemical name: 1, 3, 7-trimethylxanthine
chemical formula: $C_8H_{10}N_4O_2$

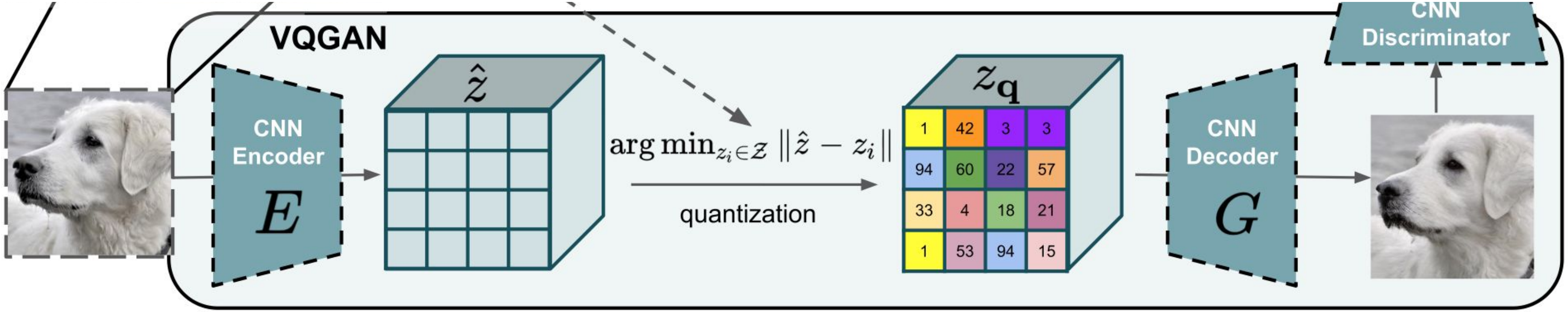
C — carbon atom
H — hydrogen atom
N — nitrogen atom
O — oxygen atom
CH_3 — methyl radical

© 2010 Encyclopædia Britannica, Inc.



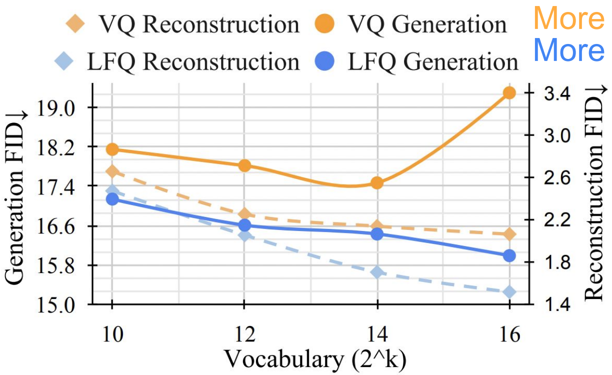


Why Discrete Data?



1	42	3	3
94	60	22	57
33	4	18	21
1	53	94	15

VQVAE backbone

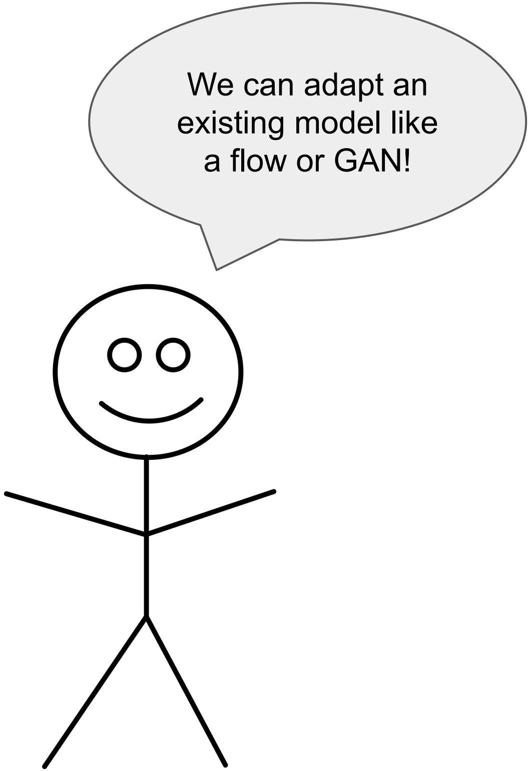


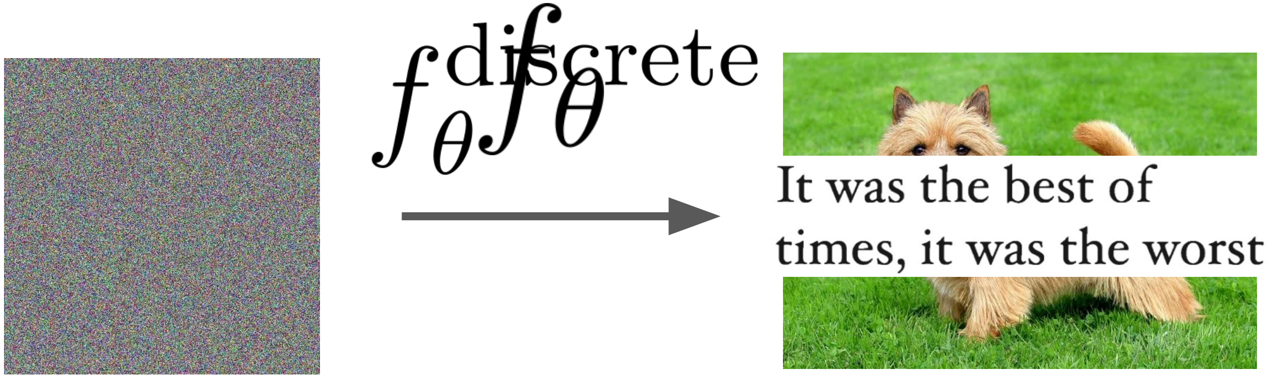
More Continuous

More Discrete

From recent Google/CMU MAGVIT-v2 paper

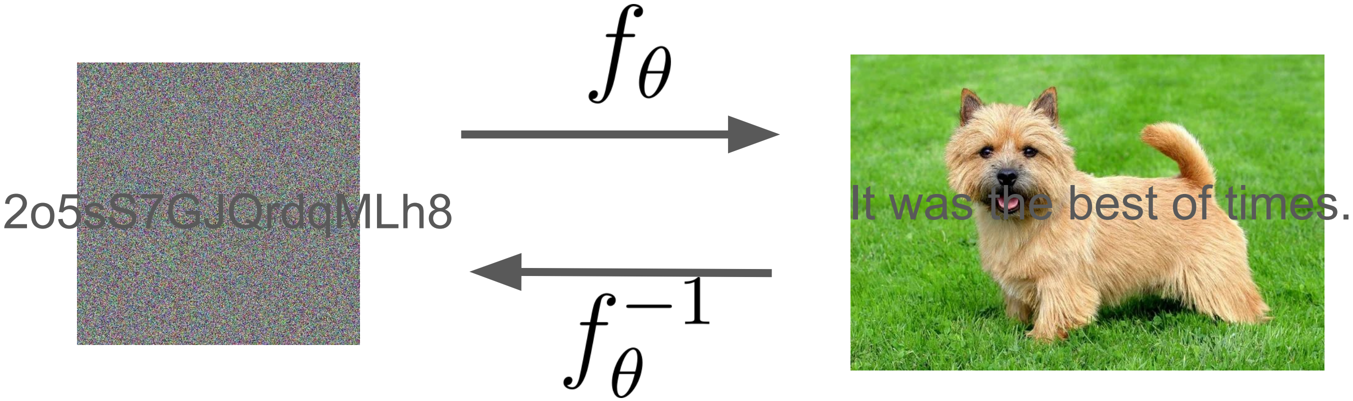
What makes Discrete Data Hard?





It was the best of times, it was the worst

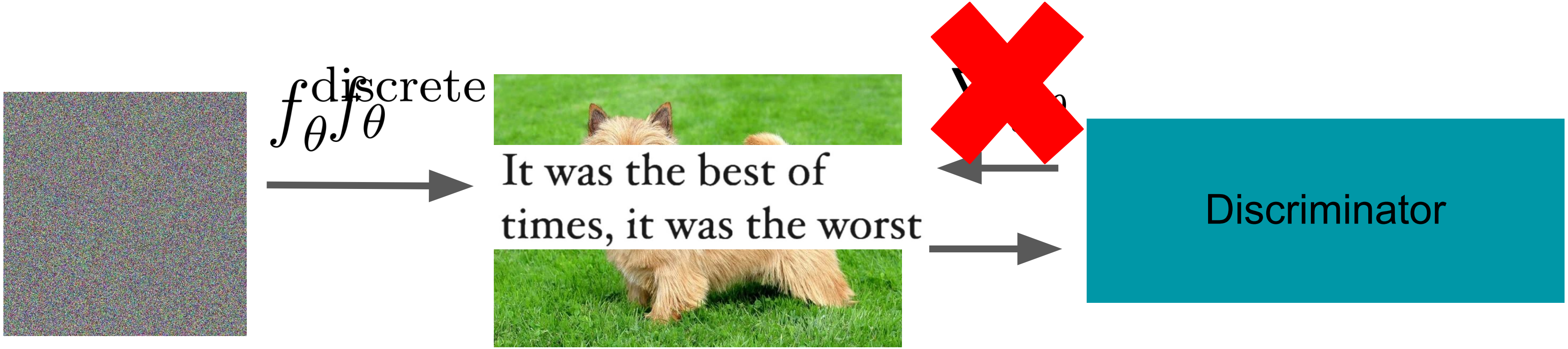
What makes Discrete Data Hard?



$$p_{\text{data}}(x) = p_{\text{base}}(f_\theta^{-1}(x)) \, |\det D_x f_\theta^{-1}|$$

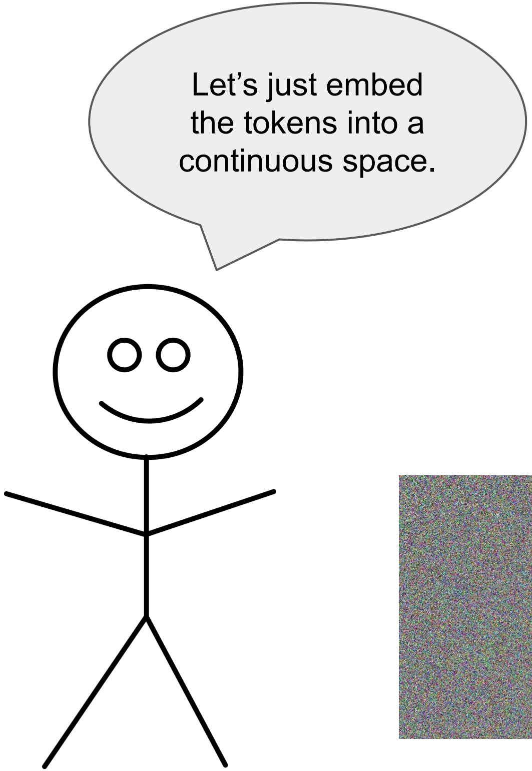
$$p_{\text{data}}(x) = p_{\text{base}}(f_\theta^{-1}(x))$$

What makes Discrete Data Hard?

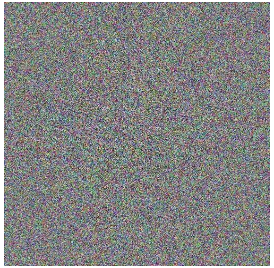


Conclusion: our models are too reliant on calculus!

What makes Discrete Data Hard?



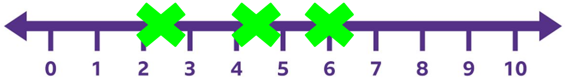
157	153	174	168	150
155	182	163	74	75
180	180	50	14	34
206	109	5	124	131
194	68	137	251	237



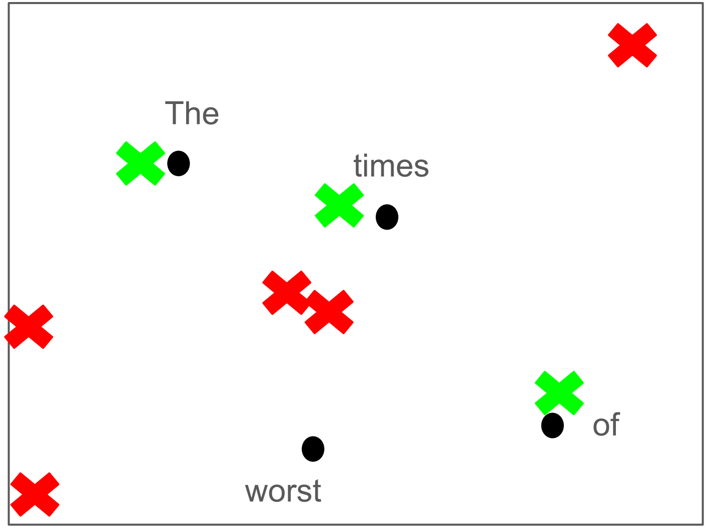
$\xrightarrow{f_\theta}$ Continuous Image $\xrightarrow{\text{Discretize}}$ Discrete Image

What makes Discrete Data Hard?

$$[0, 1, \ldots, 255]$$

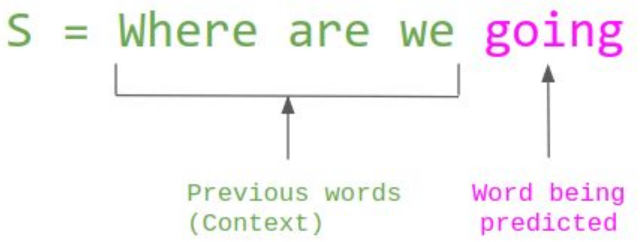


$$[\text{The}, \text{times}, \text{worst}, \text{of}, \ldots]$$



Best Approach So Far: Autoregressive Modeling

$$\begin{aligned} p_\theta(\mathbf{x}) &= p_\theta(x^1 x^2 \ldots x^d) \\ &= p_\theta(x^1) p_\theta(x^2 | x^1) \ldots p_\theta(x^d | x^1 x^2 \ldots x^{d-1}) \end{aligned}$$



S = Where are we going

Previous words (Context)

Word being predicted

Autoregressive Modeling - Upsides

✓ Scalable - each component is only a probability over D values

✓ Can theoretically represent any probability vector

✓ Reasonably inductive bias for language

Autoregressive Modeling - Downsides

- ✗ Sampling “drifts” - Yann LeCun
- ✗ Not a reasonable bias for non-language tasks
- ✗ Constrained architectures
- ✗ Slow sampling due to iterative nature

Rethinking the Problem with Score Matching

Problem: modeling $p_\theta(\mathbf{x})$ is extremely hard since we must sum to 1.

? ? ?

? ? ?

$$\nabla_x \log p$$

?

$$\min_\theta \|s_\theta(x) - \nabla_x \log p\|^2$$

? ? ?

$$dx = (f(x,t) - g(t)^2 \nabla_x \log p_t)dt + g(t)dB_t$$

?? ? ? ? ? ? ?

Outline

- Score matching on discrete spaces
- Sampling using the “concrete scores”
- Evaluating likelihoods of the generative process

Outline

- Score matching on discrete spaces
- Sampling using the “concrete scores”
- Evaluating likelihoods of the generative process

Concrete Score

$$\nabla f(x) \equiv [f(y) - f(x)]_{y \text{ neighbor of } x}$$

$$\nabla_x \log p = \frac{\nabla p(x)}{p(x)} = \boxed{\left[\frac{p(y)}{p(x)}\right]_y} - 1$$

“Concrete Score”

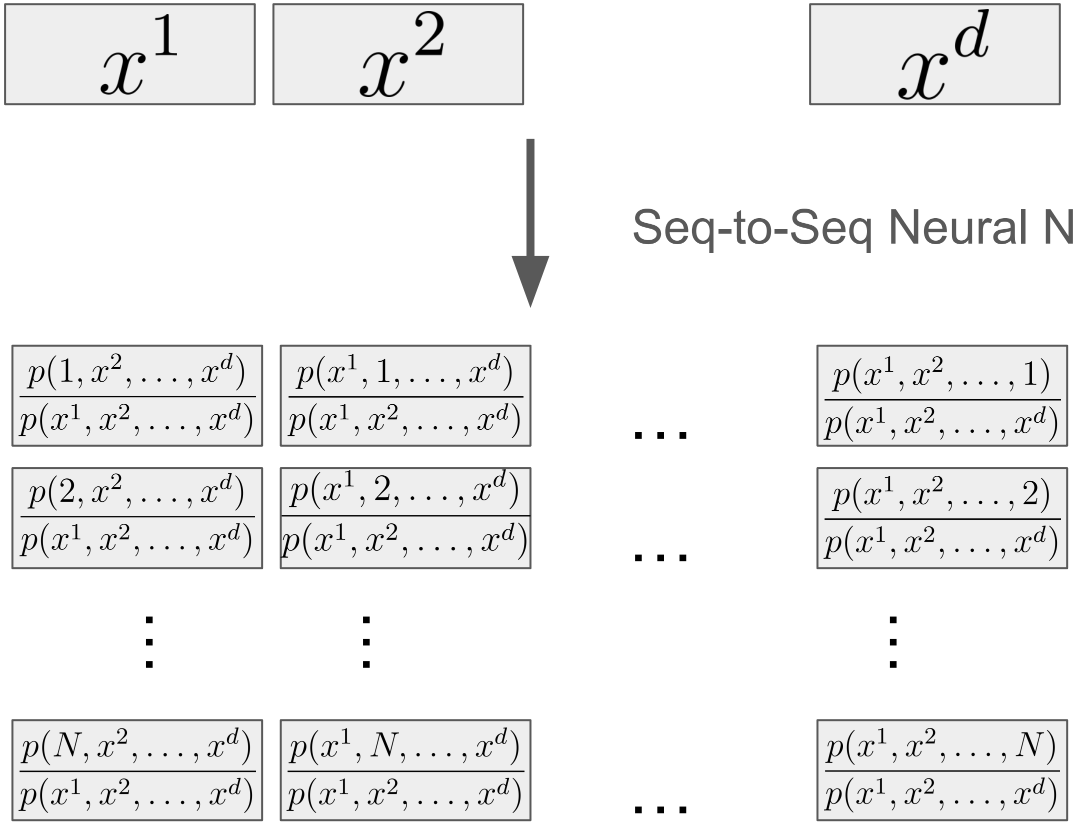
Concrete Score - Example

$$\frac{p(y)}{p(x)}$$ is way too many ratios: $$O(N^{2d})$$

$$\frac{p(x^1 \dots \widehat{x}^i \dots x^d)}{p(x^1 \dots x^i \dots x^d)}$$ is better since local: $$O(Nd)$$

We’ll write it out assuming 1 dimension (generalization is easy).

Concrete Score - Example



x^1 x^2 x^d

Seq-to-Seq Neural Network

$\frac{p(1, x^2, \ldots, x^d)}{p(x^1, x^2, \ldots, x^d)}$	$\frac{p(x^1, 1, \ldots, x^d)}{p(x^1, x^2, \ldots, x^d)}$	$\ldots$	$\frac{p(x^1, x^2, \ldots, 1)}{p(x^1, x^2, \ldots, x^d)}$
$\frac{p(2, x^2, \ldots, x^d)}{p(x^1, x^2, \ldots, x^d)}$	$\frac{p(x^1, 2, \ldots, x^d)}{p(x^1, x^2, \ldots, x^d)}$	$\ldots$	$\frac{p(x^1, x^2, \ldots, 2)}{p(x^1, x^2, \ldots, x^d)}$
$\vdots$	$\vdots$		$\vdots$
$\frac{p(N, x^2, \ldots, x^d)}{p(x^1, x^2, \ldots, x^d)}$	$\frac{p(x^1, N, \ldots, x^d)}{p(x^1, x^2, \ldots, x^d)}$	$\ldots$	$\frac{p(x^1, x^2, \ldots, N)}{p(x^1, x^2, \ldots, x^d)}$

Learning Concrete Scores with Score Entropy

Goal: learn a neural network $s_\theta(x)$ s.t. $s_\theta(x)_y \approx \frac{p(y)}{p(x)}$

Needs to be principled (doesn’t allow negative values, recovers true value)

$$\min_\theta \mathbb{E}_{x \sim p} \sum_{y \neq x} s_\theta(x)_y - \frac{p(y)}{p(x)} \log s_\theta(x)_y$$

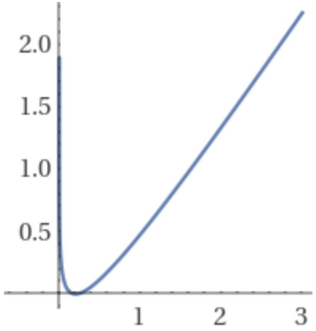
Learning Concrete Scores with Score Entropy

$$\min \left(s - \frac{p(y)}{p(x)} \log s \right)$$

$$\implies (s - \frac{p(y)}{p(x)} \log s)' = 0$$

$$\implies 1 - \frac{p(y)}{p(x)} \frac{1}{s} = 0$$

$$\implies s = \frac{p(y)}{p(x)}$$



Independently minimizes for all pairs of x, y.

Score Entropy is Intractable

$$\mathbb{E}_{x \sim p} \sum_{y \neq x} s_\theta(x)_y - \boxed{\frac{p(y)}{p(x)}} \log s_\theta(x)_y$$

1. Implicit Score Entropy - analogous to implicit score matching.

2. Denoising Score Entropy - analogous to denoising score matching.

Implicit Score Entropy

$$\mathbb{E}_{x\sim p}\sum_{y\neq x} s_\theta(x)_y - \boxed{\frac{p(y)}{p(x)}} \log s_\theta(x)_y$$

$$\mathbb{E}_{x\sim p}\sum_{y\neq x} \frac{p(y)}{p(x)} \log s_\theta(x)_y = \sum_x \sum_{y\neq x} p(y) \log s_\theta(x)_y$$

$$= \boxed{\mathbb{E}_{y\sim p}\sum_{x\neq y} \log s_\theta(x)_y}$$

Removed ratios, swapped x and y

$$\underbrace{\mathbb{E}_{x\sim p}\sum_{y\neq x} s_\theta(x)_y - \frac{p(y)}{p(x)} \log s_\theta(x)_y}_{\text{Score Entropy}} = \underbrace{\mathbb{E}_{x\sim p}\sum_{y\neq x} s_\theta(x)_y - \log s_\theta(y)_x}_{\text{Implicit Score Entropy}}$$

Implicit Score Entropy - Scalability

$$\mathbb{E}_{\boxed{x\sim p}} \sum_{y\neq x} \boxed{s_\theta(x)_y} - \log \boxed{s_\theta(y)_x}$$

Sampled

Evaluate $s_\theta(x)$ once and index for all y

Need to evaluate all $s_\theta(y)$

Denoising Score Entropy

Assume $p(x) = \sum_{x_0} p(x|x_0)p_0(x_0)$

$$\mathbb{E}_{x\sim p} \sum_{y\neq x} \frac{p(y)}{p(x)} \log s_\theta(x)_y = \sum_{x} \sum_{y\neq x} \log s_\theta(x)_y p(y)$$

$$= \sum_{x} \sum_{y\neq x} \log s_\theta(x)_y \sum_{x_0} p(y|x_0)p_0(x_0)$$

$$= \sum_{x_0} \sum_{x} \sum_{y\neq x} \log s_\theta(x)_y \frac{p(y|x_0)}{p(x|x_0)} p(x|x_0)p_0(x_0)$$

$$= \mathbb{E}_{x_0\sim p_0, x\sim p(\cdot|x_0)} \sum_{y\neq x} \boxed{\frac{p(y|x_0)}{p(x|x_0)}} \log s_\theta(x)_y$$

Denoising Score Entropy - Scalability

$$\mathbb{E}_{\boxed{x_0 \sim p_0}, \boxed{x \sim p(\cdot|x_0)}} \sum_{y \neq x} \boxed{s_\theta(x)_y} - \boxed{\frac{p(y|x_0)}{p(x|x_0)}} \log s_\theta(x)_y$$

Sampled

Sampled

Compute $s_\theta(x)$ once

Computable

Outline

- Score matching on discrete spaces

- Sampling using the “concrete scores”

- Evaluating likelihoods of the generative process

Continuous Time Markov Chains

Diffusion is just an evolution of $\boldsymbol{p_t} \in \mathbb{R}^{|\mathcal{X}|}$

$$dp_t = Q_t p_t$$

1. Columns of Q_t must sum to 0.
2. Non-diagonal entries of Q_t are ≥ 0

Continuous Time Markov Chains

Q_t controls how often one goes to other states.

$$p(x_{t+\Delta t} = j | x_t = i) = \delta_{i,j} + Q_t(j, i)\Delta t + O(\Delta t^2)$$

Jump
transition rate
from i to j.

Continuous Time Markov Chains - Examples

$$\underbrace{\begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}}_{Q_t} \underbrace{\begin{bmatrix} 0.5 \\ 0.2 \\ 0.3 \end{bmatrix}}_{p_0} = \begin{bmatrix} -0.5 \\ 0.1 \\ 0.4 \end{bmatrix}$$

$$p_t = \exp\left(t \begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \right) \begin{bmatrix} 0.5 \\ 0.3 \\ 0.2 \end{bmatrix}$$

Can check that the transition satisfies the statement.

Continuous Time Markov Chains - Examples

$$Q_t = \sigma(t)Q$$

“Linear ODE”

$$p_t = \boxed{\exp(\Sigma(t)Q)}p_0$$

Many methods to compute this matrix exponential (e.g. eigenvalues), but simpler is better.

$$p(x_t = j | x_0 = i) = \exp(\Sigma(t)Q)(j, i)$$

$$t \to \infty \quad p_t \longrightarrow p_{\text{base}}$$

Continuous Time Markov Chains - Examples

$$\begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \quad \begin{bmatrix} \frac{1}{3}+\frac{2}{3}e^{-3t} & \frac{1}{3}-\frac{1}{3}e^{-3t} & \frac{1}{3}-\frac{1}{3}e^{-3t} \\ \frac{1}{3}-\frac{1}{3}e^{-3t} & \frac{1}{3}+\frac{2}{3}e^{-3t} & \frac{1}{3}-\frac{1}{3}e^{-3t} \\ \frac{1}{3}-\frac{1}{3}e^{-3t} & \frac{1}{3}-\frac{1}{3}e^{-3t} & \frac{1}{3}+\frac{2}{3}e^{-3t} \end{bmatrix} \quad x \to \mathrm{random}$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} e^{-t} & 0 & 0 & 0 \\ 0 & e^{-t} & 0 & 0 \\ 0 & 0 & e^{-t} & 0 \\ 1-e^{-t} & 1-e^{-t} & 1-e^{-t} & 0 \end{bmatrix} \quad x \to \mathrm{MASK}$$

Continuous Time Markov Chains - Examples

1. Perturb sequence by sequence

$$x^1 \dots x^d \to y^1 \dots y^d \qquad \text{✖} \; O(\text{[illegible]}^{2d})$$

2. Perturb tokens independently with same matrix.

$$x^1 \dots x^i \dots x^d \to x^1 \dots \widehat{x}^i \dots x^d \qquad O(d^2)$$

$$p(y^1 \dots y^d | x^1 \dots x^d) = \prod_{i=1}^{d} p(y^i | x^i)$$

Continuous Time Markov Chains + Score Entropy

Assume samples from $x_0 \sim p_0$

Can we learn $s_\theta(x,t)_y \approx \frac{p_t(y)}{p_t(x)}$?

$$\mathbb{E}_{t,x_0\sim p_0,x_t\sim p_t(\cdot|x_0)} \sum_{y\neq x} s_\theta(x_t,t)_y - \boxed{\frac{p_t(y|x_0)}{p_t(x|x_0)}} \log s_\theta(x_t,t)_y$$

Given by Q_t

Reversing a Markov Chain

Assume we perturb from $p_0 \approx p_{\text{data}}$ to $p_T \approx p_{\text{base}}$

Can we go from $p_T \approx p_{\text{base}}$ to $p_0 \approx p_{\text{data}}$?

$$dp_{T-t} = \overline{Q}_{T-t} p_{T-t}$$

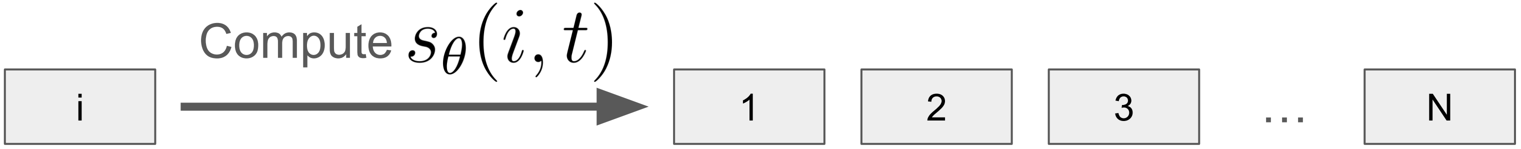
$$\overline{Q}_t(j, i) = \frac{p_t(j)}{p_t(i)} Q_t(i, j)$$

Diagonal values normalized so that the matrix is a valid diffusion matrix.

Reverse Markov Chains + Concrete Scores

$$\overline{Q}_t(i, j) = \boxed{\frac{p_t(j)}{p_t(i)}} Q_t(j, i)$$

$$\overline{Q}_t(j, i) \approx s_\theta(i, t)_j Q_t(i, j)$$



Compute $s_\theta(i, t)$

i

1

2

3

…

N

Reversing a Markov Chain - Examples

$$\begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.1 \\ 0.4 \end{bmatrix}$$

$$\begin{bmatrix} -1 & \frac{5}{3} & \frac{5}{2} \\ \frac{3}{5} & -\frac{7}{3} & \frac{3}{2} \\ \frac{2}{5} & \frac{2}{3} & -4 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.1 \\ -0.4 \end{bmatrix}$$

Reversing a Markov Chain - Examples

study ants bear burrito Stanford song

MASK MASK MASK MASK MASK MASK

Accelerating Sampling with Discretization

Problem: reverse is very slow!

$$x^1 \dots x^i \dots x^d \to x^1 \dots \widehat{x}^i \dots x^d$$

Only one token can change at a time.

Solution: allow multiple steps.

It was the MASK of MASK -> It was the best of times

Putting it all together

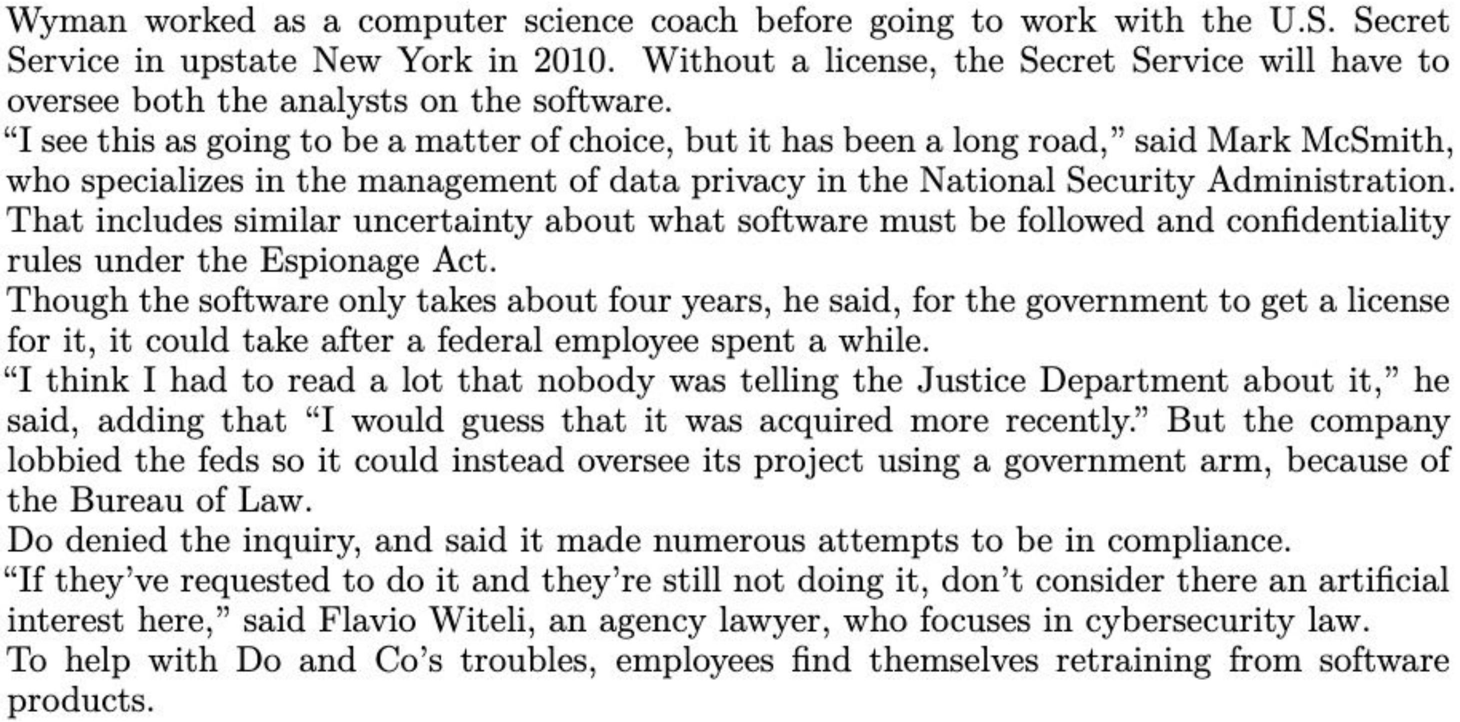
1. Get samples from desired data distribution

2. Define a forward diffusion process

3. Learn ratios using Score Entropy

4. Reverse diffusion process (possibly with some discretization).

Putting it all together



Wyman worked as a computer science coach before going to work with the U.S. Secret Service in upstate New York in 2010. Without a license, the Secret Service will have to oversee both the analysts on the software.

“I see this as going to be a matter of choice, but it has been a long road,” said Mark McSmith, who specializes in the management of data privacy in the National Security Administration. That includes similar uncertainty about what software must be followed and confidentiality rules under the Espionage Act.

Though the software only takes about four years, he said, for the government to get a license for it, it could take after a federal employee spent a while.

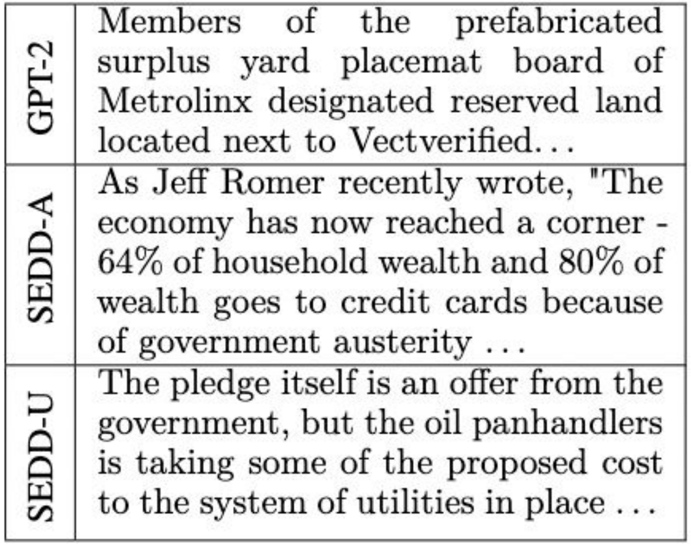
“I think I had to read a lot that nobody was telling the Justice Department about it,” he said, adding that “I would guess that it was acquired more recently.” But the company lobbied the feds so it could instead oversee its project using a government arm, because of the Bureau of Law.

Do denied the inquiry, and said it made numerous attempts to be in compliance.

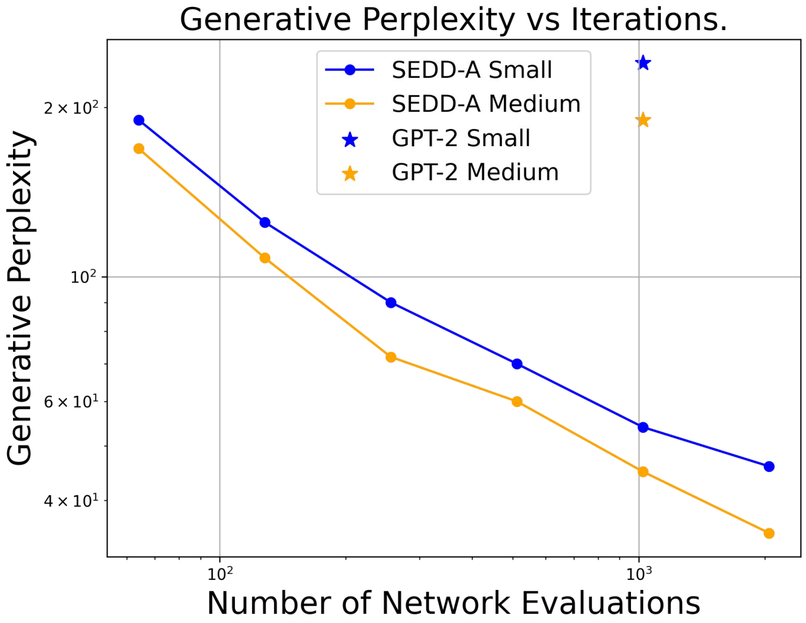
“If they’ve requested to do it and they’re still not doing it, don’t consider there an artificial interest here,” said Flavio Witeli, an agency lawyer, who focuses in cybersecurity law.

To help with Do and Co’s troubles, employees find themselves retraining from software products.

Putting it all together

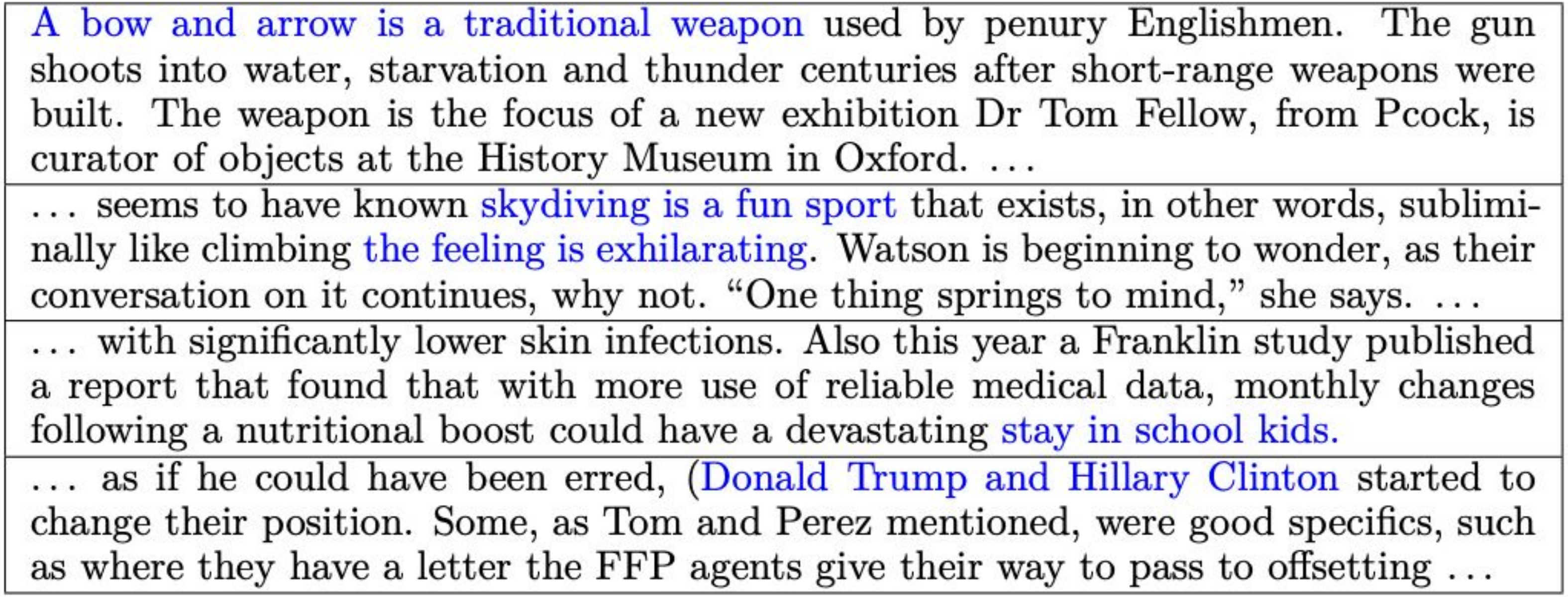


GPT-2	Members of the prefabricated surplus yard placemat board of Metrolinx designated reserved land located next to Vectverified. . .
SEDD-A	As Jeff Romer recently wrote, "The economy has now reached a corner - 64% of household wealth and 80% of wealth goes to credit cards because of government austerity . . .
SEDD-U	The pledge itself is an offer from the government, but the oil panhandlers is taking some of the proposed cost to the system of utilities in place . . .



Surpasses autoregressive transformers for generation quality/speed!

Conditional Generation (Prompt Infilling)



A bow and arrow is a traditional weapon used by penury Englishmen. The gun shoots into water, starvation and thunder centuries after short-range weapons were built. The weapon is the focus of a new exhibition Dr Tom Fellow, from Pcock, is curator of objects at the History Museum in Oxford. . . .

. . . seems to have known skydiving is a fun sport that exists, in other words, subliminally like climbing the feeling is exhilarating. Watson is beginning to wonder, as their conversation on it continues, why not. "One thing springs to mind," she says. . . .

. . . with significantly lower skin infections. Also this year a Franklin study published a report that found that with more use of reliable medical data, monthly changes following a nutritional boost could have a devastating stay in school kids.

. . . as if he could have been erred, (Donald Trump and Hillary Clinton started to change their position. Some, as Tom and Perez mentioned, were good specifics, such as where they have a letter the FFP agents give their way to pass to offsetting . . .

Outline

- Score matching on discrete spaces
- Sampling using the “concrete scores”
- Evaluating likelihoods of the generative process

Perplexity

$$PPL(x) = e^{-\frac{1}{d}\log p_\theta(x^1 ... x^d)}$$

✓ Principled measurement of model ability

✓ Directly computable for autoregressive modeling

✓ Optimized w/ standard cross entropy loss

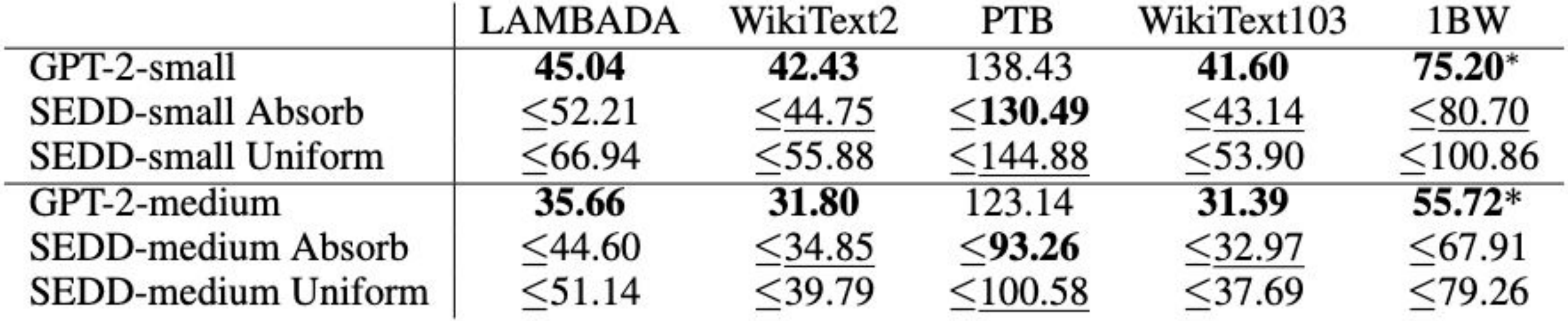
Computing Likelihood Bounds

$$-\log p_\theta(x_0) \leq \boxed{\int_0^t \mathbb{E}_{x_t \sim p_t(\cdot|x_0)} \sum_{y \neq x_t} Q_t(x_t, y) \left(s_\theta(x_t, t)_y - \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \log s_\theta(x_t, t)_y \right) dt} + C$$

(Weighted) version of score entropy.

$$PPL(x) \leq e^{-\frac{1}{d} DSE(x)}$$

Computing Likelihood Bounds



	LAMBADA	WikiText2	PTB	WikiText103	1BW
GPT-2-small	**45.04**	**42.43**	138.43	**41.60**	**75.20***
SEDD-small Absorb	≤52.21	≤44.75	≤**130.49**	≤43.14	≤80.70
SEDD-small Uniform	≤66.94	≤55.88	≤144.88	≤53.90	≤100.86
GPT-2-medium	**35.66**	**31.80**	123.14	**31.39**	**55.72***
SEDD-medium Absorb	≤44.60	≤34.85	≤**93.26**	≤32.97	≤67.91
SEDD-medium Uniform	≤51.14	≤39.79	≤100.58	≤37.69	≤79.26

Challenges autoregressive modeling on perplexities!

Summary

- It is hard to build probabilistic models for discrete space.
    - Autoregressive modeling has been (basically) the only paradigm
- Score based models extend to discrete spaces
    - Model the ratios of the data distribution (concrete scores)
    - Optimize Score Entropy loss (+ extensions)
- Sample using discrete diffusion processes
    - Synergizes with Denoising Score Entropy loss
    - Fast and controllable generation
    - Generation quality surpasses autoregressive models
- Score Entropy forms a likelihood bound.
    - Challenges autoregressive dominance