Diffusion Models for Text
Beyond autoregressive language modeling

Xiang Lisa Li
Monopoly of Autoregressive LMs

All models for text generation are autoregressive.
Monopoly of Autoregressive LMs

All models for text generation are autoregressive.
Monopoly of Autoregressive LMs

Autoregressive language models have been dominating NLP!

All models for text generation are autoregressive.
Autoregressive Language Modeling

Harry Potter graduated from

Autoregressive LM

e.g., GPT-3
Autoregressive Language Modeling

Harry Potter graduated from

e.g., GPT-3
Autoregressive Language Modeling

\[
\begin{align*}
P(\text{Hogwarts} \mid \text{Harry Potter graduated from}) & \quad 0.8 \\
P(\text{Oxford} \mid \text{Harry Potter graduated from}) & \quad 0.05 \\
P(\text{is} \mid \text{Harry Potter graduated from}) & \quad 0.0001 \\
\end{align*}
\]

Harry Potter graduated from

Autoregressive LM

\[\ldots\]

e.g., GPT-3
Autoregressive Language Modeling

- $P(\text{Hogwarts} \mid \text{Harry Potter graduated from}) = 0.8$
- $P(\text{Oxford} \mid \text{Harry Potter graduated from}) = 0.05$
- $P(\text{is} \mid \text{Harry Potter graduated from}) = 0.0001$

Harry Potter graduated from

Autoregressive LM

e.g., GPT-3
Autoregressive Language Modeling

Hogwarts

\[
P(\text{Hogwarts} \mid \text{Harry Potter graduated from}) \quad 0.8 \\
P(\text{Oxford} \mid \text{Harry Potter graduated from}) \quad 0.05 \\
P(\text{is} \mid \text{Harry Potter graduated from}) \quad 0.0001 \\
\ldots
\]

Harry Potter graduated from

e.g., GPT-3
Autoregressive Language Modeling

Harry Potter graduated from Hogwarts

Autoregressive LM

e.g., GPT-3
Autoregressive Language Modeling

Harry Potter graduated from Hogwarts

e.g., GPT-3
Autoregressive Language Modeling

\[
P(\text{and} \mid \text{Harry Potter graduated from Hogwarts}) \quad 0.65 \\
P(\text{school} \mid \text{Harry Potter graduated from Hogwarts}) \quad 0.15 \\
P(\ . \mid \text{Harry Potter graduated from Hogwarts}) \quad 0.2
\]

Harry Potter graduated from Hogwarts

e.g., GPT-3
Autoregressive Language Modeling

\[
\begin{align*}
P(\text{and} \mid \text{Harry Potter graduated from Hogwarts}) &= 0.65 \\
P(\text{school} \mid \text{Harry Potter graduated from Hogwarts}) &= 0.15 \\
P(\ . \mid \text{Harry Potter graduated from Hogwart}) &= 0.2 \\
\end{align*}
\]

\[\text{Harry Potter graduated from Hogwarts}\]
Autoregressive Language Modeling

P( and | Harry Potter graduated from Hogwarts) 0.65
P( school | Harry Potter graduated from Hogwarts) 0.15
P( . | Harry Potter graduated from Hogwarts) 0.2

Harry Potter graduated from Hogwarts

e.g., GPT-3
Autoregressive Language Modeling

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

Harry Potter graduated from Hogwarts
Decoding from Autoregressive LM
Decoding from Autoregressive LM

Generating text from left-to-right, one at a time.
Decoding from Autoregressive LM

Generating text from left-to-right, one at a time.
Decoding from Autoregressive LM

Generating text from left-to-right, one at a time.

Transformer blocks

1. Time complexity: $O(n)$ where $n$ is the length of text.
Decoding from Autoregressive LM

Generating text from left-to-right, one at a time.

1. Time complexity: $O(n)$ where $n$ is the length of text.

2. Fixed generation order.

What if we want to generate right-to-left? Or given left and right context, fill in the middle?
Can we generate all words at once?
Can we generate all words at once?

Diffusion-LM: Diffusion based Language Models
Can we generate all words at once?

Diffusion-LM: Diffusion based Language Models

1. What is diffusion model?

2. Apply diffusion to text

3. Discussion
   1. Distinction btw text and images
   2. Compare with autoregressive LM
Diffusion Model for Images is very successful!

A brain riding a rocketship heading towards the moon.

A bald eagle made of chocolate powder, mango, and whipped cream.

A small cactus wearing a straw hat and neon sunglasses in the Sahara desert.

A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach hat.
Diffusion Model for Images

Generative Process: \( p_\theta(x_{t-1} \mid x_t) \)
Diffusion Model for Images

Generative Process: \[ p_\theta(x_{t-1} \mid x_t) \]

\( x_T \)

Gaussian Noise
Diffusion Model for Images

Generative Process: $p_\theta(x_{t-1} \mid x_t)$
Diffusion Model for Images

Generative Process:

\[ p_\theta(x_{t-1} \mid x_t) \]

\[ X_T \rightarrow \cdots \rightarrow X_t \rightarrow X_{t-1} \]

Gaussian Noise
Diffusion Model for Images

Generative Process:

\[ p_\theta(x_{t-1} \mid x_t) \]

\[ X_T \rightarrow \cdots \rightarrow X_t \rightarrow X_{t-1} \rightarrow \cdots \rightarrow X_0 \]

Gaussian Noise
Diffusion Model for Images

Generative Process:

\[ p_\theta(x_{t-1} \mid x_t) \]

\[ x_T \rightarrow \cdots \rightarrow x_t \rightarrow x_{t-1} \rightarrow \cdots \rightarrow x_0 \]
Diffusion Model for Images

Training: Construct latent variables pairs, then apply supervised training.

\[ q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I) \]
Training: Construct latent variables pairs, then apply supervised training.

Supervised Training: $L_{\text{simple}}(x_0) = \sum_{t=1}^{T} \mathbb{E}_{q(x_t \mid x_0)} \left\| \mu_{\theta}(x_t, t) - \mu_{t-1}(x_t, x_0) \right\|^2$

\[
\mu_{t-1}(x_t, x_0) = \mathbb{E}_{q(x_{t-1} \mid x_t = x_t, x_0 = x_0)}[x_{t-1}]
\]
1. What is diffusion model?

2. Apply diffusion to text

3. Discussion
   1. Distinction btw text and images
   2. Compare with autoregressive LM
Diffusion Model for Discrete Text

\[ p_\theta(x_{t-1} \mid x_t) \]
\[ q(x_t \mid x_{t-1}) \]

Gaussian Noise

Continuous

Discrete Text

\[ X_T \rightarrow \cdots \rightarrow X_t \rightarrow X_{t-1} \rightarrow \cdots \rightarrow X_0 \]
Diffusion Model for Discrete Text

Gaussian Noise

$X_T \rightarrow \cdots \rightarrow X_t \xrightarrow{p_{\theta}(x_{t-1} \mid x_t)} X_{t-1} \rightarrow \cdots \rightarrow X_0$

Denoising

$q(x_t \mid x_{t-1})$

Noising

Continuous

$q_{\phi}(x_0 \mid \mathbf{w})$

Embedding

Discrete

Text

$W$
Diffusion Model for Discrete Text

$X_T \rightarrow \cdots \rightarrow X_t \rightarrow X_{t-1} \rightarrow \cdots \rightarrow X_0 \rightarrow W$

- **Gaussian Noise**
- **Denoising**: $p_\theta(x_{t-1} \mid x_t)$
- **Noising**: $q(x_t \mid x_{t-1})$
- **Continuous**
- **Rounding**: $p_\theta(w \mid x_0)$
- **Embedding**: $q_\phi(x_0 \mid w)$
- **Discrete**

Text
Embedding

\[ \text{EMB} (w_i) \] maps each word to \( \mathbb{R}^d \).
\[ \text{EMB}(w) = [\text{EMB}(w_1), \ldots, \text{EMB}(w_n)] \in \mathbb{R}^{nd} \]

\[ q_\phi(x_0 | w) = \mathcal{N}(\text{EMB}(w), \sigma_0 I) \]

\( \sigma_0 \) is a hyper parameter (a small number)

How to choose the embedding function \( \text{EMB}(w_i) \)

\[ x_0 \in \mathbb{R}^{nd} \]
\[ x_t \in \mathbb{R}^{nd} \]
\[ x_T \in \mathbb{R}^{nd} \]
Embedding

How to choose the embedding function $\text{EMB}(w_i)$

1. Random Embedding?
2. Learn it end-to-end

$x_0 \in \mathbb{R}^{n_d}$
$x_t \in \mathbb{R}^{n_d}$
$x_T \in \mathbb{R}^{n_d}$
Embedding: End-to-end Training

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

maximize probability of Embedding

reconstruction loss
Embedding: End-to-end Training

maximize probability of Embedding

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

reconstruction loss

\[
\mathcal{L}_{\text{simple}}(x_0) = \sum_{t=1}^T \mathbb{E}_{q(x_t \mid x_0)} \left\| \mu_\theta(x_t, t) - \mu_{t-1}(x_t, x_0) \right\|^2
\]
Embedding: End-to-end Training

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

maximize probability of Embedding

reconstruction loss

$$p_\theta(w \mid x_0) = \prod_{i=1}^{n} p_\theta(w_i \mid x_i)$$
Embedding: End-to-end Training

Figure 3: A t-SNE [41] plot of the learned word embeddings. Each word is colored by its POS.
Rounding

\[ \hat{w}_i = \arg\max_{w_i} p_\theta(w_i \mid x_0[i]) \]

😊 Ideally, the model should predict \( x_0 \) that lies exactly on a word embedding.

😢 Reality: there is still rounding error.
Reducing Rounding Error

\[ \hat{w}_i = \arg\max_{w_i} p_\theta(w_i | x_0[i]) \]

😊 Ideally, the model should predict \(x_0\) that lies exactly on a word embedding.

😢 Reality: there is still rounding error.

My ice cream is [BLANK].

- melting ✅
- saving ❌

“melting” and “saving” are close in the embedding space. But they are not substitutable in this context.
Reducing Rounding Error (training time)

\[ \mathcal{L}_{\text{simple}}(x_0) = \sum_{t=1}^{T} \mathbb{E}_{q(x_t|x_0)} \left| \left| \hat{\mu}_\theta(x_t, t) - \mu_{t-1}(x_t, x_0) \right| \right|^2 \]
Reducing Rounding Error (training time)

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

Intuition: Rounding happens all at the last step: \( x_1 \to x_0 \) which is hard and prone to error.
Reducing Rounding Error (training time)

\[ \mathcal{L}_{\text{simple}}(x_0) = \sum_{t=1}^{T} \mathbb{E}_{q(x_t|x_0)} \left\| \hat{f}_\theta(x_t, t) - x_0 \right\|^2 \]
Reducing Rounding Error (training time)

\[ \mathcal{L}_{\text{simple}}(x_0) = \sum_{t=1}^{T} \mathbb{E}_{q(x_t|x_0)} \left\| \hat{f}_\theta(x_t, t) - x_0 \right\|^2 \]

Intuition: Training to predict \( x_0 \) at each diffusion step makes predicted \( x_0 \) more precise and reduce rounding error.
Diffusion-LM: Diffusion based Language Models

1. What is diffusion model?

2. Apply diffusion to text

3. Discussion
   1. Distinction btw text and images
   2. Compare with autoregressive LM
Discussion: Text v.s. Images

High frequency v.s. Low frequency

Low frequency ➡ the large scale structure of the images
Medium frequency ➡ the details.
High frequency ➡ perceptually meaningless.

For text, what does high frequency even mean?
Discussion: Text v.s. Images

High frequency v.s. Low frequency

For text, what does high frequency even mean?

e.g., the frequency can be defined as the rate of change in the embedding of the neighboring words.

The most natural way to capture high frequency components is to do next token prediction. But diffusion LM can still benefit from the low frequency (coarse-grained) planning…
Discussion: Diffusion v.s. Autoregressive
Discussion: Diffusion v.s. Autoregressive

How human writes long text:
Core concepts ➡ writing structure ➡ wording and phrasing
Discussion: Diffusion v.s. Autoregressive

How human writes long text:
Core concepts ➡ writing structure ➡ wording and phrasing

Very different

How autoregressive LM produce long text:
Left-to-right.
Discussion: Diffusion v.s. Autoregressive

How human writes long text:
Core concepts ➡️ writing structure ➡️ wording and phrasing

How autoregressive LM produce long text:
Left-to-right.

How diffusion LM produce long text:
Coarse-to-fine.
Discussion: Diffusion v.s. Autoregressive

Training Efficiency
Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM can be regarded as a special form of iterative refinement, where each step refines the next token.
Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM can be regarded as a special form of iterative refinement, where each step refines the next token.

During training, autoregressive LM can obtain gradient signal from each refinement step, because we can parallelize all tokens by causal masking.
Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM can be regarded as a special form of iterative refinement, where each step refines the next token.

During training, autoregressive LM can obtain gradient signal from each refinement step, because we can parallelize all tokens by causal masking.

Diffusion-LM’s refinement step denoises a particular noise level.
Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM can be regarded as a special form of iterative refinement, where each step refines the next token.

During training, autoregressive LM can obtain gradient signal from each refinement step, because we can parallelize all tokens by causal masking.

Diffusion-LM’s refinement step denoises a particular noise level.

For each forward pass, diffusion-LM can only obtain gradient signal from one noise level.
Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM can be regarded as a special form of iterative refinement, where each step refines the next token.

During training, autoregressive LM can obtain gradient signal from each refinement step, because we can parallelize all tokens by causal masking.

Diffusion-LM’s refinement step denoises a particular noise level.

For each forward pass, diffusion-LM can only obtain gradient signal from one noise level.

Autoregressive LM > diffusion-LM
Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM > diffusion-LM

Decoding Efficiency
Discussion: Diffusion v.s. Autoregressive

**Training Efficiency**

Autoregressive LM > diffusion-LM

**Decoding Efficiency**

Autoregressive LM: O(sequence length)

Diffusion-LM: O(number of diffusion steps)
Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM > diffusion-LM

Decoding Efficiency

Autoregressive LM: O(sequence length)

Diffusion-LM: O(number of diffusion steps)

Diffusion-LM might become more beneficial with longer text.
Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM: $O(\text{sequence length})$

Decoding Efficiency

Autoregressive LM: $O(\text{sequence length})$

Diffusion-LM: $O(\text{number of diffusion steps})$

Diffusion-LM might become more beneficial with longer text.

But not all steps are equally costly.
For autoregressive LM, we can use caching;
For diffusion-LM, each step has to be recomputed.
Discussion: Diffusion v.s. Autoregressive

Training Efficiency  
Autoregressive LM > diffusion-LM

Decoding Efficiency  
Autoregressive LM > diffusion-LM

Flexible Decoding time steering
Discussion: Diffusion v.s. Autoregressive

Training Efficiency: Autoregressive LM > diffusion-LM

Decoding Efficiency: Autoregressive LM > diffusion-LM

Flexible Decoding time steering

1. Flexible Generation Order
   Diffusion-LM > Autoregressive LM
Discussion: Diffusion v.s. Autoregressive

**Training Efficiency**
- Autoregressive LM > diffusion-LM

**Decoding Efficiency**
- Autoregressive LM > diffusion-LM

Flexible Decoding time steering

1. **Flexible Generation Order**
   - Diffusion-LM > Autoregressive LM

2. **Controllable Text Generation**
   - Diffusion-LM > Autoregressive LM (Spoiler)
Text Generation

Pretrained LM

Sample from the LM

e.g., GPT-3
Text Generation

- Harry Potter is graduated from Hogwarts...
- Starbucks is a great coffee shop originated from Seattle, ...
- Once upon a time, there are three little pigs...
- To demonstrate the effectiveness of the method, the doctors conducted...

Sample from the LM

Pretrained LM

e.g., GPT-3
Text Generation

Pretrained LM

Sample from the LM

e.g., GPT-3

Generated Text

Harry Potter is graduated from Hogwarts…
Starbucks is a great coffee shop originated from Seattle, …
Once upon a time, there are three little pigs…
To demonstrate the effectiveness of the method, the doctors conducted…

How to generate positive reviews about a coffee shop called Coupa?

Coupa is a delicious coffee shop located on Stanford campus.
How to Control Text Generation?

Goal: generate positive reviews about a coffee shop called Coupa?

Plug-and-play Control

$$p(x \mid c) \propto p(x)p(c \mid x)$$

Frozen LM Parameters $$p(x)$$  Pretrained LM

Attempt 1

Once upon a time, there are three little pigs…

Control Critic
How to Control Text Generation?

Goal: generate positive reviews about a coffee shop called Coupa?

Plug-and-play Control

\[ p(x \mid c) \propto p(x)p(c \mid x) \]

**Frozen LM Parameters**

\[ p(x) \]

Pretrained LM

\[ p(c \mid x) \]

Control Critic

Attempt 1

*Once upon a time, there are three little pigs...*
How to Control Text Generation?

Goal: generate positive reviews about a coffee shop called Coupa?

Plug-and-play Control

\[ p(x \mid c) \propto p(x)p(c \mid x) \]

Frozen LM Parameters

\[ p(x) \]

Pretrained LM

Steering Signal

\[ p(c \mid x) \]

Query

Control Critic

Attempt 1

Once upon a time, there are three little pigs…
How to Control Text Generation?

Goal: generate positive reviews about a coffee shop called Coupa?

Plug-and-play Control

\[ p(x \mid c) \propto p(x)p(c \mid x) \]

Frozen LM Parameters

\[ p(x) \]

Pretrained LM

Attempt 1

Once upon a time, there are three little pigs…

Attempt 2

Starbucks is a great coffee shop originated from Seattle, …

Control Critic
How to Control Text Generation?

Goal: generate positive reviews about a coffee shop called Coupa?

Plug-and-play Control

\[ p(x \mid c) \propto p(x)p(c \mid x) \]

Frozen LM Parameters \[ p(x) \]

Pretrained LM

Steering Signal

\[ p(c \mid x) \]

Control Critic

Attempt 1

Once upon a time, there are three little pigs…

Attempt 2

Starbucks is a great coffee shop originated from Seattle, …
How to Control Text Generation?

Goal: generate positive reviews about a coffee shop called Coupa?

Plug-and-play Control

Frozen LM Parameters $p(x)$

Pretrained LM

$p(x | c) \propto p(x)p(c | x)$

Attempt 1

*Once upon a time, there are three little pigs…*

Attempt 2

*Starbucks is a great coffee shop originated from Seattle, …*

Attempt 3

*Coupa is a great coffee shop originated from CA, …*
How to Control Text Generation?

Goal: generate positive reviews about a coffee shop called Coupa?

Plug-and-play Control

\[ p(x | c) \propto p(x)p(c | x) \]

Frozen LM Parameters \( p(x) \)

Pretrained LM

Steering Signal

Control Critic

\( p(c | x) \)

Query

Attempt 1

\textit{Once upon a time, there are three little pigs…}

Attempt 2

\textit{Starbucks is a great coffee shop originated from Seattle, …}

Attempt 3

\textit{Coupa is a great coffee shop originated from CA, …}
How to Control Text Generation?

Goal: generate positive reviews about a coffee shop called Coupa?

Plug-and-play Control

$p(x) \propto p(x)p(c \mid x)$

Frozen LM Parameters $p(x)$

Pretrained LM

Steering Signal

Attempt 1

Once upon a time, there are three little pigs…

Attempt 2

Starbucks is a great coffee shop originated from Seattle, …

Attempt 3

Coupa is a great coffee shop originated from CA, …

Control Critic

$p(c \mid x)$

Query

👍 lightweight. Frozen LM; only need to specify the classifier.
How to Control Text Generation?

Goal: generate positive reviews about a coffee shop called Coupa?

Plug-and-play Control

\[ p(x \mid c) \propto p(x)p(c \mid x) \]

Frozen LM Parameters

\[ p(x) \]

Pretrained LM

Steering Signal

\[ p(c \mid x) \]

Satisfaction Critic

Positive Sentiment

Semantic Content

Attempt 1

Once upon a time, there are three little pigs…

Attempt 2

Starbucks is a great coffee shop originated from Seattle, …

Attempt 3

Coupa is a great coffee shop originated from CA, …

✅ lightweight. Frozen LM; only need to specify the classifier.

✅ Enables composition.

Smiley face
Controllable Generation

Goal: sample from $p(x_{0:T}|c)$

Diffusion-LM

Gaussian Noise $X_T$ $X_{T-1}$ $X_{T-2}$ ... $X_0$ Text

Starbucks is a coffee shop.

Critique Model
Controllable Generation

Goal: sample from $p(x_{0:T} | c)$

**Diffusion-LM**

- Gaussian Noise: $X_T$
- Gradually Denoising
- Word Vectors: $X_0$
- Text: $W$
- Starbucks is a coffee shop.

Query Classifier

Gradient Update

Critique Model
Controllable Generation

Goal: sample from $p(x_{0:T}|c)$

**Diffusion-LM**

Gradually Denoising

Gaussian Noise

$X_T \xrightarrow{\text{Gradual Denoising}} X_{T-1} \xrightarrow{} X_{T-2} \xrightarrow{\text{...}} X_0 \xrightarrow{} \text{Word Vectors}$

Text $W$

Starbucks is a coffee shop.

Query Classifier

Gradient Update

Critique Model
Controllable Generation

Goal: sample from $p(x_{0:T} | c)$

Diffusion-LM

Gaussian Noise $\rightarrow$ Gradually Denoising $\rightarrow$ Word Vectors $\rightarrow$ Text

$X_T \rightarrow X_{T-1} \rightarrow X_{T-2} \rightarrow \ldots \rightarrow X_0$

Query Classifier

Gradient Update

Starbucks is a coffee shop.

Critique Model
Controllable Generation

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

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

**Diffusion-LM transition**

**Classifier score**
Controlling Semantic Content

Task:
Given a field (e.g., rating) and value (e.g., 5 star), generate a sentence that covers field=value. Report success rate by exact match of ‘value’.

Example:
- Semantic Content: Food = Japanese
- Output Text: Browns Cambridge is good for Japanese food and also children friendly near The Sorrento.

Results:
- Success Rate (Higher is better)
- Sentence Fluency (Lower is better)
Controlling Semantic Content

Task:
Given a field (e.g., rating) and value (e.g., 5 star), generate a sentence that covers field=value. Report success rate by exact match of ‘value’.

Example:

<table>
<thead>
<tr>
<th>Semantic Content</th>
<th>Output Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food = Japanese</td>
<td>Browns Cambridge is good for Japanese food and also children friendly near The Sorrento.</td>
</tr>
</tbody>
</table>

Results:

Takeaway:
Diffusion-LM outperform other controllable generation baselines, and perform on par with the fine-tuning oracle.
Thanks :)