Diffusion Language Models

Zhihan Huang, Kevin Jiang

Wharton Statistics & Data Science
University of Pennsylvania

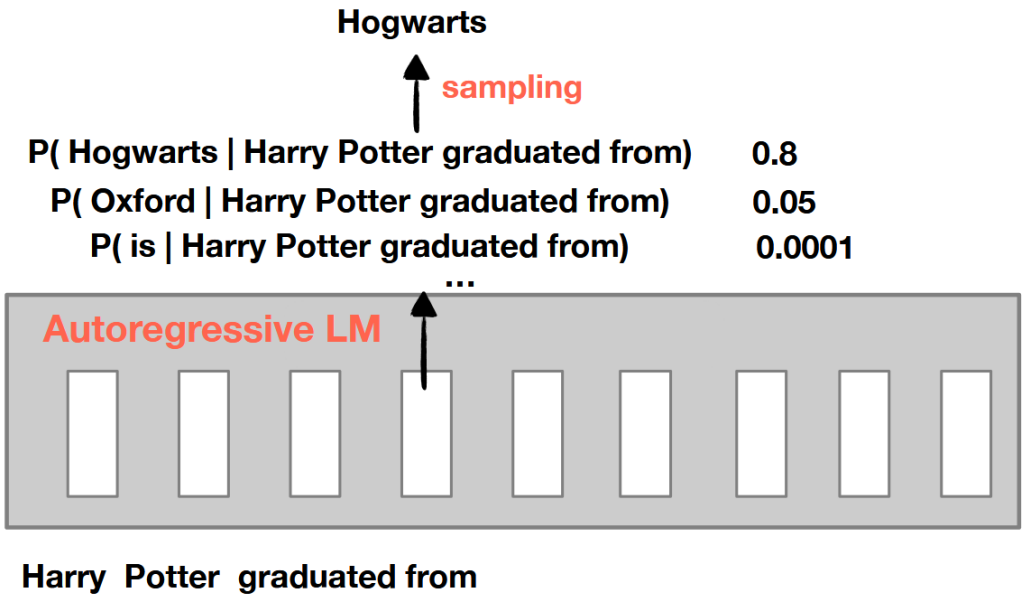
March 6, 2025

Wharton
University of Pennsylvania

Outline

Autoregressive LMs

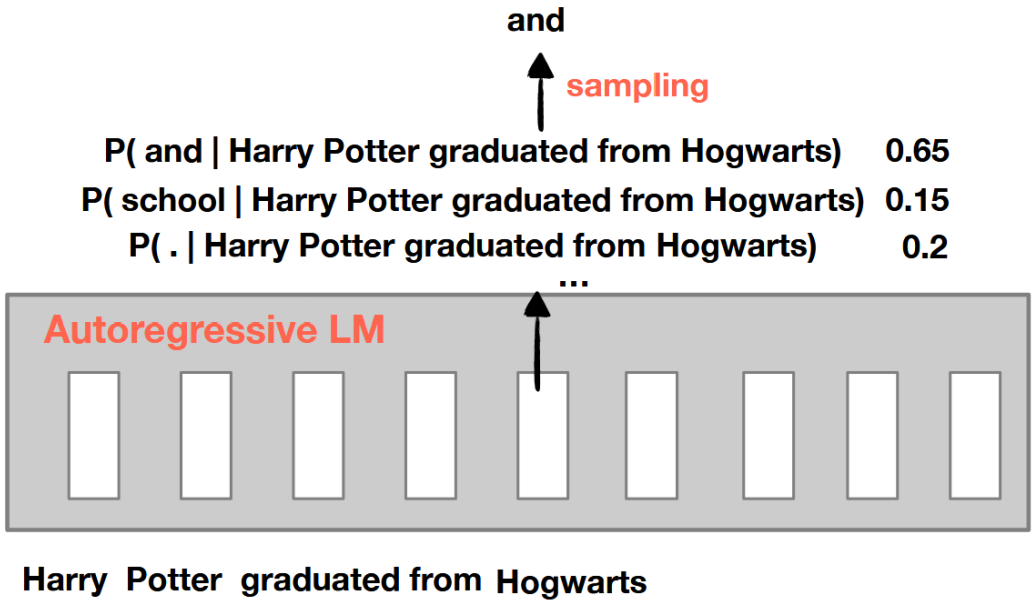
Generate text from the underlying distribution:



e.g., GPT-3

Autoregressive LMs

Generate text from the underlying distribution:



e.g., GPT-3

Next-token prediction: Generating text from left-to-right, one at a time.

— Parametrize the probability of a sentence via chain rule:
— For each token, compute the next-token distribution.

Wharton
University of Pennsylvania

Motivation

Some motivation for going beyond next-token prediction:

1. What if we want to generate right-to-left? Or given left and right context, fill in the middle text (inpainting)?

Motivation

Some motivation for going beyond next-token prediction:

1. What if we want to generate right-to-left? Or given left and right context, fill in the middle text (inpainting)?
2. For Autogressive models, time complexity is always O(L) where L is the length of generated text.
   — Can we do parallel sampling from a language model? Generate the whole senetence at once?

Motivation

Some motivation for going beyond next-token prediction:

1. What if we want to generate right-to-left? Or given left and right context, fill in the middle text (inpainting)?
2. For Autogressive models, time complexity is always O(L) where L is the length of generated text.
   — Can we do parallel sampling from a language model? Generate the whole senetence at once?

Can we model the text distribution by one joint distribution (instead of a sequential model) and sample from it: Diffusion Lnaguage Models (Diffusion LMs)

$$x^{1:L} \sim P_\theta(x)$$

Diffusion LMs

How to do it?

1. Learn an implicit probabilistic model P_θ from the data: a very high dimensional distribution in $\mathbb{R}^{L \cdot d}$ (d the embedding dimension, L the length of token)
2. Find an accurate and efficient way to sample from P_θ: discrete, low-dimensional manifold, non-smooth?

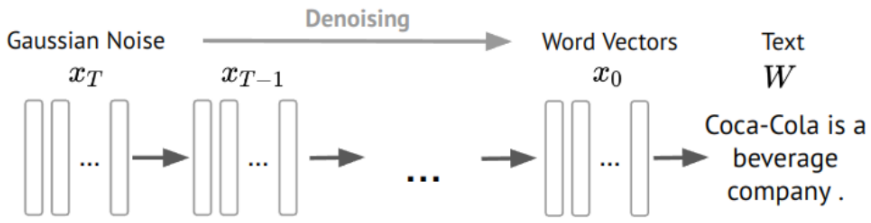
Diffusion LMs

How to do it?

1. Learn an implicit probabilistic model P_θ from the data: a very high dimensional distribution in $\mathbb{R}^{L \cdot d}$ (d the embedding dimension, L the length of token)
2. Find an accurate and efficient way to sample from P_θ: discrete, low-dimensional manifold, non-smooth?

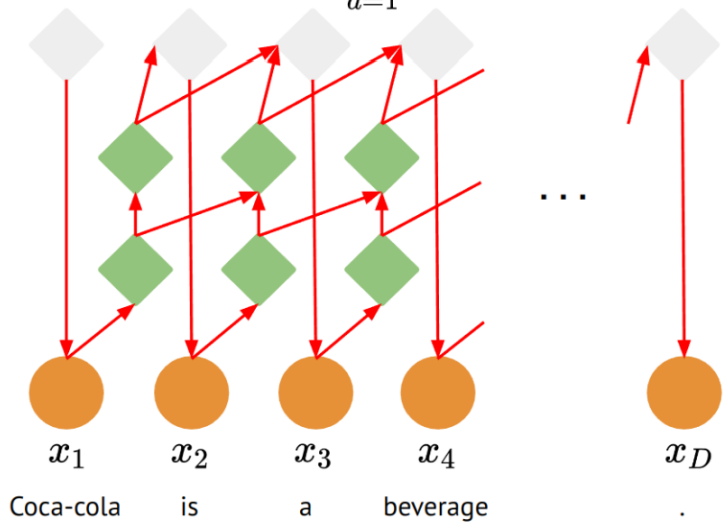
- Use Diffusion Model to model and sample from P_θ
- Diffusion LM is more like an experimental model exploring different architectures for tasks — no standard answers
- Very different from the autoregressive model — Can diffusion LMs be competitive with autoregressive (AR) models in some tasks?

Diffusion LMs



(a) Diffusion-based language model

$$p(x) = \prod_{d=1}^{D} p(x_d \mid x_{<d})$$



(b) AR language model

Wharton
University of Pennsylvania

What is a diffusion model?

Diffusion Models are successful in image generation

Wharton
University of Pennsylvania

What is a diffusion model?

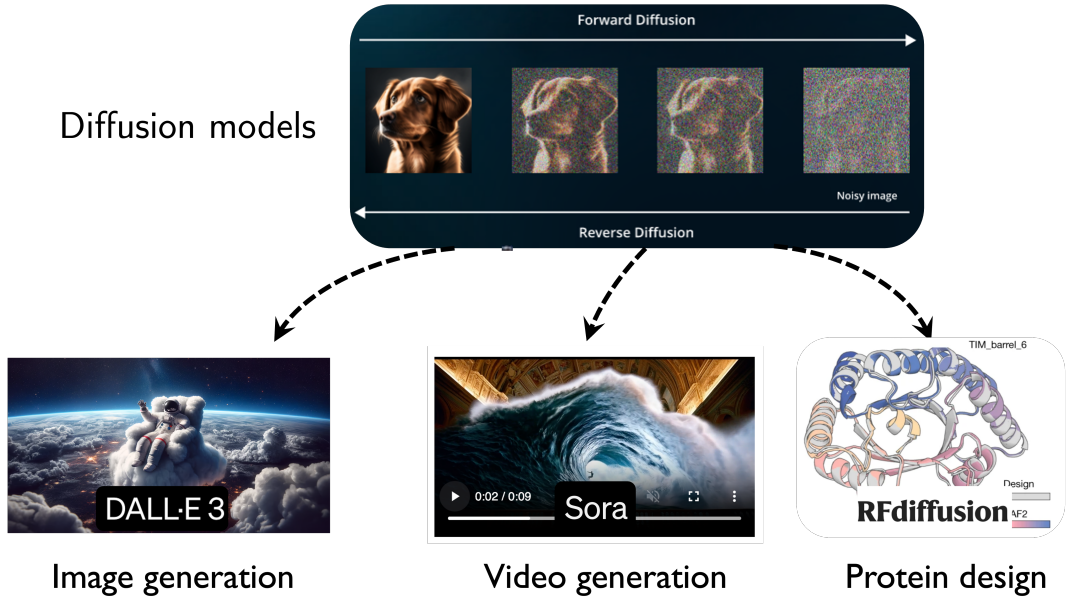
Diffusion Models are successful in image generation

Efficient for conditional generation, e.g., generation with prompts

Wharton
University of Pennsylvania

What is a diffusion model?

Apply in various tasks across image generation, video generation and some scientific fields



Diffusion Model

A stochastic/deterministic sampler to sample from a high-dimensional, non-smooth, implicit distribution:

Diffusion Model

A stochastic/deterministic sampler to sample from a high-dimensional, non-smooth, implicit distribution:

- Given training data $X^{\text{train,i}} \sim p_{\text{data}}$ $(1 \leq i \leq N)$ in Euclidean space
- Generate rate new samples $X \sim p_{\text{data}}$

Diffusion Model

A stochastic/deterministic sampler to sample from a high-dimensional, non-smooth, implicit distribution:

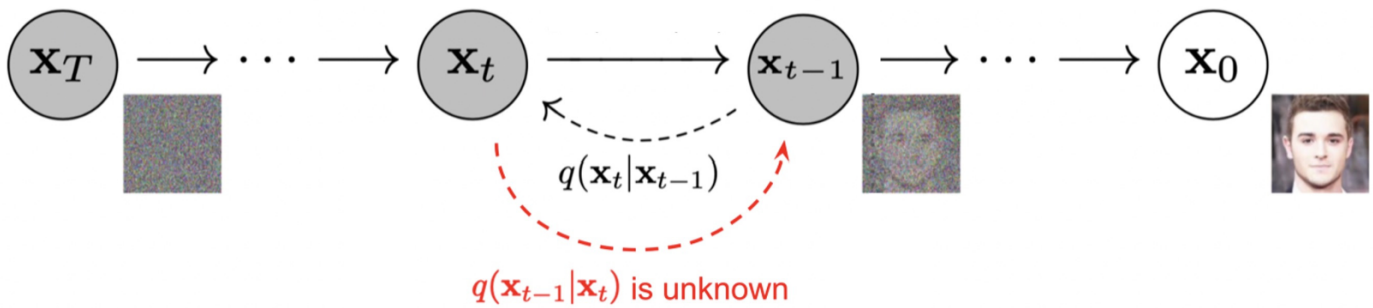
- Given training data $X^{\text{train,i}} \sim p_{\text{data}}$ $(1 \leq i \leq N)$ in Euclidean space
- Generate rate new samples $X \sim p_{\text{data}}$
- Usually p_{data} is a general distribution (density hard to be expressed or may not even exist)

Diffusion Model

A stochastic/deterministic sampler to sample from a high-dimensional, non-smooth, implicit distribution:

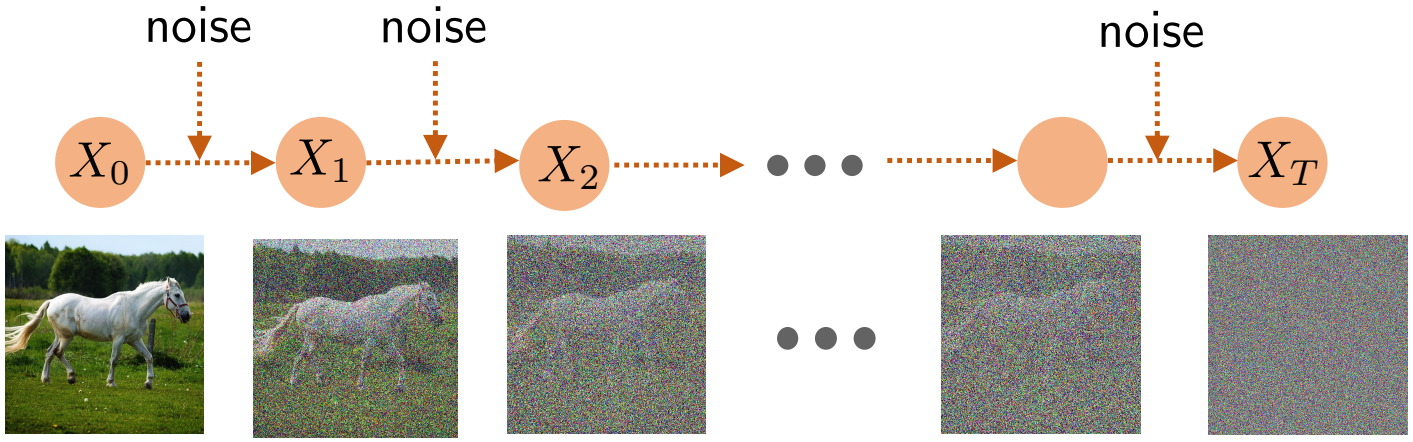
- Given training data $X^{\text{train,i}} \sim p_{\text{data}}$ $(1 \leq i \leq N)$ in Euclidean space
- Generate rate new samples $X \sim p_{\text{data}}$
- Usually p_{data} is a general distribution (density hard to be expressed or may not even exist)

Overcome the challenges?
— Learn the information from p_{data} through a diffusion process
— Sample by reversing this process and introducing new randomness



Wharton
University of Pennsylvania

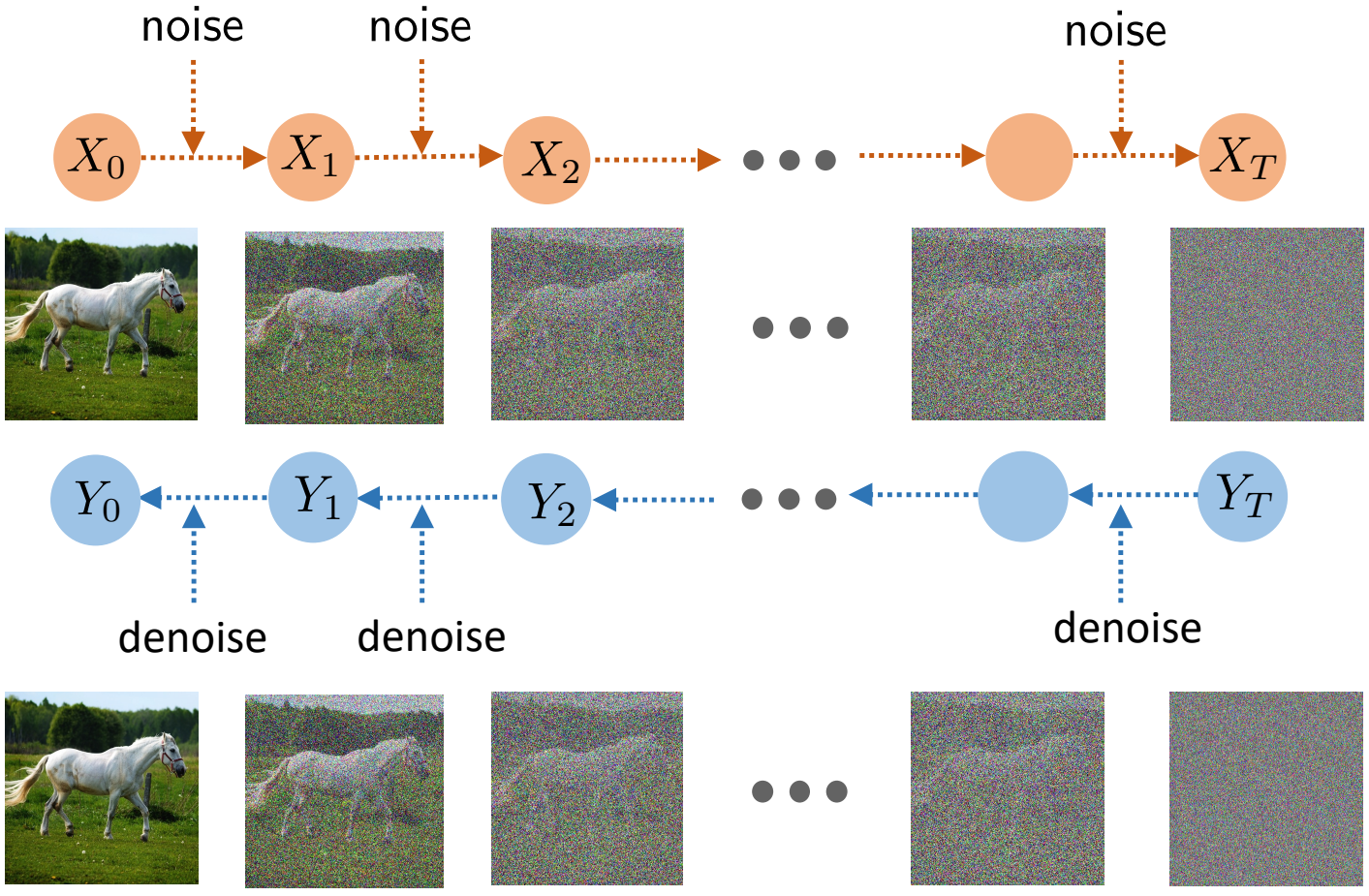
A Example with Gaussian noise



- X_T is very close to pure Gaussian noise when T is large

Wharton
University of Pennsylvania

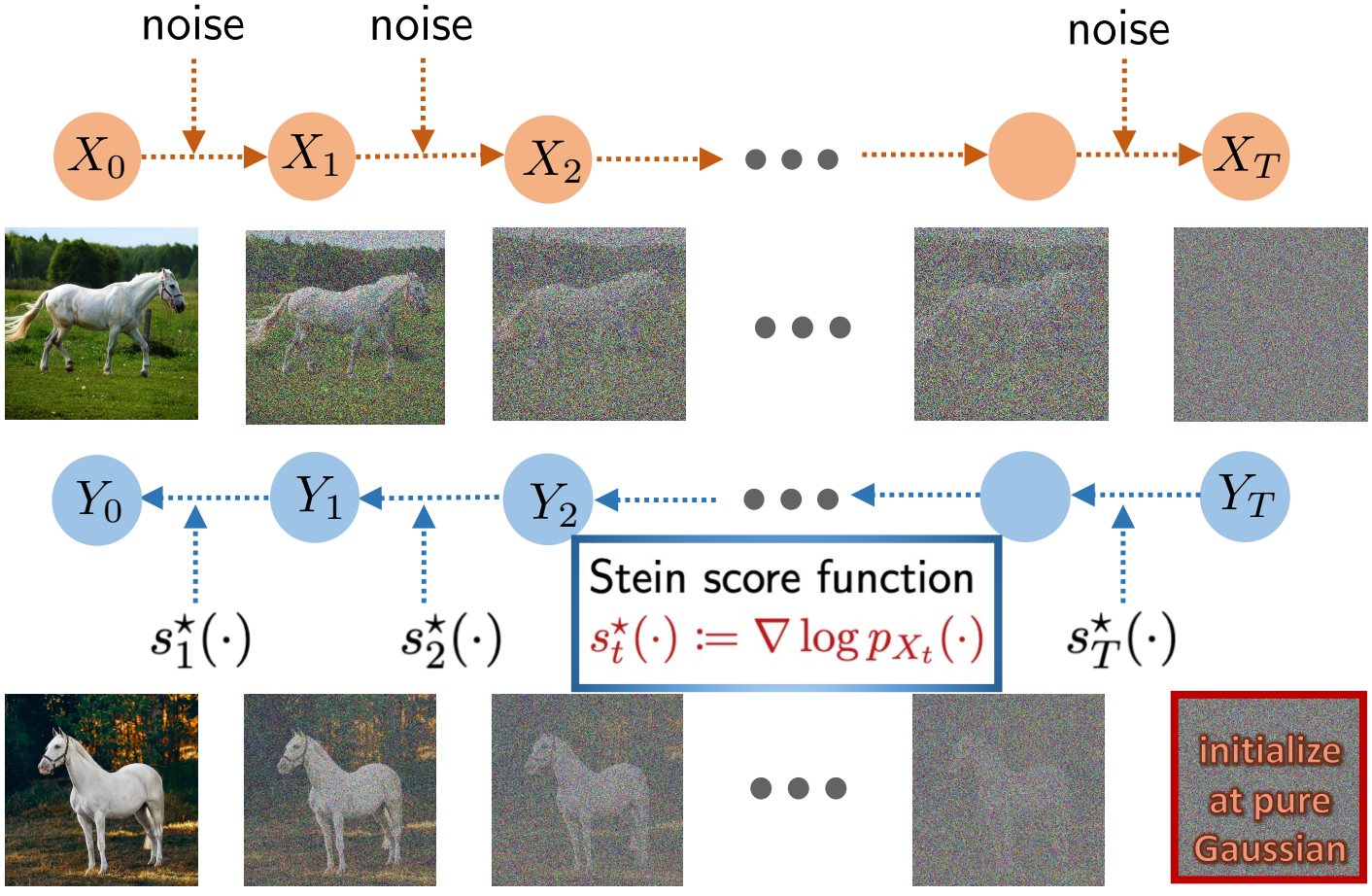
A Example with Gaussian noise



- X_T is very close to pure Gaussian noise when T is large
- Pretend $Y_T = X_T = \mathcal{N}(0, I_d)$ and reverse it

Wharton
University of Pennsylvania

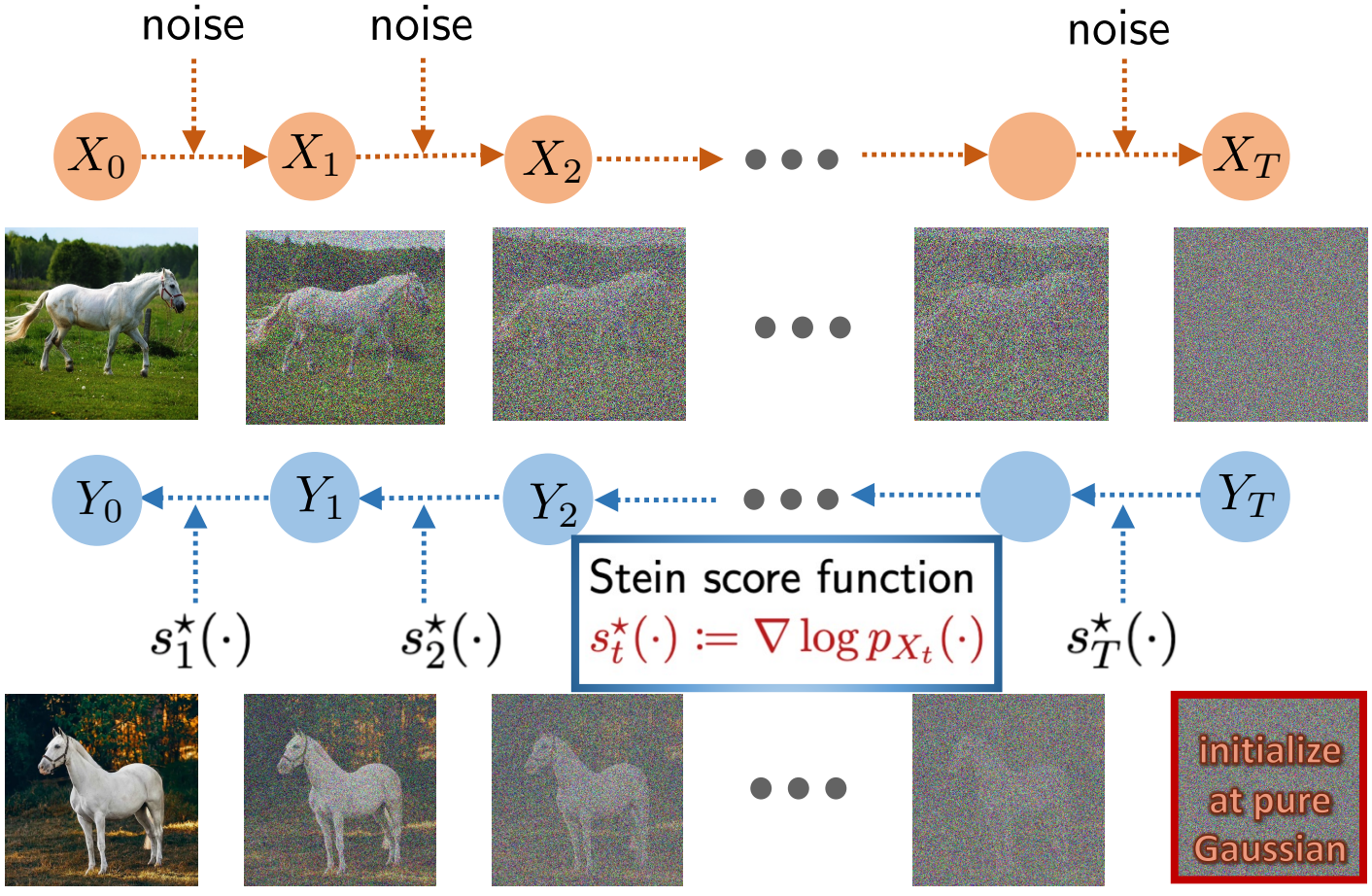
A Example with Gaussian noise



- X_T is very close to pure Gaussian noise when T is large
- Pretend $Y_T = X_T = \mathcal{N}(0, I_d)$ and reverse it

Wharton
University of Pennsylvania

A Example with Gaussian noise



- X_T is very close to pure Gaussian noise when T is large
- Pretend $Y_T = X_T = \mathcal{N}(0, I_d)$ and reverse it

Wharton
University of Pennsylvania

Diffusion Model

- Stochastic Differential Equation perspective:

$$\textit{forward process}: \quad X_0 \sim p_{\text{data}},$$
$$\mathrm{d}X_t = -X_t \mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$
$$\overset{\text{discretize}}{\Rightarrow} \quad X_t = \sqrt{1-\sigma_t^2} X_{t-1} + \sigma_t \mathcal{N}(0, I_d), \quad 1 \le t \le T$$

Based on this forward process, we aim to construct a (Markovian) backward process such that

$$\textbf{Goal}: X_t \overset{d}{\approx} Y_t \quad 1 \le t \le T$$

$$\textit{backward process}: \quad Y_T \sim \mathcal{N}(0, I_d),$$
$$\mathrm{d}Y_t = (Y_t + 2s_t^*(Y_t))\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$
$$\overset{\text{discretize}}{\Rightarrow} \quad Y_{t-1} = c_t^1 Y_t + c_t^2 s_t^*(Y_t) + c_t^3 \mathcal{N}(0, I_d)$$

$$\Rightarrow \text{Output } Y_1$$

Wharton
University of Pennsylvania

Diffusion Model

- Variational Inference perspective: In each backward step, find the best Gaussian approximation of $X_{t-1}|X_t$ by minimizing the variational lower bound:

$$\begin{aligned}&\min_\theta \mathbb{E}[\mathrm{KL}\left(q(X_{t-1}|X_t)||p_\theta(X_{t-1}|X_t)\right])\\ \approx &\min_\theta \mathbb{E}[\mathrm{KL}\left(q(X_{t-1}|X_t, X_0)||p_\theta(X_{t-1}|X_t, X_0)\right)]\end{aligned}$$

1. Notice $X_{t-1}|X_t \Rightarrow$ intractable $\Rightarrow X_{t-1}|X_t, X_0$ is Gaussian $\Rightarrow$ tractable with estimated X_0
2. In practice, minimize the sum of KL error from $1 \rightarrow T$, parametrized by unified θ

Implicitly learn the score function $s^*_{X_t}(\cdot) \Leftrightarrow$ the SDE form

Wharton
University of Pennsylvania

Discrete Diffusion Model

- When the data is supported on a discrete space, Gaussian noise is not the ideal choice
- Design a discrete space Markov chain as the forward process, whose the limit is some uniform distribution (easy to sample from)

$$q(X_t) = Q_t q(X_0) \quad \text{for } 1 \leq t \leq T$$

  Similar to the continuous case, learn the discrete score function $s_t^*(x) = \mathbf{p}_{X_t}/p_{X_t}(x)$
- The backward process can be designed as

$$p_\theta(Y_{t-1}) = \widetilde{Q}_t p_\theta(Y_t)$$

  where $\widetilde{Q}_t$ is a function of Q_t and $s_t^*(x)$

Wharton
University of Pennsylvania

Two Example Frameworks

Diffusion-LM Improves Controllable Text Generation

Xiang Lisa Li
Stanford University
xlisali@stanford.edu

John Thickstun
Stanford University
jthickst@stanford.edu

Ishaan Gulrajani
Stanford Univeristy
igul@stanford.edu

Percy Liang
Stanford Univeristy
pliang@cs.stanford.edu

Tatsunori B. Hashimoto
Stanford Univeristy
thashim@stanford.edu

Simple and Effective Masked Diffusion Language Models

Subham Sekhar Sahoo
Cornell Tech, NYC, USA.
ssahoo@cs.cornell.edu

Marianne Arriola
Cornell Tech, NYC, USA.
ma2238@cornell.edu

Yair Schiff
Cornell Tech, NYC, USA.
yzs2@cornell.edu

Aaron Gokaslan
Cornell Tech, NYC, USA.
akg87@cs.cornell.edu

Edgar Marroquin
Cornell Tech, NYC, USA.
emm392@cornell.edu

Justin T Chiu
Cornell Tech, NYC, USA.
jtc257@cornell.edu

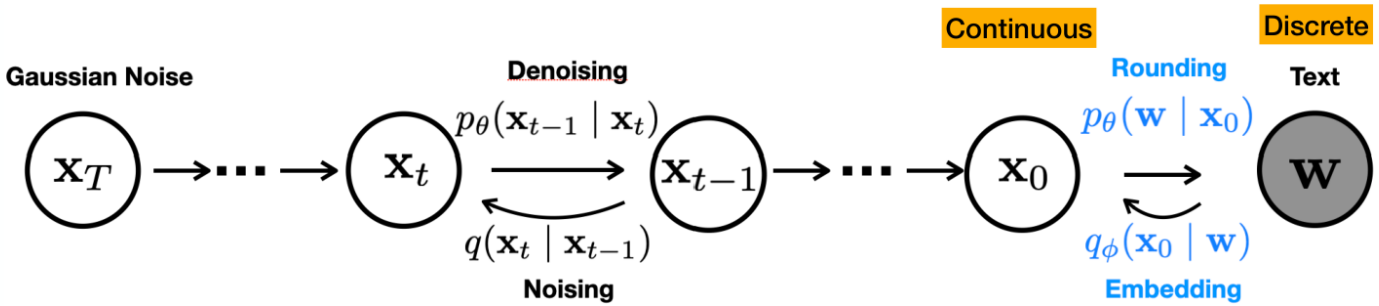
Alexander Rush
Cornell Tech, NYC, USA.
ar459@cornell.edu

Volodymyr Kuleshov
Cornell Tech, NYC, USA.
kuleshov@cornell.edu

Wharton
University of Pennsylvania

Diffusion LM

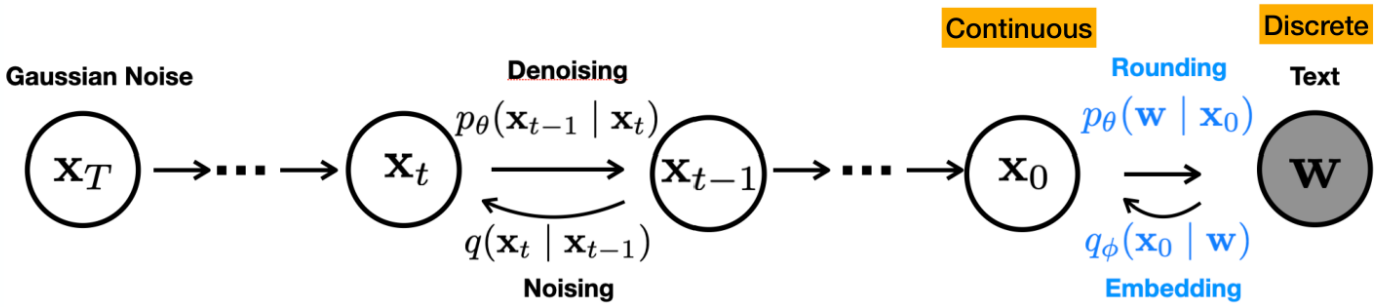
Encode discrete texts into continuous space and add Gaussian noise



- Requires embedding and rounding steps
- Lifting to continuous space allows one to compute gradients
- Ultimately helps with controllable text generation

Wharton
University of Pennsylvania

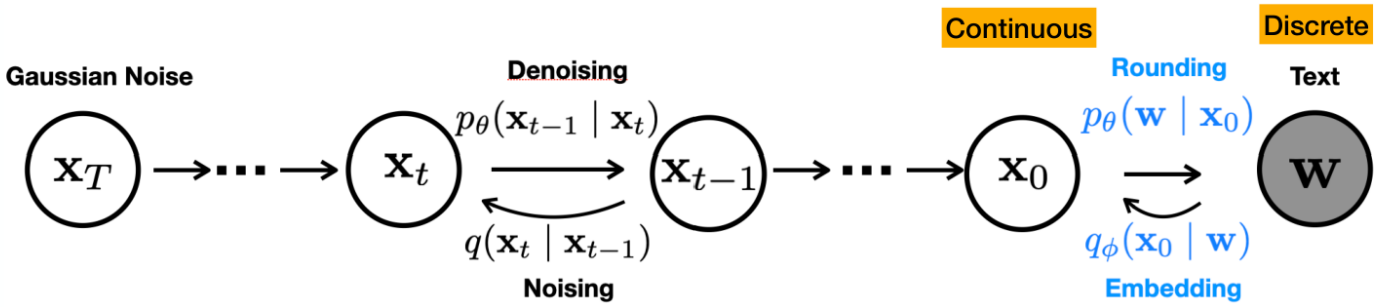
Diffusion LM



- Embedding step? End-to-end learning (e2e), parameterize

$$q_\phi(x_0|w) = \mathcal{N}(\text{Emb}_\phi(w), \sigma^2 \cdot I)$$

Diffusion LM



- Embedding step? End-to-end learning (e2e), parameterize

$$q_\phi(x_0|w) = \mathcal{N}(\text{Emb}_\phi(w), \sigma^2 \cdot I)$$

- Modified objective

$$\mathcal{L}_{\text{e2e}}(w) = \mathbb{E}_{x_0 \sim q_\phi}[\mathcal{L}_{\text{simple}}(x_0) - \log p_\theta(w|x_0)]$$

where $\mathcal{L}_{\text{simple}}(x_0)$ is the standard objective for continuous time diffusion models

$$\mathcal{L}_{\text{simple}}(x_0) = \sum_{t=1}^{T} \mathbb{E}_{q(x_t|x_0)}[\|\widehat{\mu}_\theta(x_t, t) - \mu_{t-1}(x_t, x_0)\|^2].$$

Wharton University of Pennsylvania

Diffusion LM

How to specify rounding step?

- Ideally, we would like to take most likely word from softmax output

$$\hat{w}_i = \arg\max_{w_i} p_\theta(w_i | x_{0,i})$$

Diffusion LM

How to specify rounding step?

- Ideally, we would like to take most likely word from softmax output

$$\hat{w}_i = \arg\max_{w_i} p_\theta(w_i | x_{0,i})$$

- Issue: x_0 does not 'commit' to a single word embedding

Diffusion LM

How to specify rounding step?

- Ideally, we would like to take most likely word from softmax output

$$\hat{w}_i = \arg\max_{w_i} p_\theta(w_i | x_{0,i})$$

- Issue: x_0 does not 'commit' to a single word embedding
- Solution, modify the standard objective further

$$\mathcal{L}_{\text{simple}}(x_0) \leftarrow \mathcal{L}_{\text{round}}(x_0) = \sum_{t=1}^{T} \mathbb{E}_{q(x_t|x_0)}[\|\widehat{f_\theta}(x_t, t) - x_0\|^2].$$

Diffusion LM

How to specify rounding step?

- Ideally, we would like to take most likely word from softmax output

$$\hat{w}_i = \arg\max_{w_i} p_\theta(w_i | x_{0,i})$$

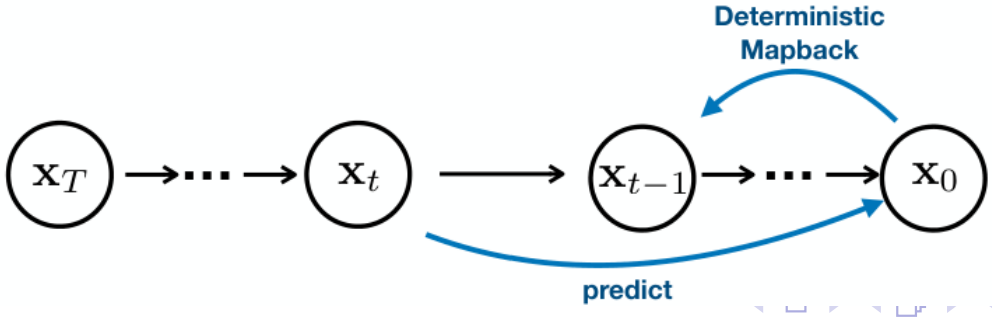
- Issue: x_0 does not 'commit' to a single word embedding
- Solution, modify the standard objective further

$$\mathcal{L}_{\text{simple}}(x_0) \leftarrow \mathcal{L}_{\text{round}}(x_0) = \sum_{t=1}^{T} \mathbb{E}_{q(x_t|x_0)}[\|\widehat{f}_\theta(x_t, t) - x_0\|^2].$$

- At decoding, use *clamping*

$$x_{t-1} = \sqrt{\bar{\alpha}} \cdot \text{Clamp}(f_\theta(x_t, t)) + \sqrt{1 - \bar{\alpha}} \cdot \epsilon.$$

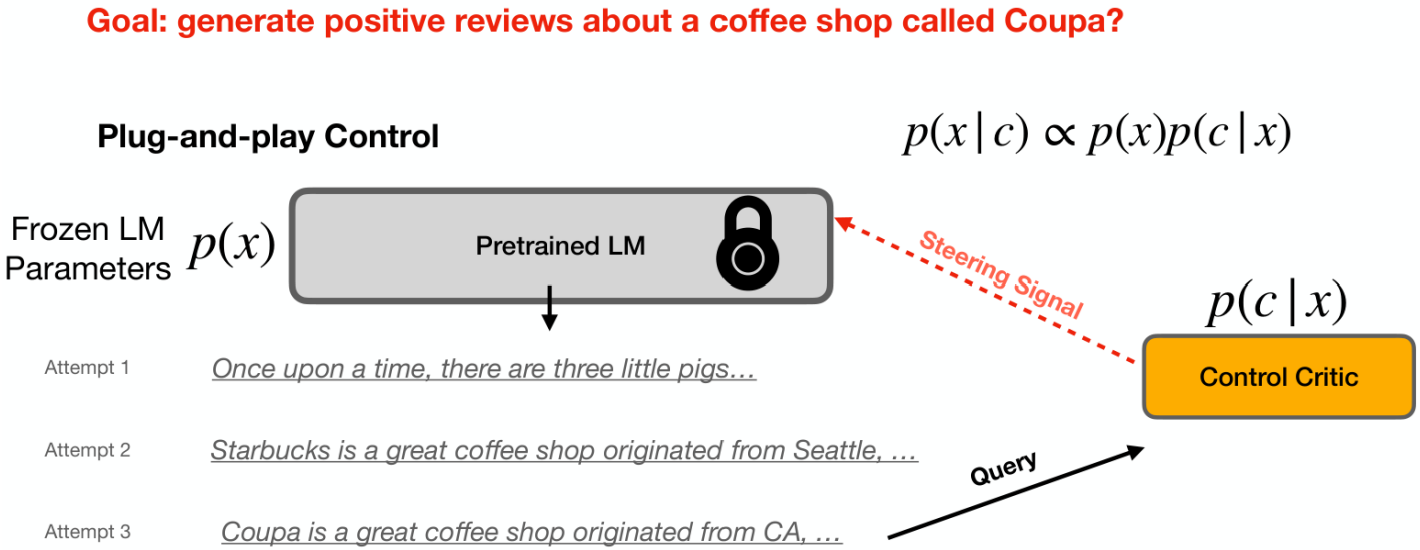
where $\epsilon \sim \mathcal{N}(0, I)$.



Wharton
University of Pennsylvania

Diffusion LM

Controllable text generation via plug-and-play



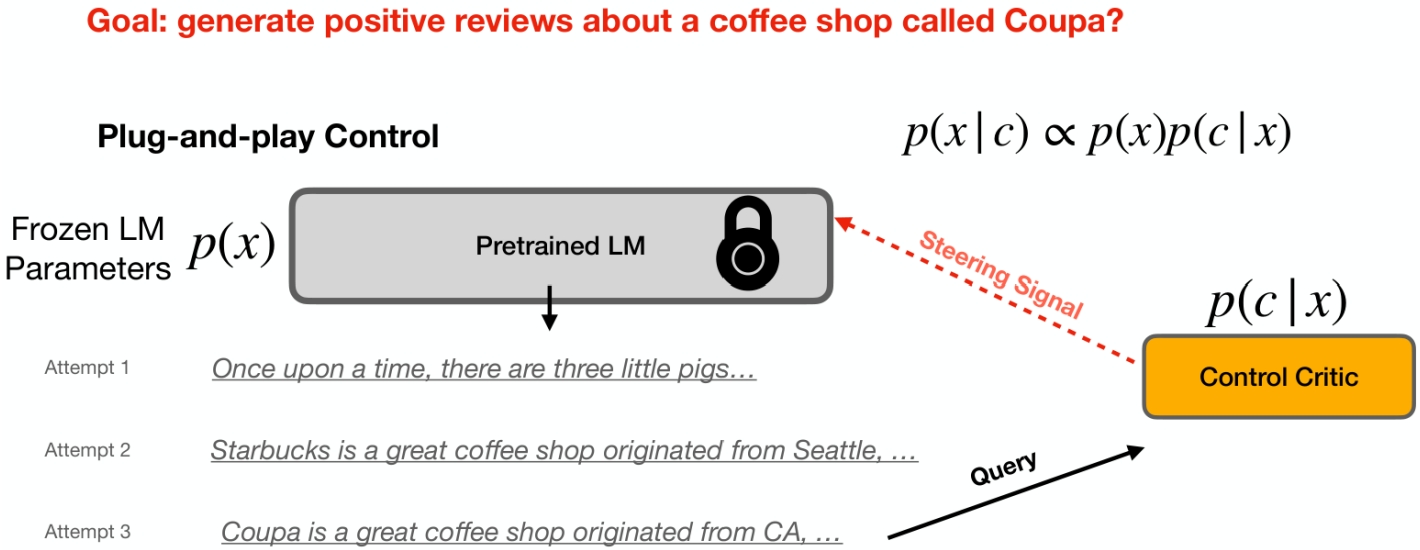
- Adds control in a post-hoc fashion

$$e_i^h \leftarrow e_i^h + \gamma \nabla_{e_i^h} \log p(c|e_i^h)$$

Wharton
University of Pennsylvania

Diffusion LM

Controllable text generation via plug-and-play



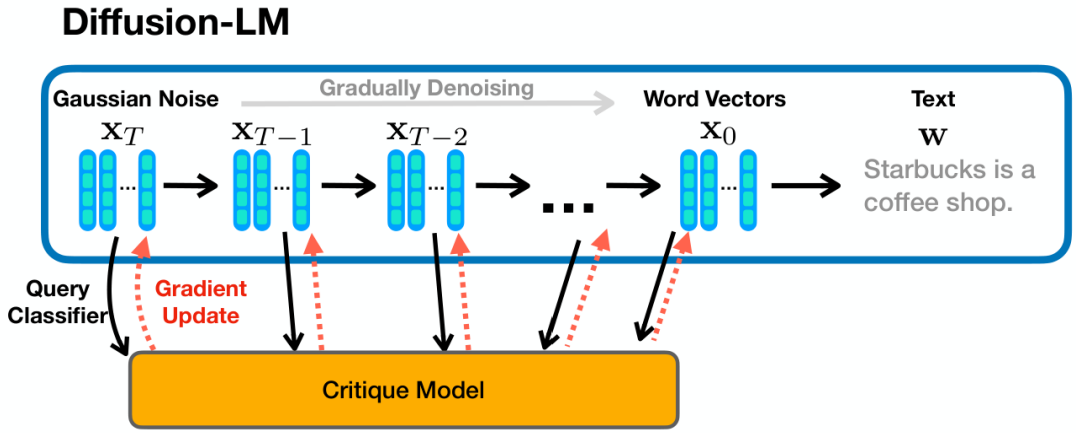
- Adds control in a post-hoc fashion

$$e_i^h \leftarrow e_i^h + \gamma \nabla_{e_i^h} \log p(c|e_i^h)$$

- Can we integrate control directly into the sampling process?

Wharton University of Pennsylvania

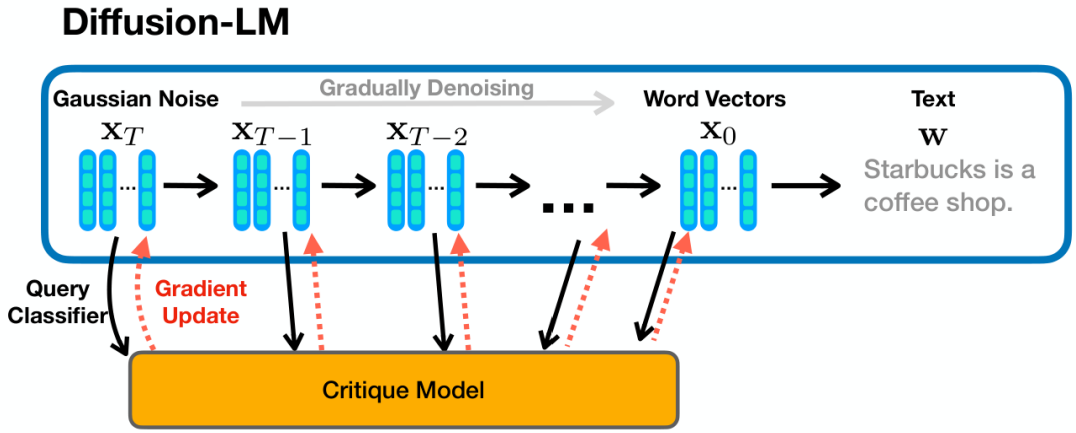
Diffusion LM



- Goal: learn to sample from

$$p(x_{0:T}|c) = \prod_{t=1}^{T} p(x_{t-1}|x_t, c) \propto \prod_{t=1}^{T} p(x_{t-1}|x_t) \cdot p(c|x_{t-1}, x_t)$$

Diffusion LM



- Goal: learn to sample from

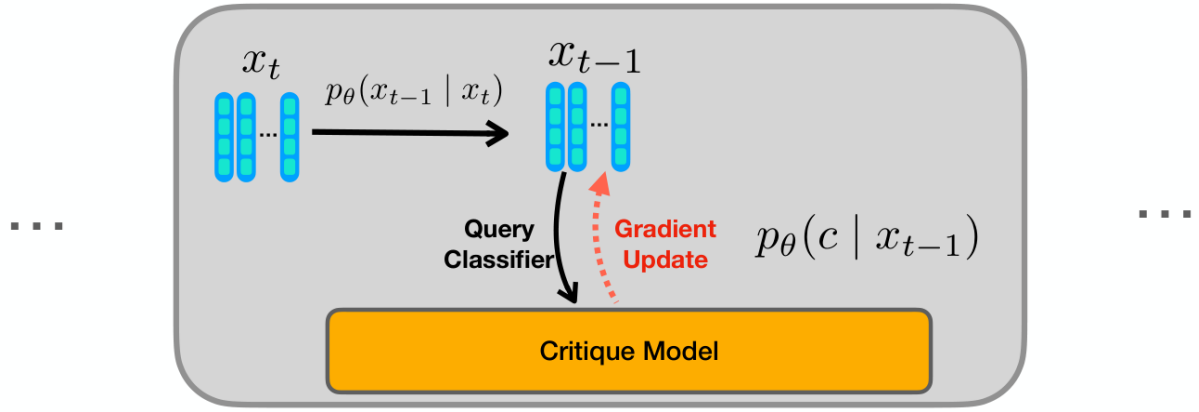
$$p(x_{0:T}|c) = \prod_{t=1}^{T} p(x_{t-1}|x_t, c) \propto \prod_{t=1}^{T} p(x_{t-1}|x_t) \cdot p(c|x_{t-1}, x_t)$$

- Motivates the following gradient update

$$\nabla_{x_{t-1}} \log p(x_{t-1}|x_t, c) = \nabla_{x_{t-1}} \log p(x_{t-1}|x_t) + \nabla_{x_{t-1}} \log p(c|x_{t-1})$$

Wharton
University of Pennsylvania

Diffusion LM



Update x_{t-1} in the following gradient direction:

$$\nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{x}_{t-1} \mid \mathbf{x}_t) + \nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{c} \mid \mathbf{x}_{t-1}),$$

Diffusion-LM transition | Classifier score

- use *fluency regularization*

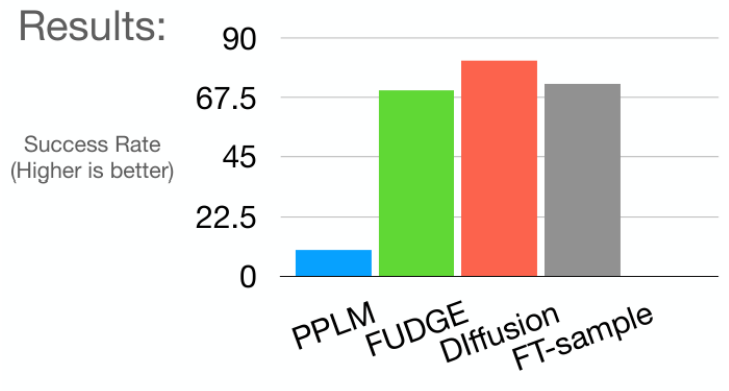
$$\lambda \nabla_{x_{t-1}} \log p(x_{t-1}|x_t) + \nabla_{x_{t-1}} \log p(c|x_{t-1})$$

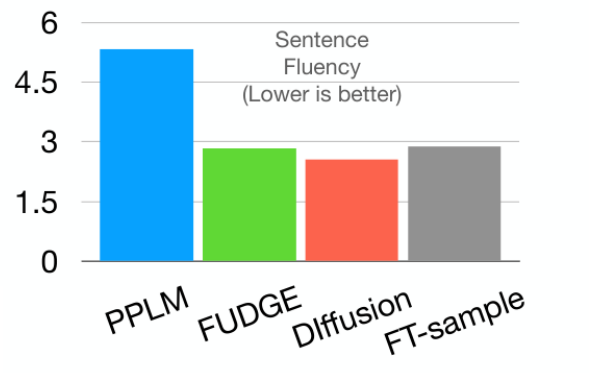
balances maximizing and sampling akin to nucleus sampling and temperature scaling

Wharton University of Pennsylvania

Diffusion LM

- Semantic Content task: generate a sensible sentence with a given field and value
- Food = Japanese $\rightarrow$ "Browns Cambridge is good for Japanese food and also children friendly."

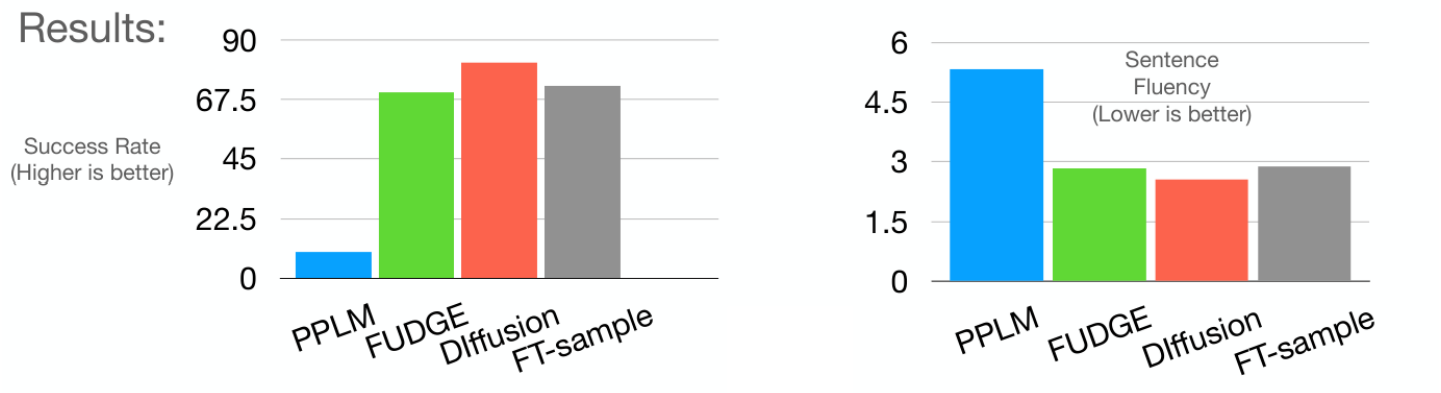




Wharton
University of Pennsylvania

Diffusion LM

- Semantic Content task: generate a sensible sentence with a given field and value
- Food = Japanese $\rightarrow$ "Browns Cambridge is good for Japanese food and also children friendly."



80M parameters, sequence length $n = 64$, diffusion steps $T = 2000$, square-root noise schedule; teacher LM is *GPT-2 model*

Wharton
University of Pennsylvania

Diffusion LM

input (Semantic Content) output text	food : Japanese Browns Cambridge is good for Japanese food and also children friendly near The Sorrento .
input (Parts-of-speech) output text	PROPN AUX DET ADJ NOUN NOUN VERB ADP DET NOUN ADP DET NOUN PUNCT Zizzi is a local coffee shop located on the outskirts of the city .
input (Syntax Tree) output text	(TOP (S (NP (*) (*) (*)) (VP (*) (NP (NP (*) (*))))) The Twenty Two has great food
input (Syntax Spans) output text	(7, 10, VP) Wildwood pub serves multicultural dishes and is ranked 3 stars
input (Length) output text	14 Browns Cambridge offers Japanese food located near The Sorrento in the city centre .
input (left context) input (right context) output text	My dog loved tennis balls. My dog had stolen every one and put it under there. One day, I found all of my lost tennis balls underneath the bed.

- for infilling control task, learn a conditional score

$$\nabla_{x_{t-1}} \log p(x_{t-1}|x_t, c)$$

where $c = (O_L, O_R)$ is left and right context

- this is more similar to question/response text generation in LLMs

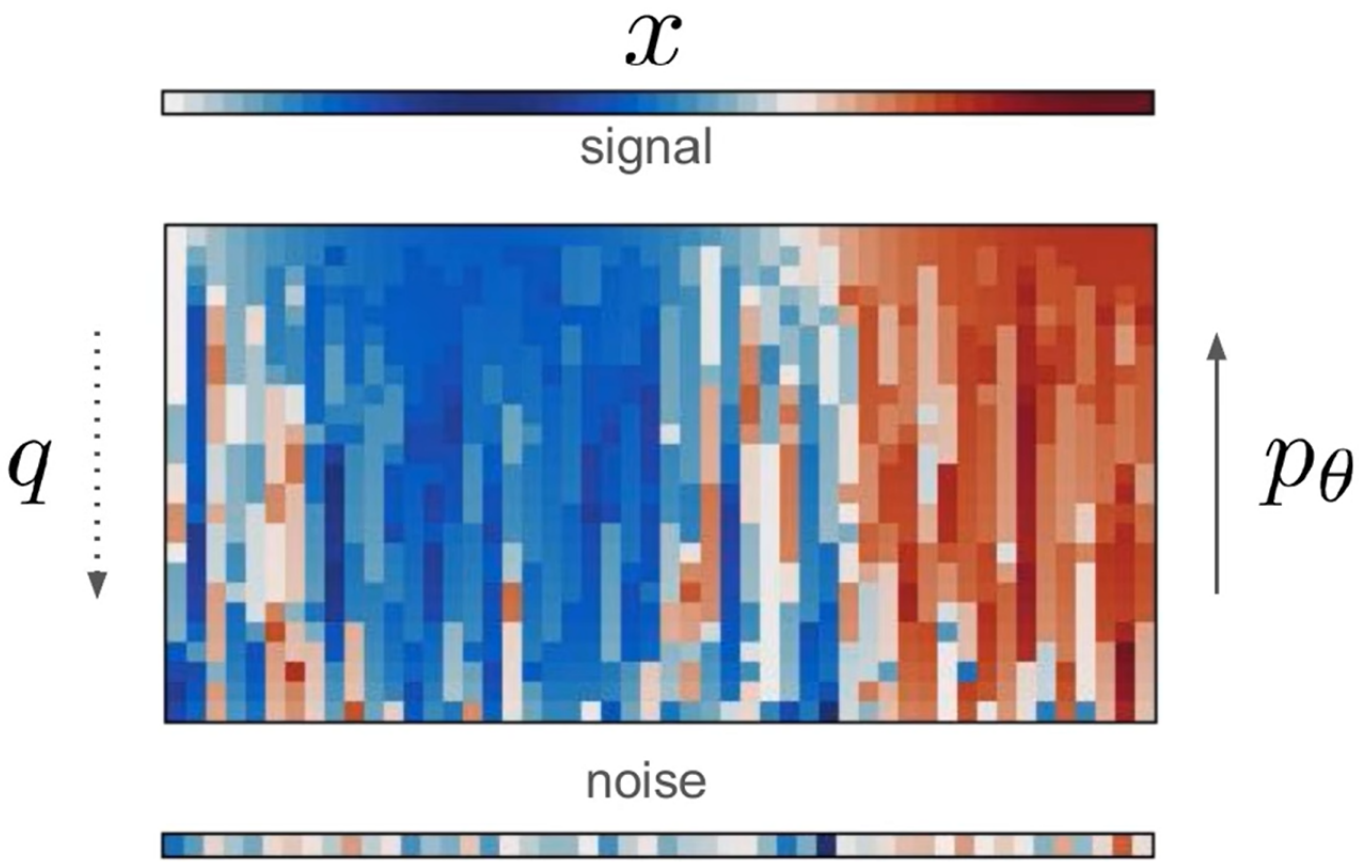
Wharton
University of Pennsylvania

Diffusion LM

	Semantic Content		Parts-of-speech		Syntax Tree		Syntax Spans		Length	
	ctrl ↑	lm ↓	ctrl ↑	lm ↓	ctrl ↑	lm ↓	ctrl ↑	lm ↓	ctrl ↑	lm ↓
PPLM	9.9	5.32	-	-	-	-	-	-	-	-
FUDGE	69.9	2.83	27.0	7.96	17.9	**3.39**	54.2	4.03	46.9	3.11
Diffusion-LM	**81.2**	**2.55**	**90.0**	**5.16**	**86.0**	3.71	**93.8**	**2.53**	**99.9**	**2.16**
FT-sample	72.5	2.87	89.5	4.72	64.8	5.72	26.3	2.88	98.1	3.84
FT-search	89.9	1.78	93.0	3.31	76.4	3.24	54.4	2.19	100.0	1.83

Figure: Diffusion-LM achieves high success rate (ctrl) and good fluency (lm) across all 5 control tasks, outperforming the PPLM and FUDGE baselines. Our method even outperforms the fine-tuning oracle (FT) on controlling syntactic parse trees and spans.

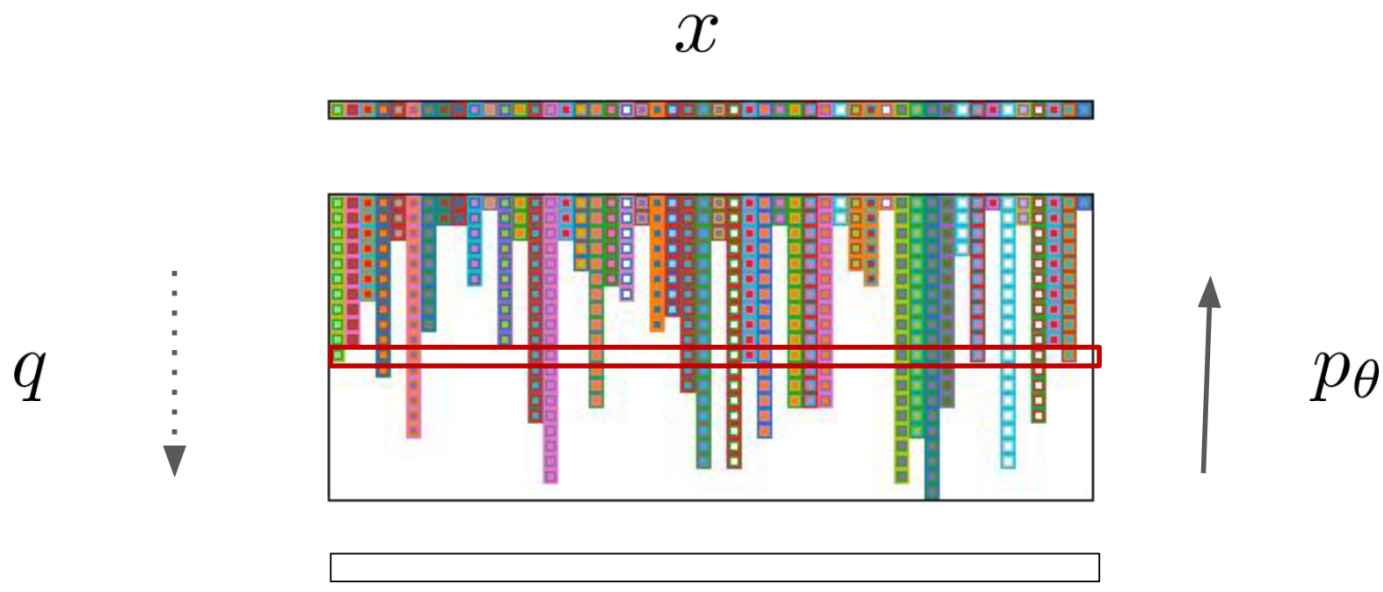
Masked Diffusion LM



Wharton
University of Pennsylvania

Masked Diffusion LM

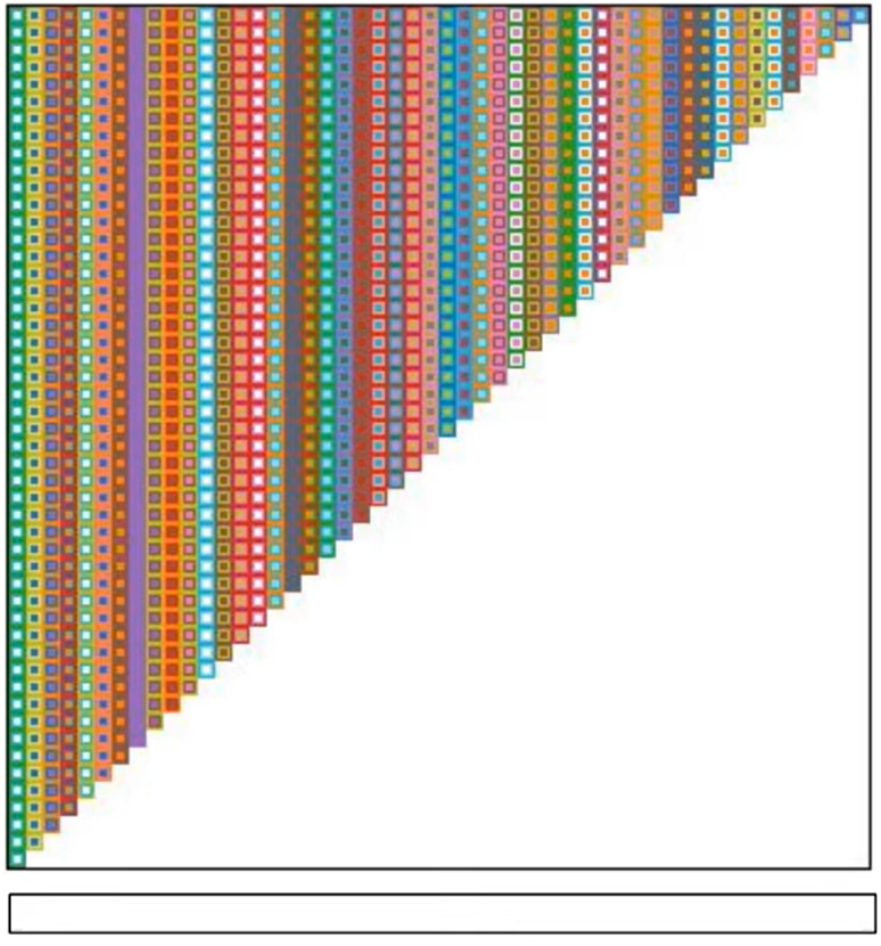
A demonstration of the discrete forward and backward process in the model



- q is the measure along the forward process
- Limit is "masked"— $[m]$, a special symbol
- p_θ is the measure along the backward process

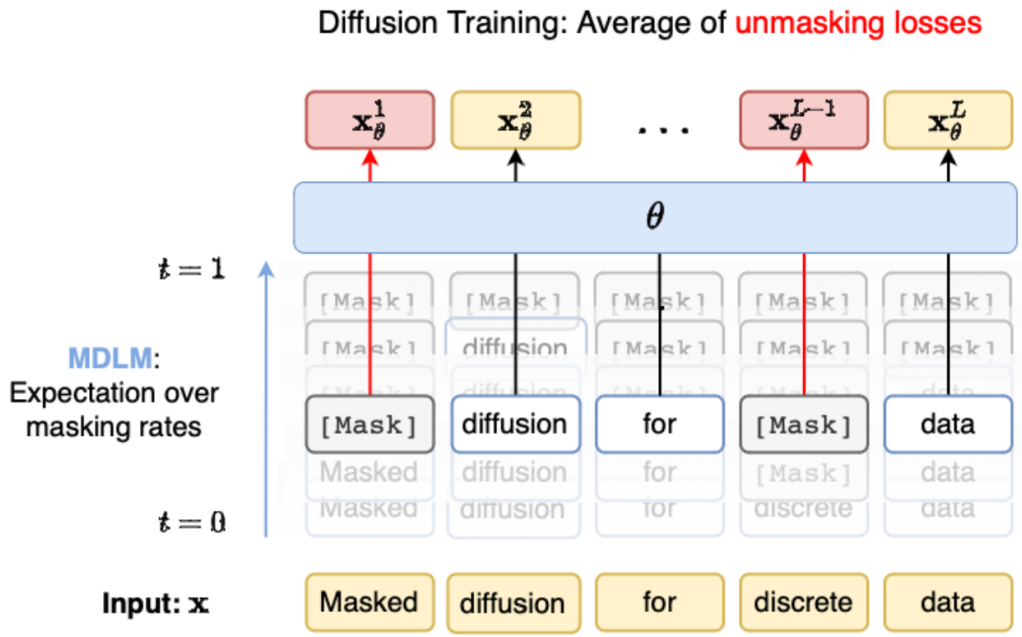
Wharton
University of Pennsylvania

Autoregressive Analogy



Wharton
University of Pennsylvania

Training Procedure



The forward Markov chain $\{Z_t\}_{t\in[0,1]}$, $Z_0 = X$, $Z_t = \alpha_t X + (1-\alpha_t)m$:

$$P(Z_t|X) = Q_t \ldots Q_{t_1} X$$

- Mixture of data X and noise m; Mixing time control α_t
- Simulate by a sequence of Markov transition kernels Q_{t_i}, $i = 1, \ldots, T$

Wharton
University of Pennsylvania

Training Objective Function

Information from the training procedure: discrete score function or A denoising predictor $x_\theta(Z_t, t)$. The loss function (continuous version) is

$$\mathcal{L}(\theta) = -\mathbb{E}_{Z_t} \int_0^1 \beta_t \log\langle x_\theta(Z_t, t), X\rangle \mathrm{d}t$$

The prediction model is set such that $\langle x_\theta(Z_t, t), m\rangle = 0$. Empirical version:

$$\mathcal{L}(\theta) = -\mathbb{E}_{t\sim \mathrm{Unif}[0,1], X\sim p_{\mathrm{data}}} \beta_t \log\langle x_\theta(\alpha_t X + (1-\alpha_t)m, t), X\rangle \mathrm{d}t$$

Wharton
University of Pennsylvania

Training Objective Function

Information from the training procedure: discrete score function or A denoising predictor $x_\theta(Z_t, t)$. The loss function (continuous version) is

$$\mathcal{L}(\theta) = -\mathbb{E}_{Z_t} \int_0^1 \beta_t \log\langle x_\theta(Z_t, t), X\rangle \mathrm{d}t$$

The prediction model is set such that $\langle x_\theta(Z_t, t), m\rangle = 0$. Empirical version:

$$\mathcal{L}(\theta) = -\mathbb{E}_{t\sim \mathrm{Unif}[0,1], X\sim p_{\mathrm{data}}} \beta_t \log\langle x_\theta(\alpha_t X + (1-\alpha_t)m, t), X\rangle \mathrm{d}t$$

Implementation of denosing predictor

- Tokenization is critical to performance, small vocabulary $\downarrow$
- Architectural matters: diffusion transformer (encoder-only, rotary positional embeddings for time conditioning)
- Choose correlated t_i, low-discrepency sequence

Wharton
University of Pennsylvania

Sampling Procedure

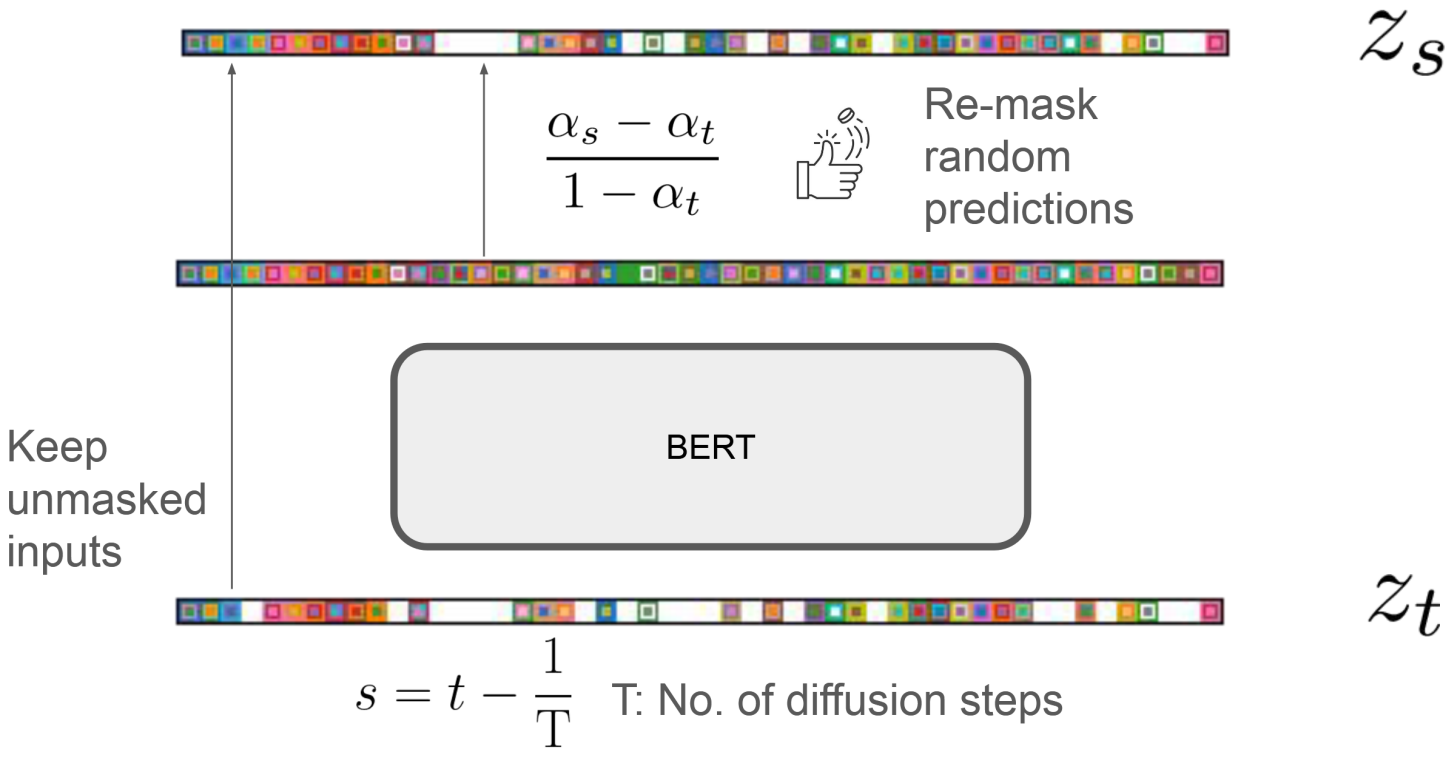
Sampling procedure is a reversed version of forward Markov chain, for any $s < t$

$$Z_s|Z_t = \begin{cases} Z_t^i & \text{if } Z_t^i \neq m \\ \frac{1}{1-\alpha_t}((1-\alpha_s)x_\theta(Z_t,t)^i + (\alpha_s - \alpha_t)m) & \text{if } Z_t^i = m \end{cases}$$

Entrywise denoising process based on the rule:

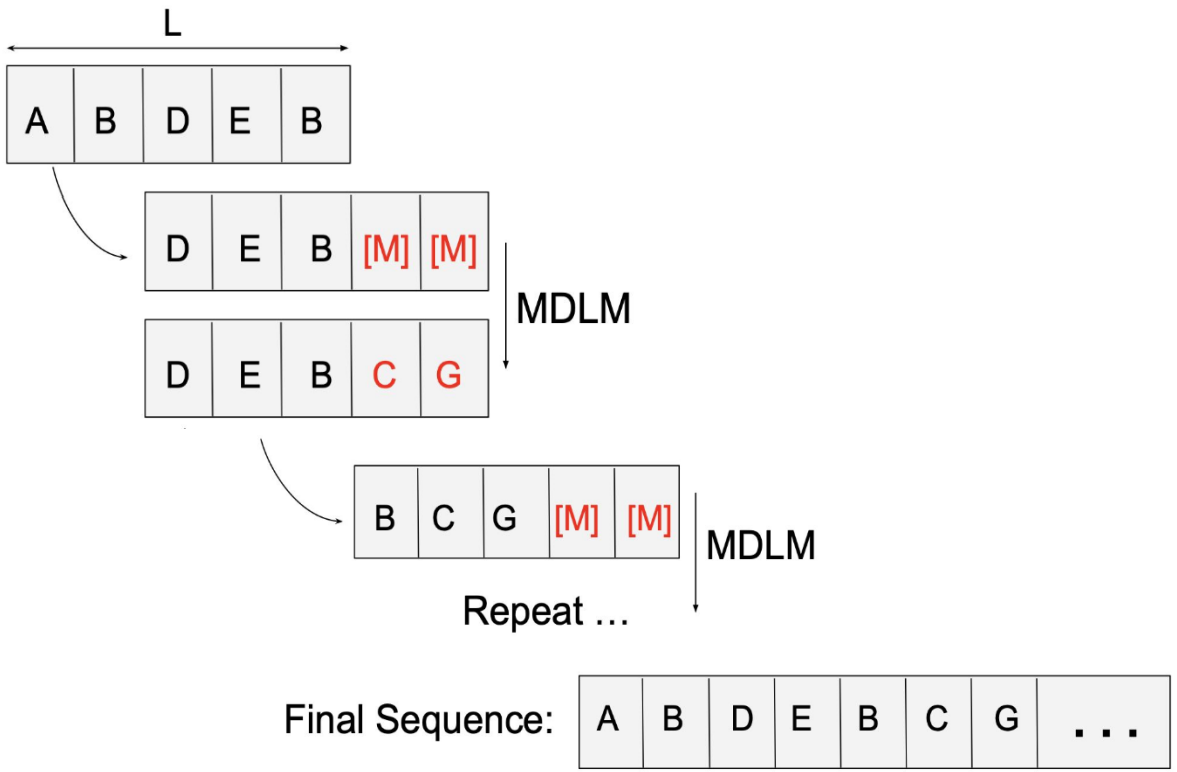
1. Begin with pure noise, all tokens are masked
2. When a token is unmasked ($\neq m$), it remains unchanged during reverse diffusion
3. For masked token, w.p. $(\alpha_s - \alpha_t)/(1-\alpha_t)$ unmask it and apply denoiser x_θ

One Step of Sampling Procedure



Extension: Semi-autoregressive Method

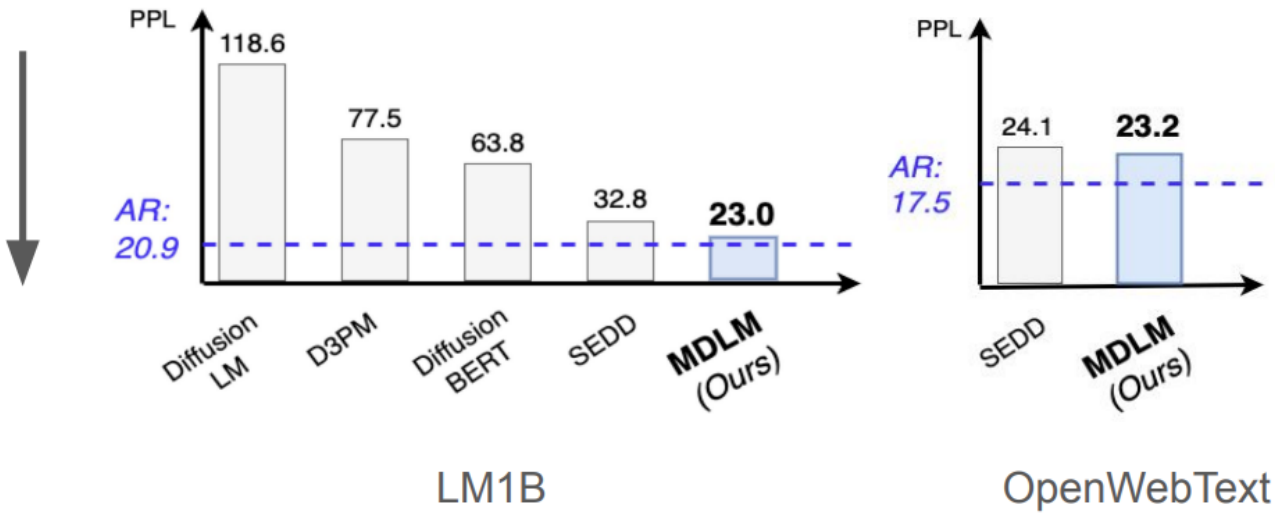
Instead of generating the whole sequence at once, generate part of it and continue to generate sequence of arbitrary length
— Add "masked" somewhere in the sequence and run the backward process again



Wharton
University of Pennsylvania

Experiments

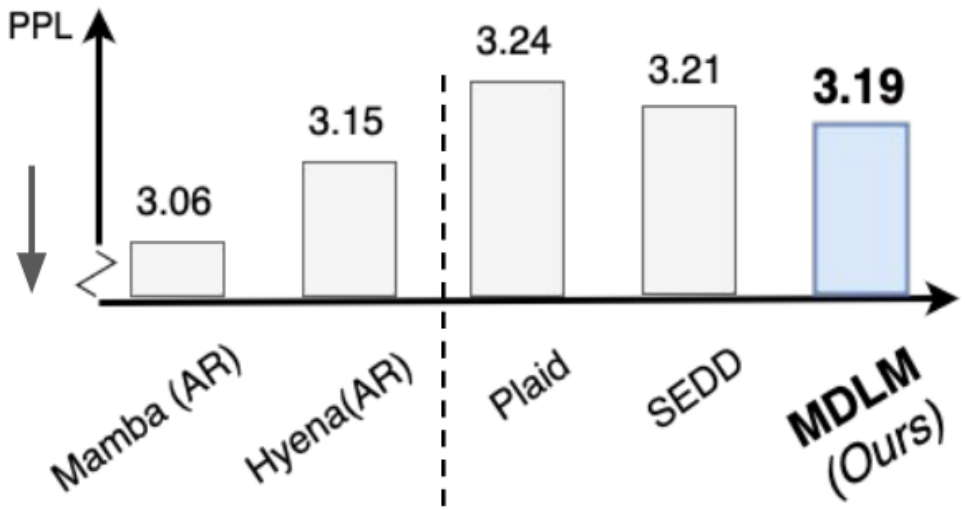
- Tested on standard One Billion Words (LM1B) and OpenWebText (OWT) datasets; evaluate perplexity
- $\sim$ 110M parameters; trained on $\sim$ 300B tokens; optional time conditioning
- Beat other diffusion-based models, 10% $\sim$ 25% worse than AR models
- faster sampling compared with other diffusion-based models, slower than AR models (same perplexity)



Wharton
University of Pennsylvania

Experiments

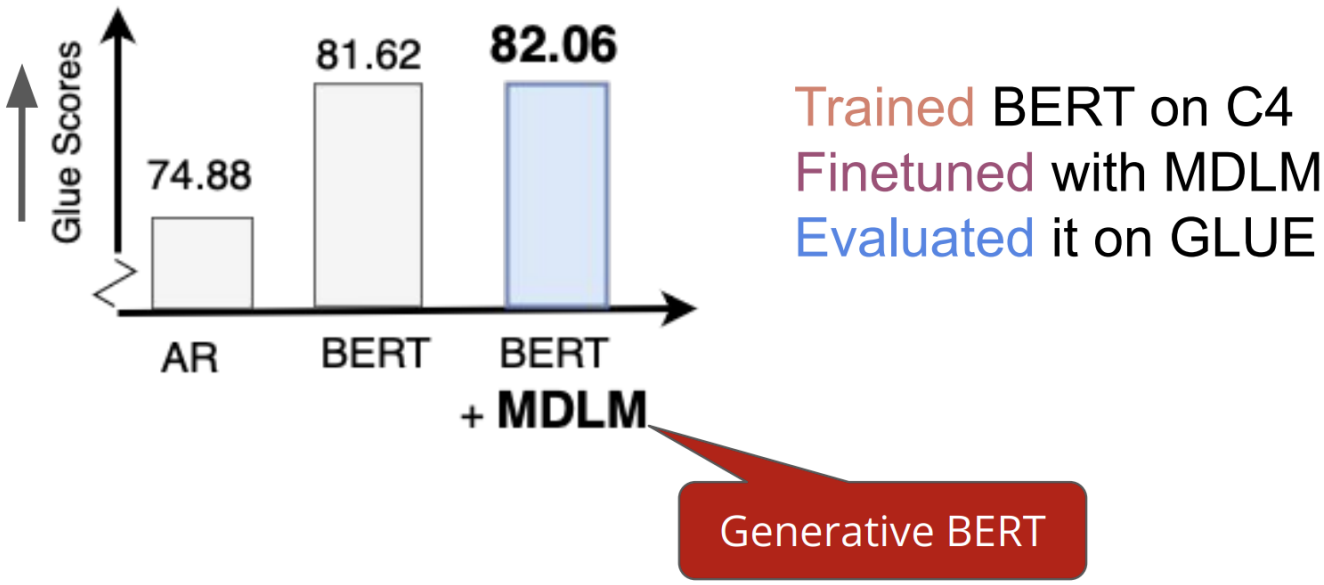
- Apply to the generative modeling of DNA sequences; finetune Caduceus DNA model with both AR and diffusion LMs
- $\sim$ 500K parameters, trained on the HG38 human genome dataset
- Evaluate on the downstream bio tasks — weak correlation between performance and perplexity



Wharton
University of Pennsylvania

Experiments

- Tested on General Language Understanding Evaluation (GLUE): sentiment analysis, linguistic acceptability...
- Score: Improve BERT a little bit, better than AR models
- Perplexity: Diffusion LM fine tuning reduce perplexity ($78 \rightarrow 35$); still higher than AR models(~ 23)



Wharton
University of Pennsylvania

Summary of the Diffusion LM Frameworks

- Parallel Generation: sequentially conditioned on preceding tokens $\rightarrow$ parallel generation of all output tokens. Enhance the speed and efficiency of some real world applications
- Text Interpolation: generate intermediate sentences between two given parts, smooth transitions and coherent outputs.
- Token-level control: conditional diffusion process
- Robust to corruption: denoising mechanism aids in mitigating errors and noise present in the input sequence
- Different Paradigm...

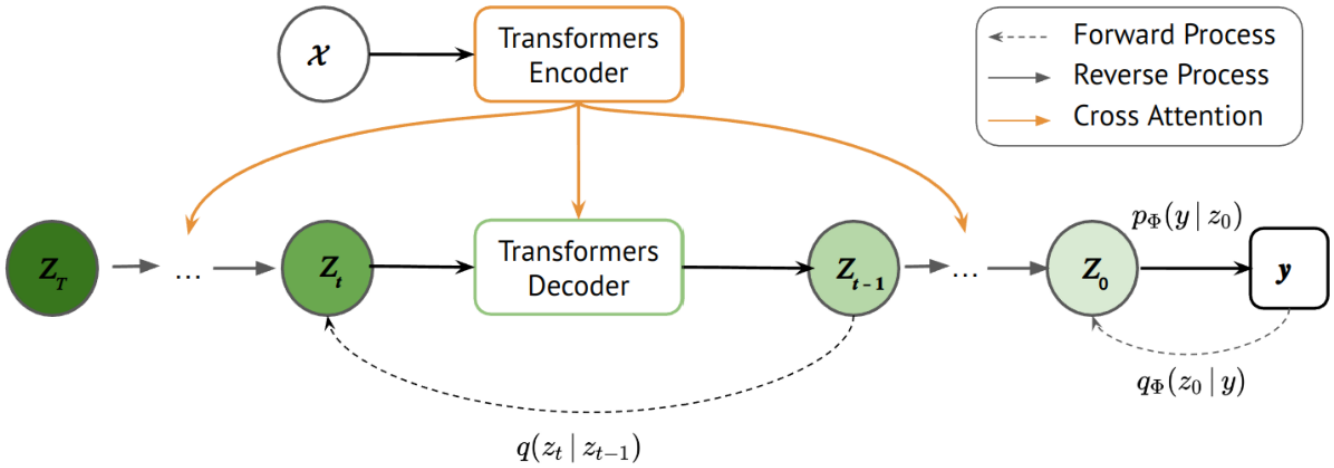
Summary of the Diffusion LM Frameworks

- Parallel Generation: sequentially conditioned on preceding tokens $\rightarrow$ parallel generation of all output tokens. Enhance the speed and efficiency of some real world applications
- Text Interpolation: generate intermediate sentences between two given parts, smooth transitions and coherent outputs.
- Token-level control: conditional diffusion process
- Robust to corruption: denoising mechanism aids in mitigating errors and noise present in the input sequence
- Different Paradigm...

	Advantages				*Disdvantages*	
Models	**Parallel Generation**	**Text Interpolation**	**Token-level Controls**	**Robustness to Input Corruption**	**Training Complexity**	**Model Interpretability**
AR Models	✗	✗	✗	✗	$\mathcal{O}(n)$	☺
Diffusion Models	✓	✓	✓	✓	$\mathcal{O}(n^T)$	☹

Wharton
University of Pennsylvania

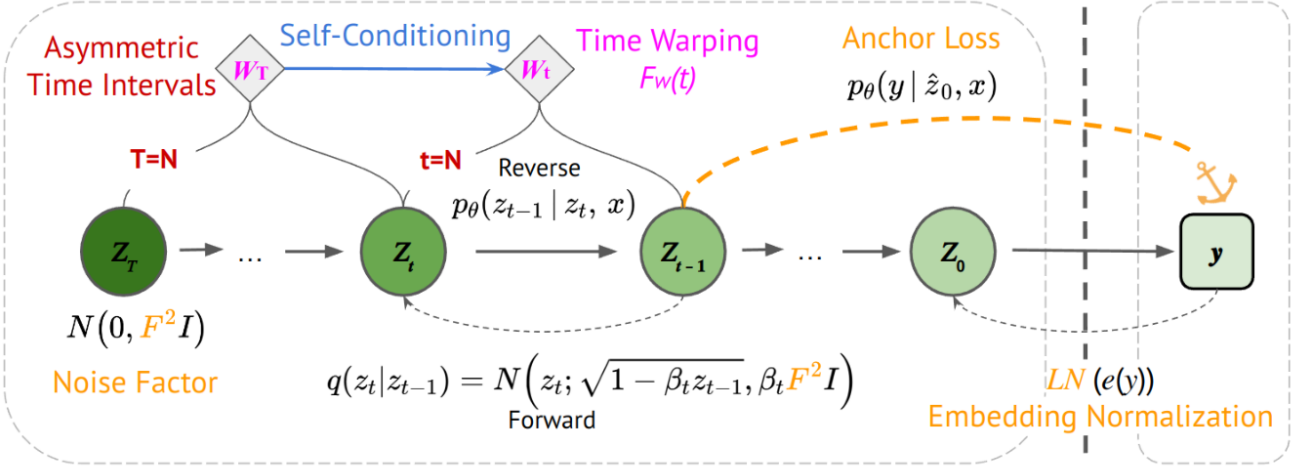
Summary of the Diffusion LM Frameworks



- transformers use bi-directional attention
- encoding used for contextual embeddings
- has spurred *lots* of algorithms & techniques for handling discrete data

Wharton
University of Pennsylvania

Summary of the Diffusion LM Frameworks



Anchor Loss, Noise Factor, Embedding Normalization: *Differmer*
Asymmetric Time Intervals: *Bit Diffusion*
Self-Conditioning: *Bit Diffusion, SED, CDCD, SeqDiffuSeq, DiNoiSer*
Time Warping: *CDCD, SeqDiffuSeq*

- Broadly fall into two categories (i) adapting models to discrete variables, and (ii) improved sampling procedures

Summary of the Diffusion LM Frameworks

Model	Tasks	Schedule	Sampling
Discrete Diffusion Models			
Multinomial Diffusion (Hoogeboom et al., 2021)	unconditional text generation, unsupervised spell-checking	Transition matrices	—
D3PM (Discrete Denoising Diffusion Probabilistic Models) (Austin et al., 2021b)	char-level text and image generation	Uniform Transition Matrices	—
Zero-shot Diffusion (Nachmani and Dovrat, 2021)	machine translation	Partial Noising	Classifier-free conditional denoising
SUNDAE (Step-unrolled Denoising Autoencoders) (Savinov et al., 2021)	machine translation and unconditional text generation	Uniform Transition Matrices	Low-temperature sampling, Argmax-unrolled decoding, fewer token update
DiffusionBERT (He et al., 2022)	unconditional text generation	Spindle	x0-parameterization
SSD-LM (Semi-autoregressive Simplex-based Diffusion) (Han et al., 2022)	unconditional and controlled text generation	Logits generation	Greedy projection, Sampling, Multi-hot
Bit Diffusion (Generating Discrete Data using Diffusion Models with Self-Conditioning) (Chen et al., 2023b)	categorical image generation and image captioning	—	Self-Conditioning, Asymmetric Time Intervals
DiffusER (Discrete Diffusion via Edit-	machine translation, summa-	Edit-based Corrup-	Beam Search, 2D Beam

For more architectures, see Zou et al. "A survey of diffusion models in natural language processing."

Wharton
University of Pennsylvania

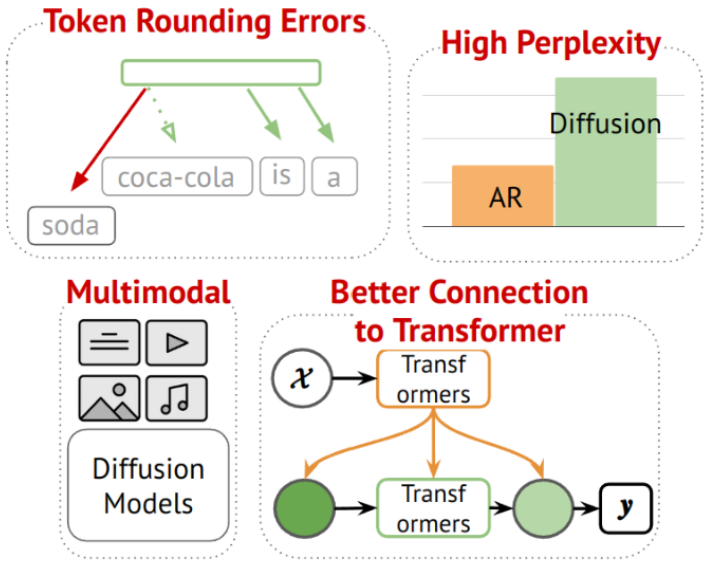
Summary of the Diffusion LM Frameworks

Size and computational cost of models?

Model	**# Parameters**	**GPU Hours**
Stable Diffusion 2.0	1B	$\approx$ 200k A100 GPU hours
Diffusion-LM (variant)	500M	$\approx$ 50k A100 GPU hours
GPT-3	175B	$\approx$ 1,000k A100 GPU hours

There's a gap in the computational aspect, so text-based diffusion models have potential to perform well

Summary of the Diffusion LM Frameworks



- connecting spatiality of transformers and temporality of diffusion models, e.g., including time step t into transformer
- diffusion models naturally excel in image generation, how to create more semantically accurate visual outputs?

Wharton
University of Pennsylvania

References I

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2025.

Hao Zou, Zae Myung Kim, and Dongyeop Kang. A survey of diffusion models in natural language processing. *arXiv preprint arXiv:2305.14671*, 2023.