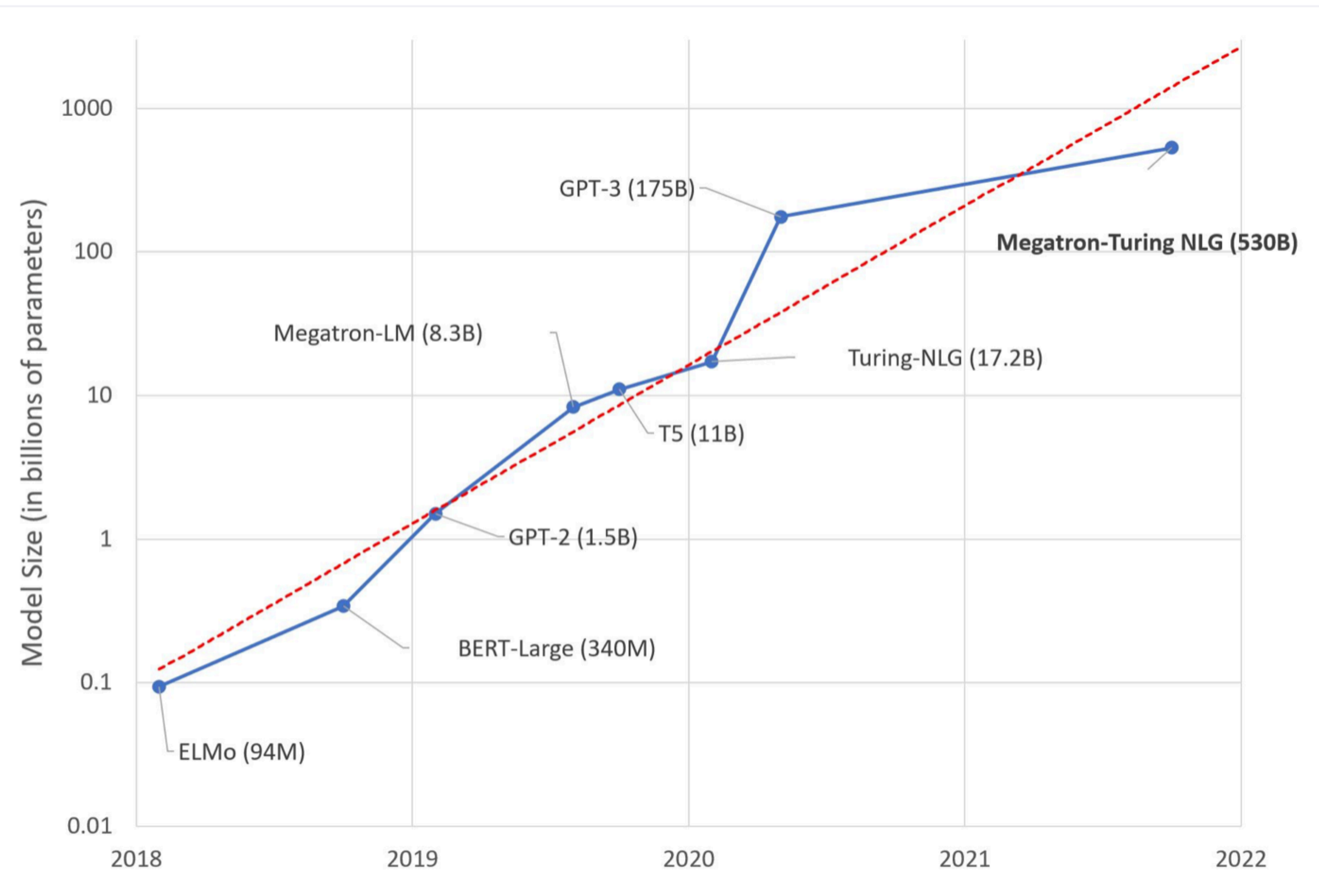


Diffusion Models for Text

Beyond autoregressive language modeling

Xiang Lisa Li

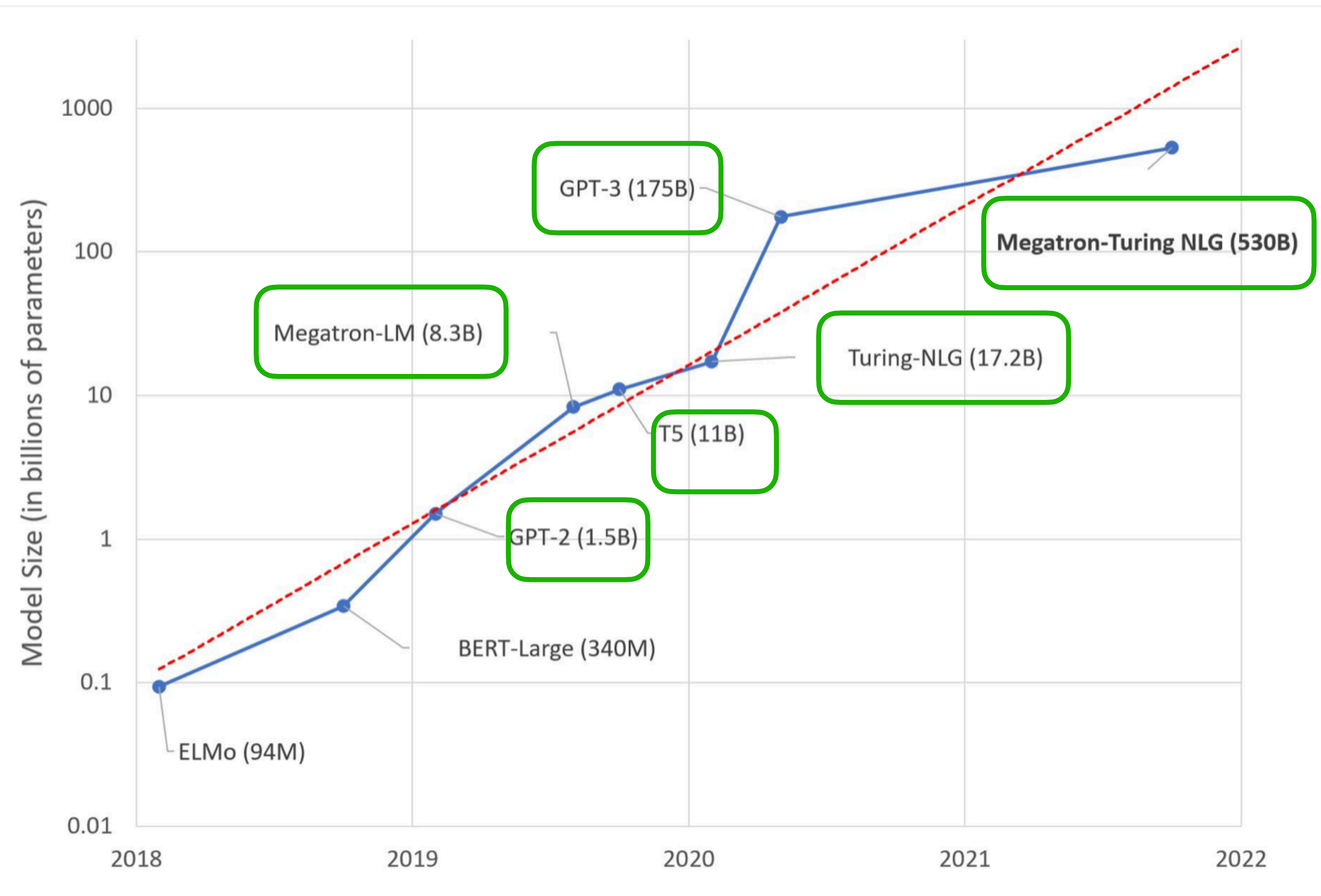
Monopoly of Autoregressive LMs



GPT-2	PaLM	T5-decoder
GPT-3	OPT	Bard
GPT-4	Claude	LLAMA
...		

All models for text generation are autoregressive.

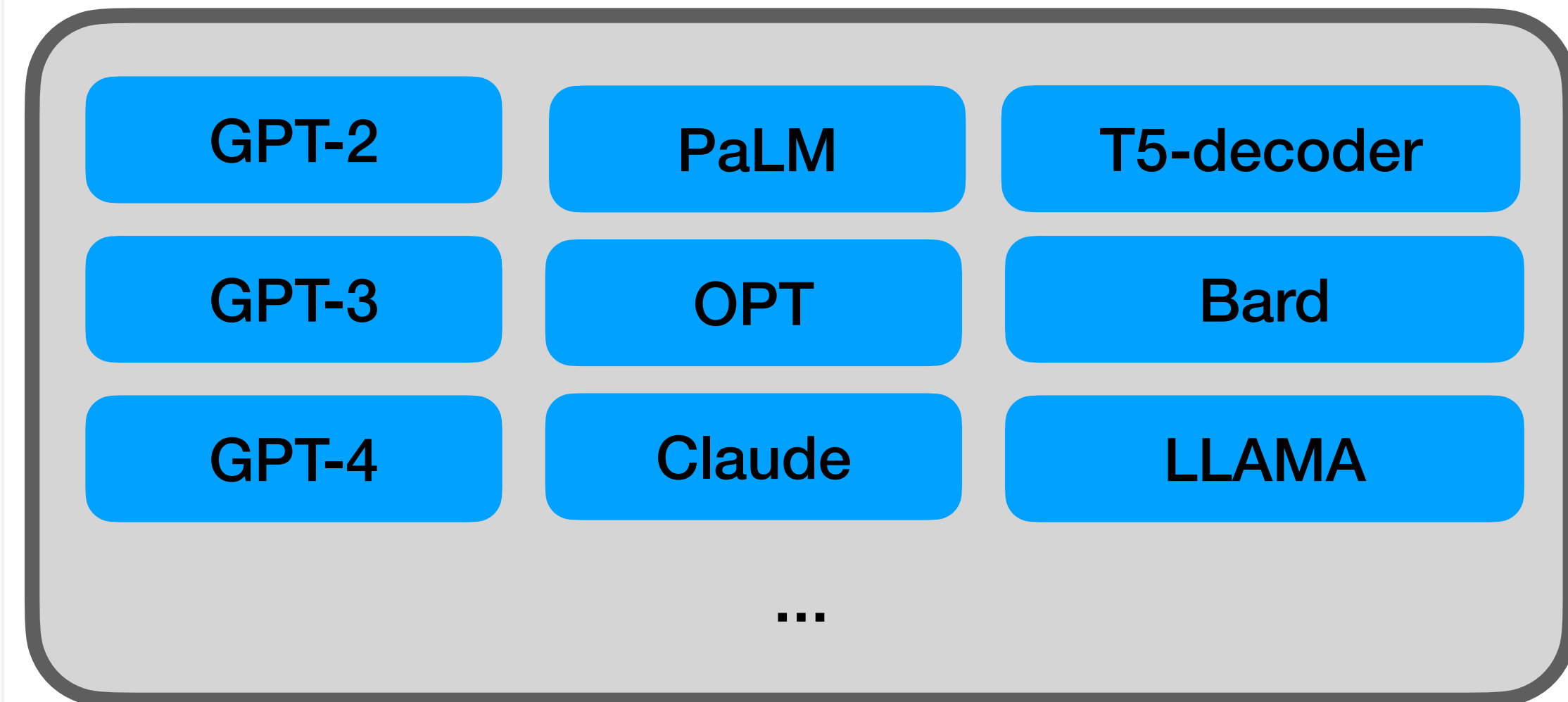
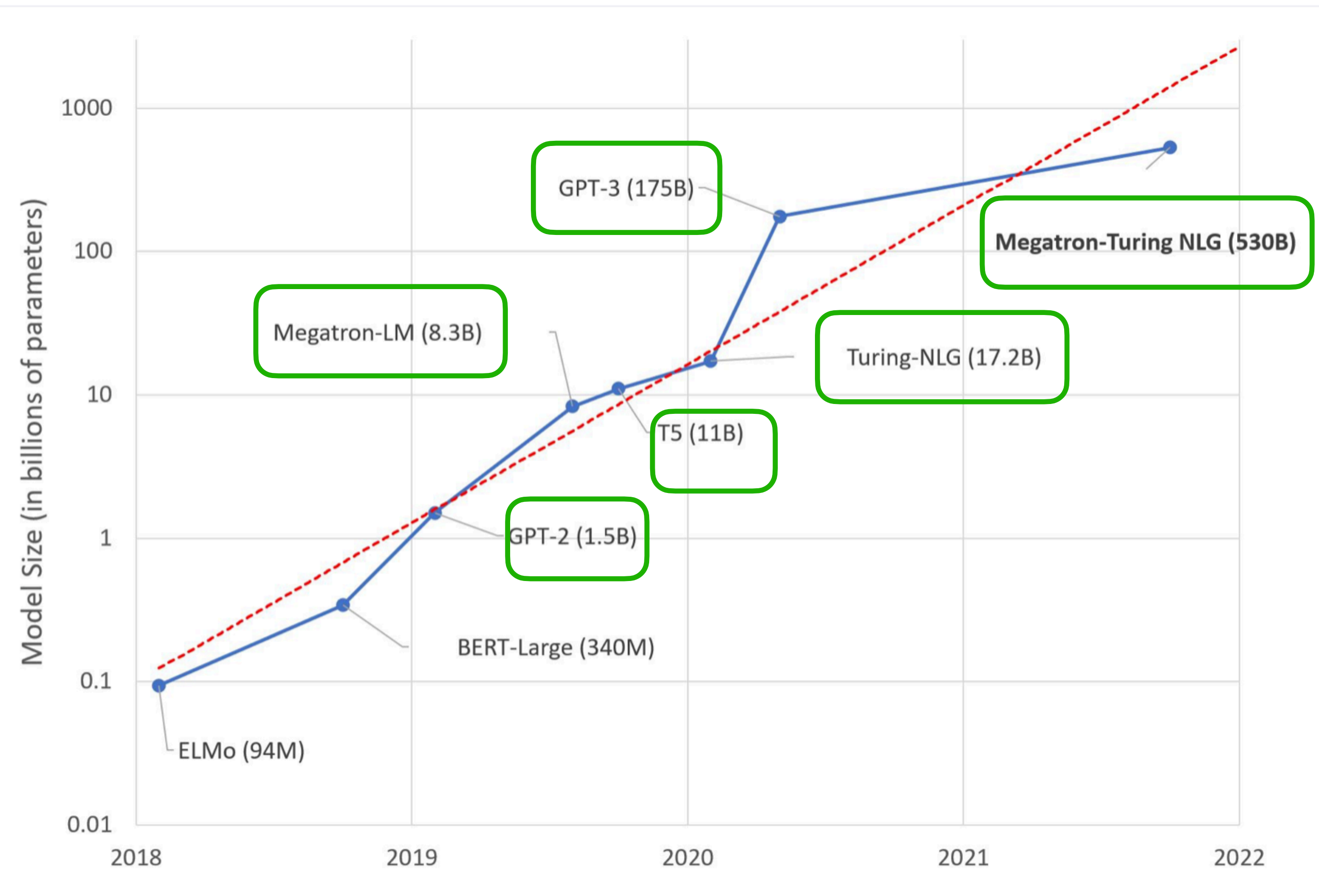
Monopoly of Autoregressive LMs



GPT-2	PaLM	T5-decoder
GPT-3	OPT	Bard
GPT-4	Claude	LLAMA
...		

All models for text generation are autoregressive.

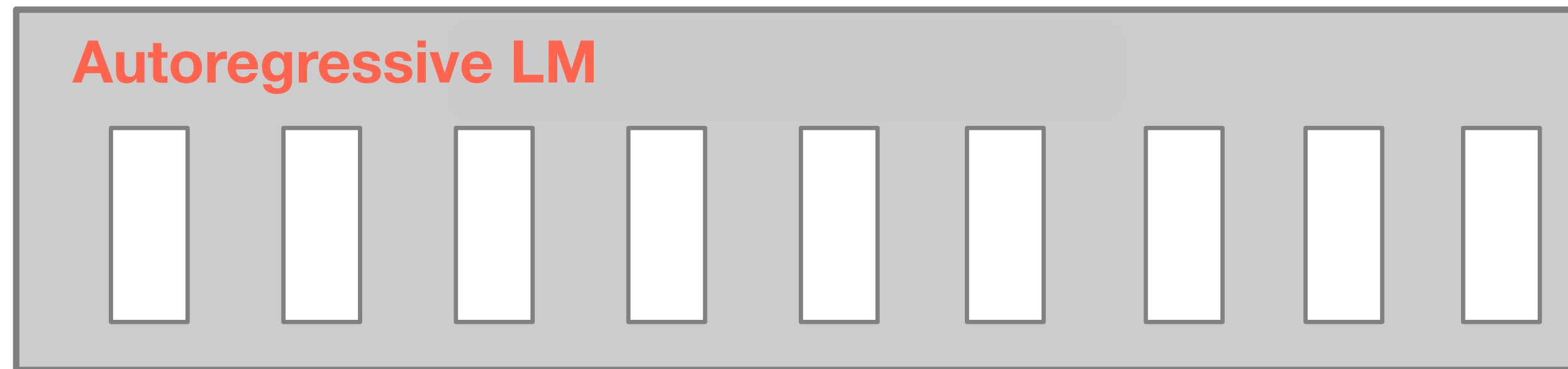
Monopoly of Autoregressive LMs



All models for text generation are autoregressive.

Autoregressive language models have been dominating NLP !

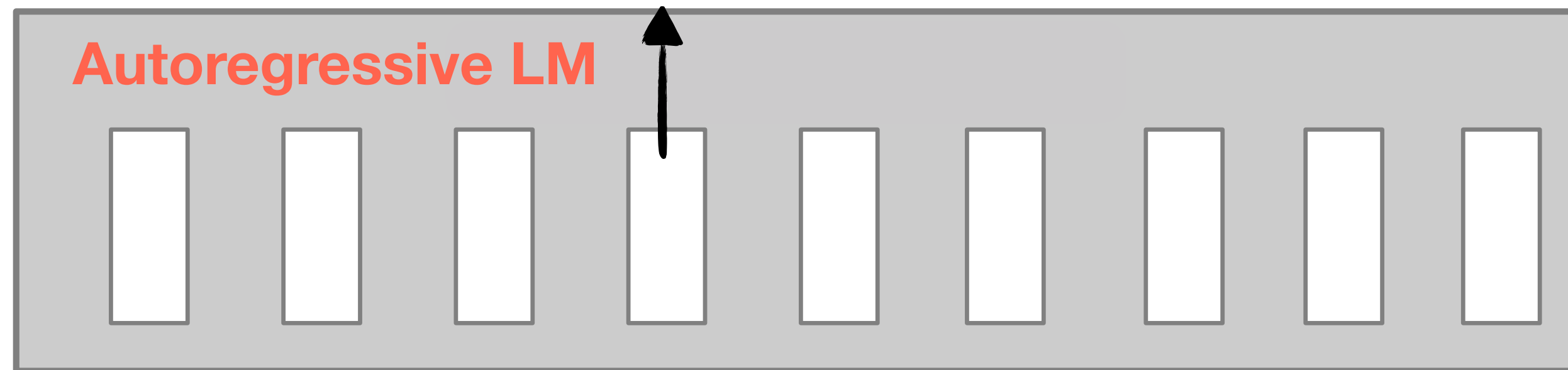
Autoregressive Language Modeling



e.g., GPT-3

Harry Potter graduated from

Autoregressive Language Modeling

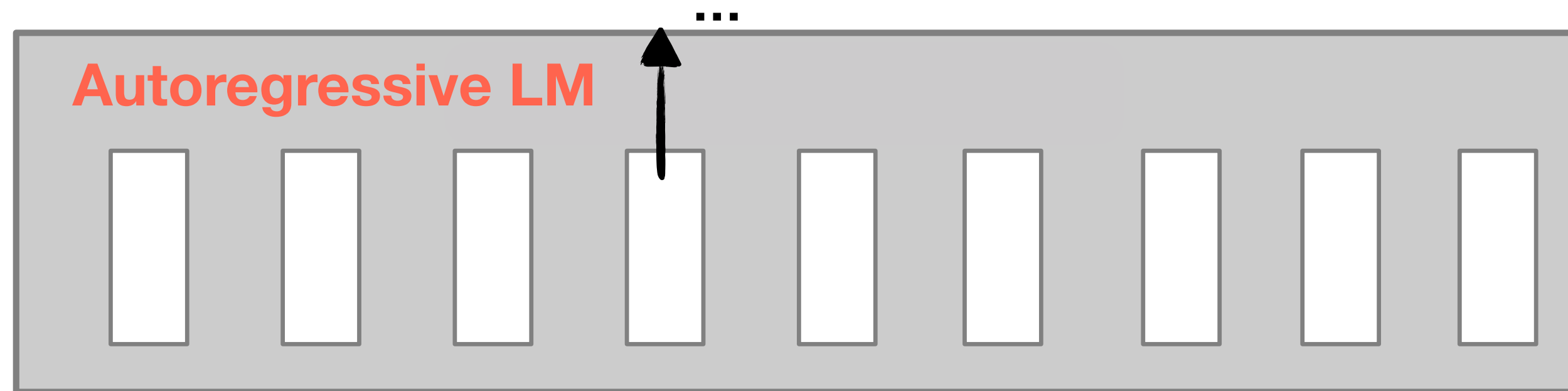


e.g., GPT-3

Harry Potter graduated from

Autoregressive Language Modeling

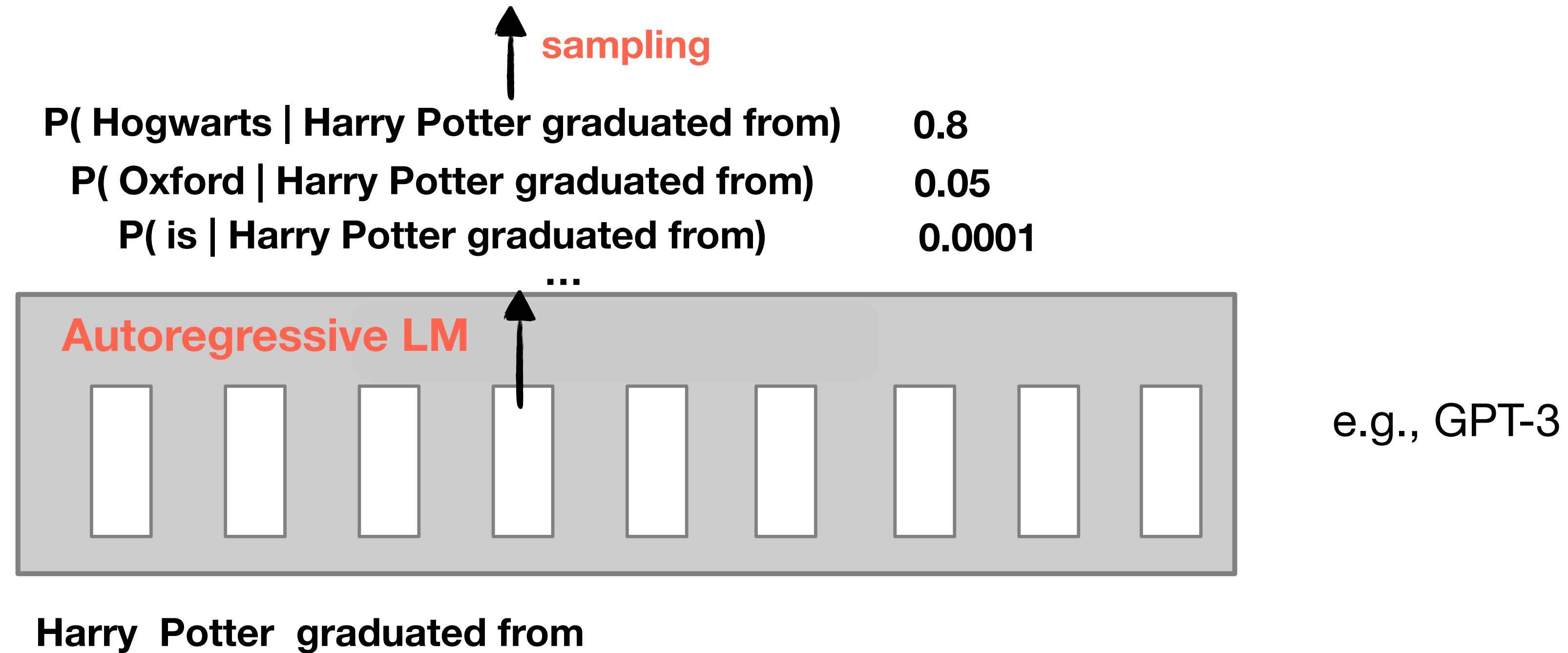
$P(\text{Hogwarts} \mid \text{Harry Potter graduated from})$ 0.8
 $P(\text{Oxford} \mid \text{Harry Potter graduated from})$ 0.05
 $P(\text{is} \mid \text{Harry Potter graduated from})$ 0.0001



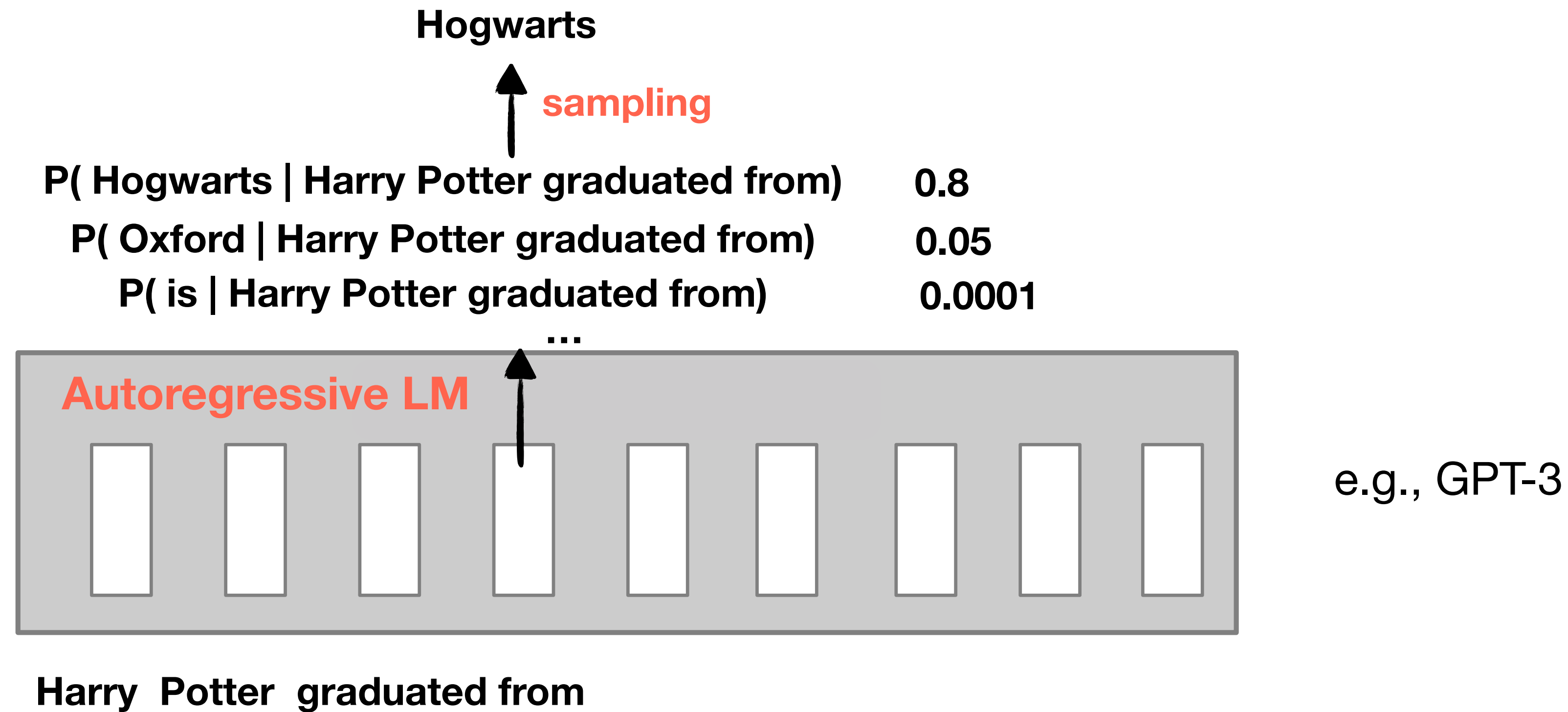
Harry Potter graduated from

e.g., GPT-3

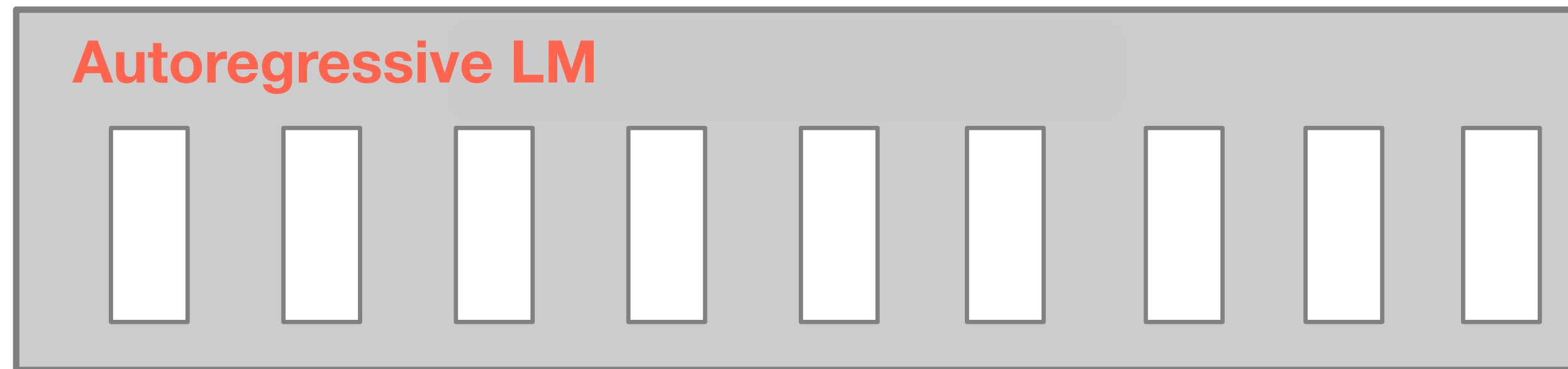
Autoregressive Language Modeling



Autoregressive Language Modeling



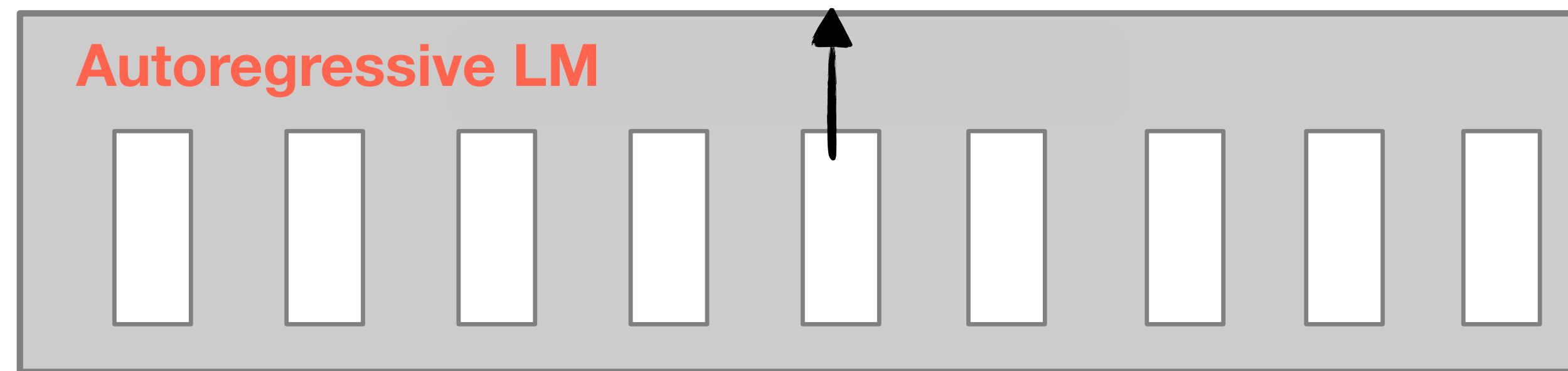
Autoregressive Language Modeling



e.g., GPT-3

Harry Potter graduated from Hogwarts

Autoregressive Language Modeling



e.g., GPT-3

Harry Potter graduated from Hogwarts

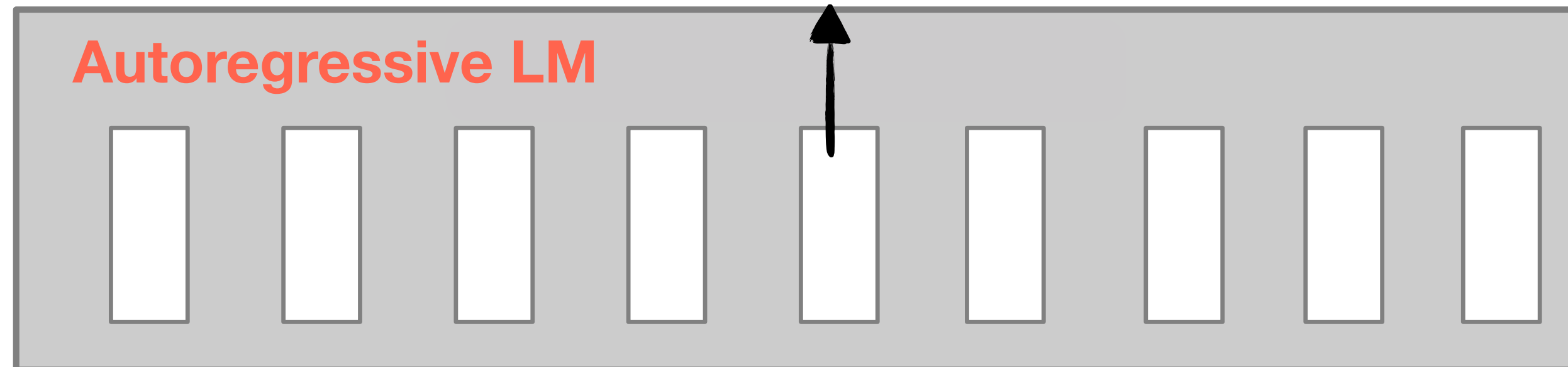
Autoregressive Language Modeling

$P(\text{and} \mid \text{Harry Potter graduated from Hogwarts})$ 0.65

$P(\text{school} \mid \text{Harry Potter graduated from Hogwarts})$ 0.15

$P(. \mid \text{Harry Potter graduated from Hogwarts})$ 0.2

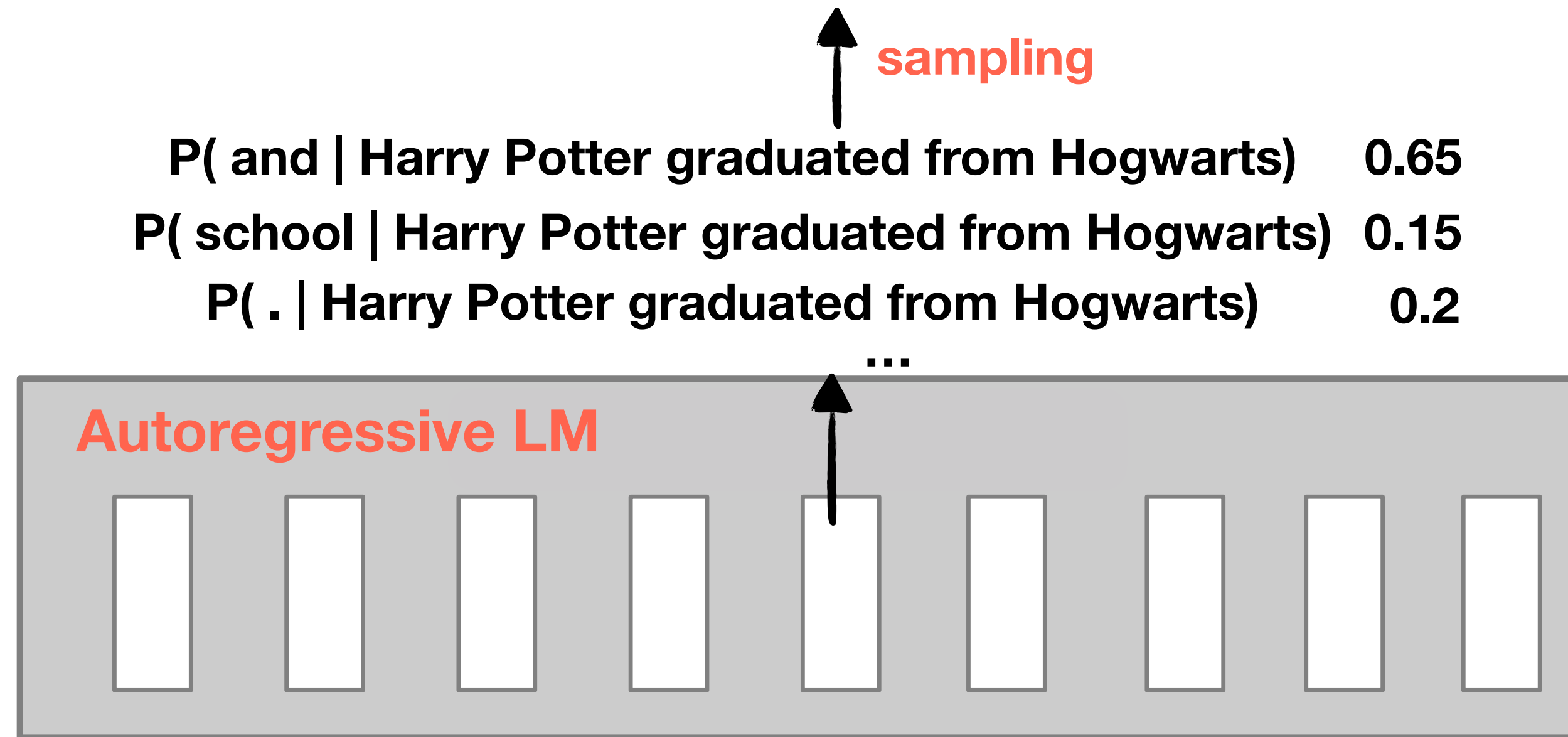
...



Harry Potter graduated from Hogwarts

e.g., GPT-3

Autoregressive Language Modeling



Harry Potter graduated from Hogwarts

e.g., GPT-3

Autoregressive Language Modeling

and

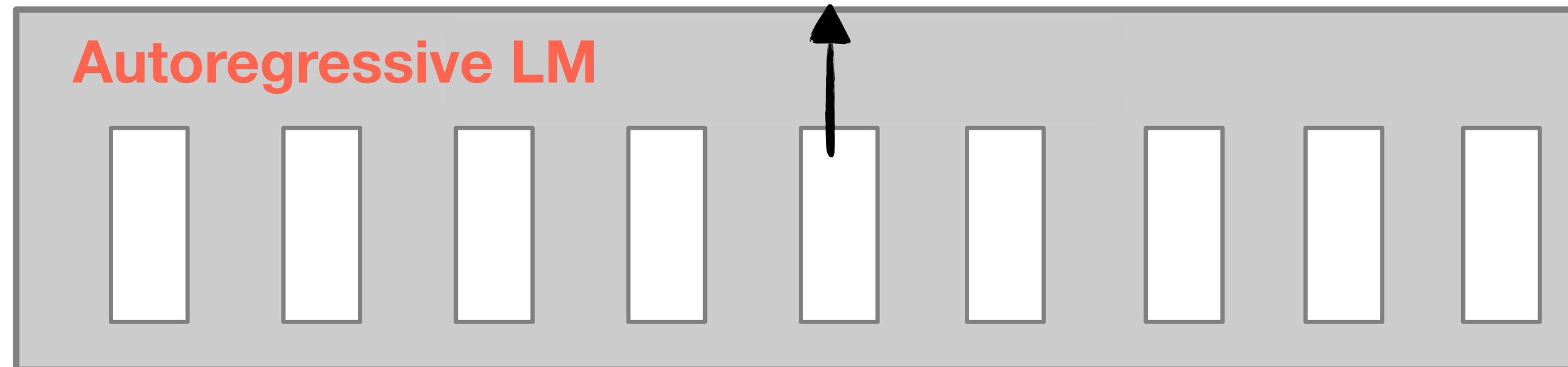
↑ **sampling**

$P(\text{and} \mid \text{Harry Potter graduated from Hogwarts})$ 0.65

$P(\text{school} \mid \text{Harry Potter graduated from Hogwarts})$ 0.15

$P(. \mid \text{Harry Potter graduated from Hogwarts})$ 0.2

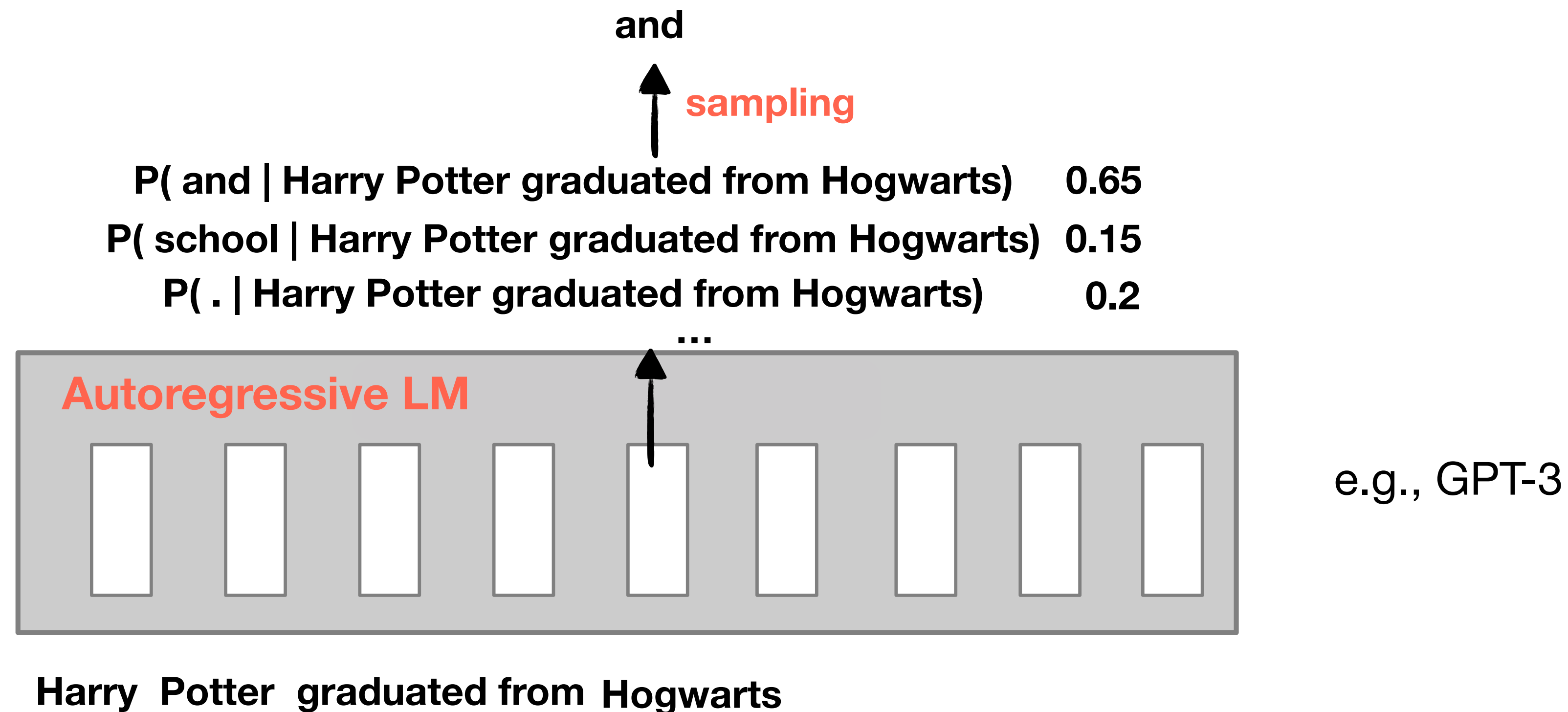
...



e.g., GPT-3

Harry Potter graduated from Hogwarts

Autoregressive Language Modeling



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

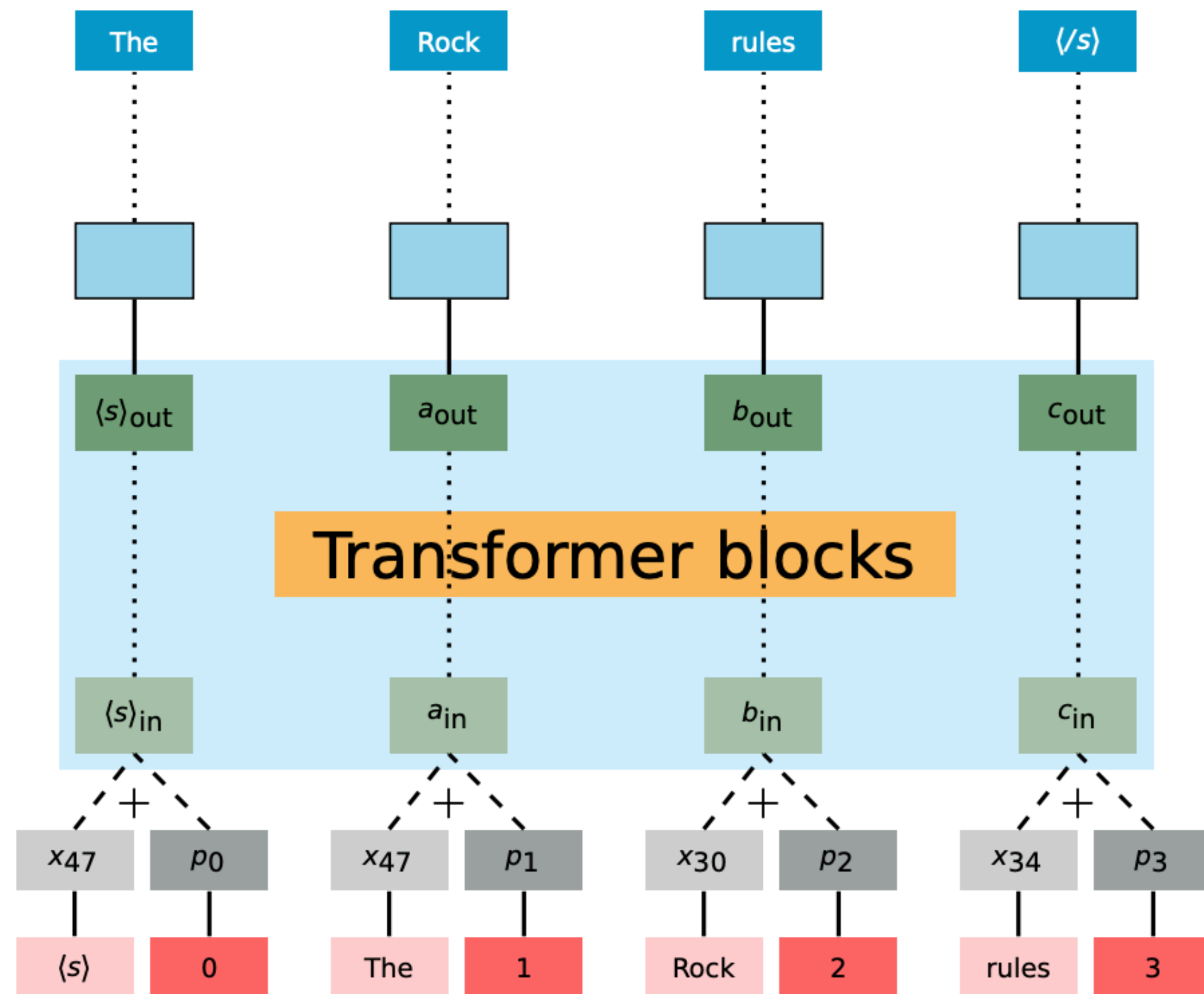
Decoding from Autoregressive LM

Decoding from Autoregressive LM

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

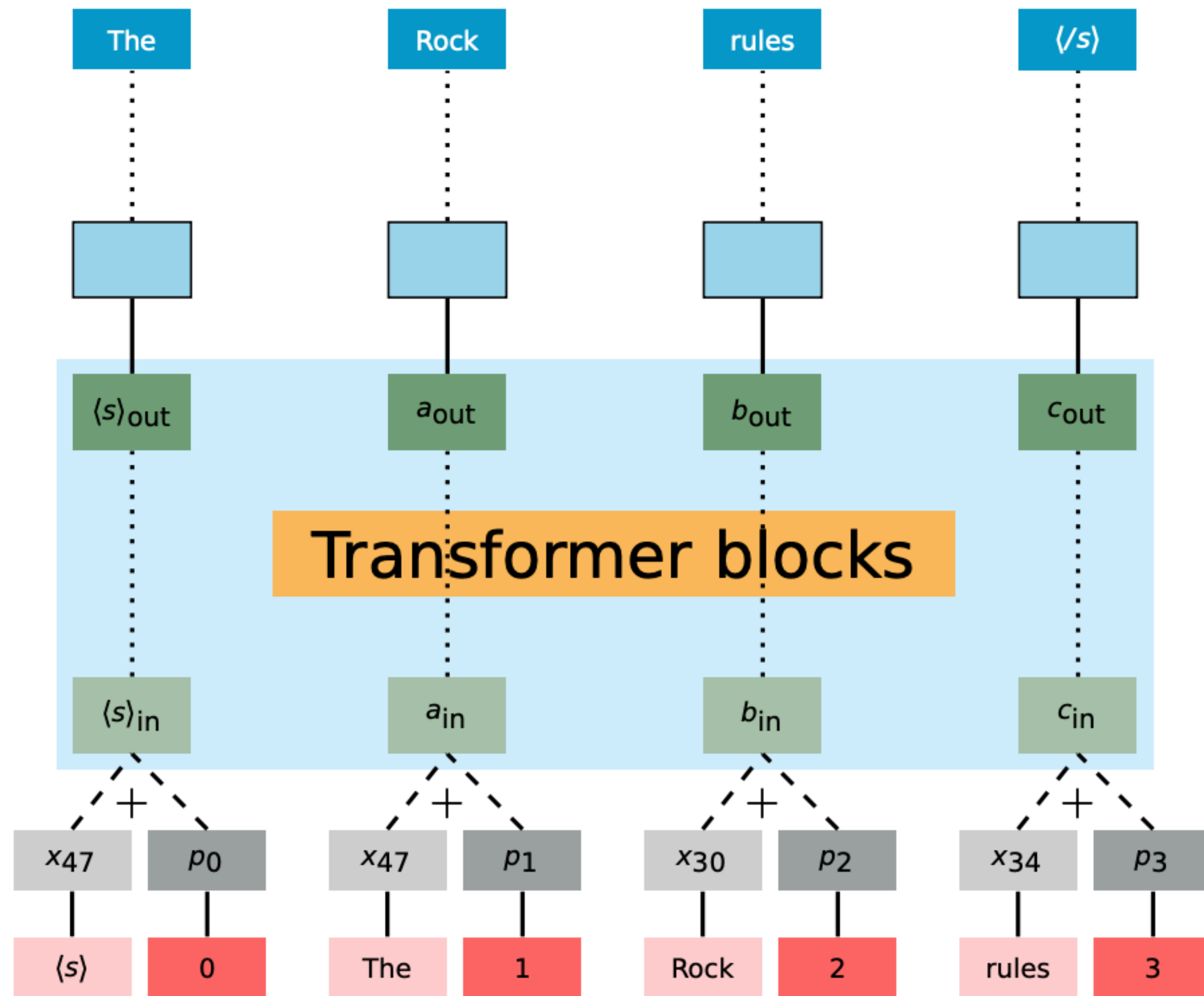
Decoding from Autoregressive LM

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



Decoding from Autoregressive LM

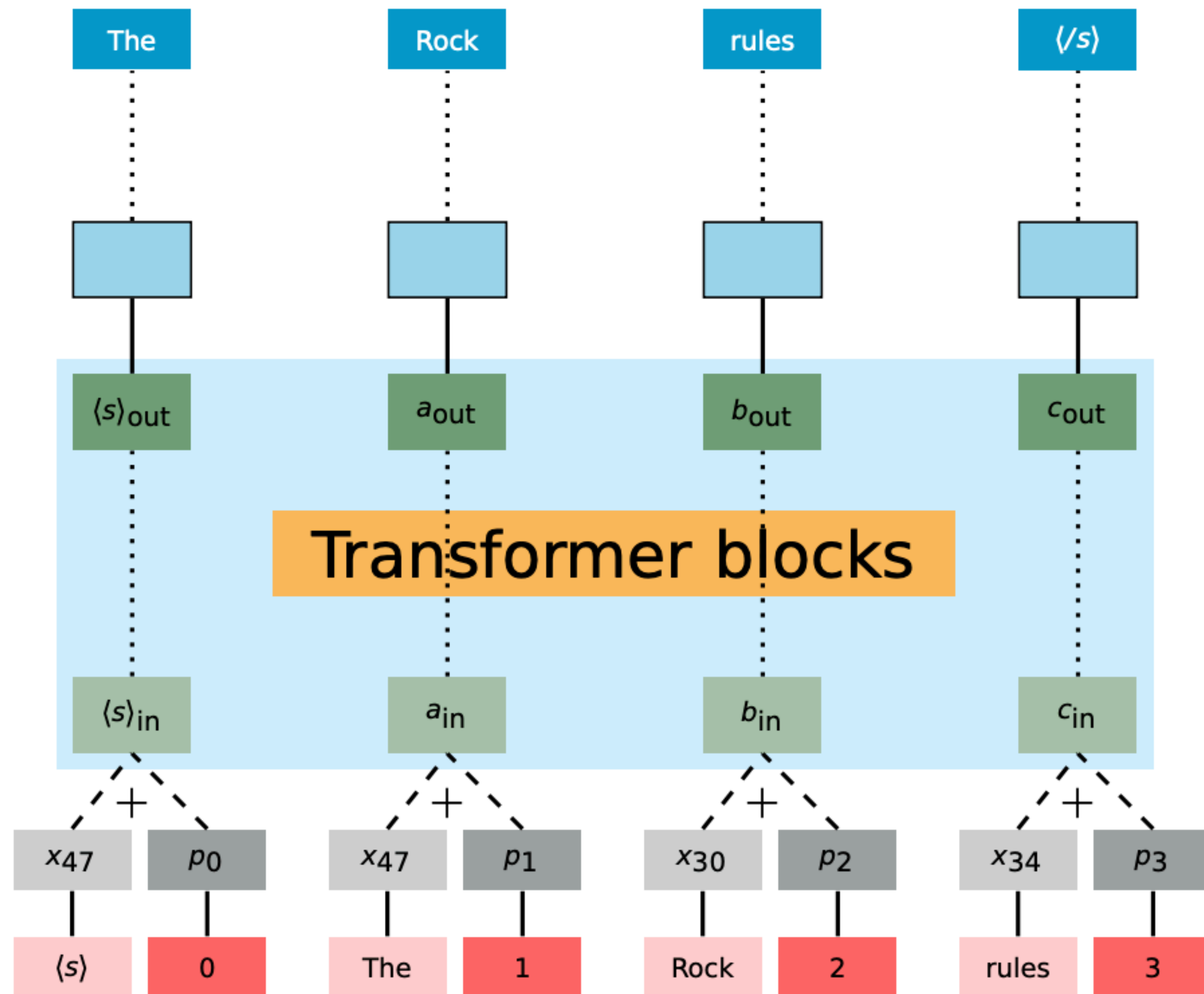
Generating text from left-to-right, one at a time.



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?

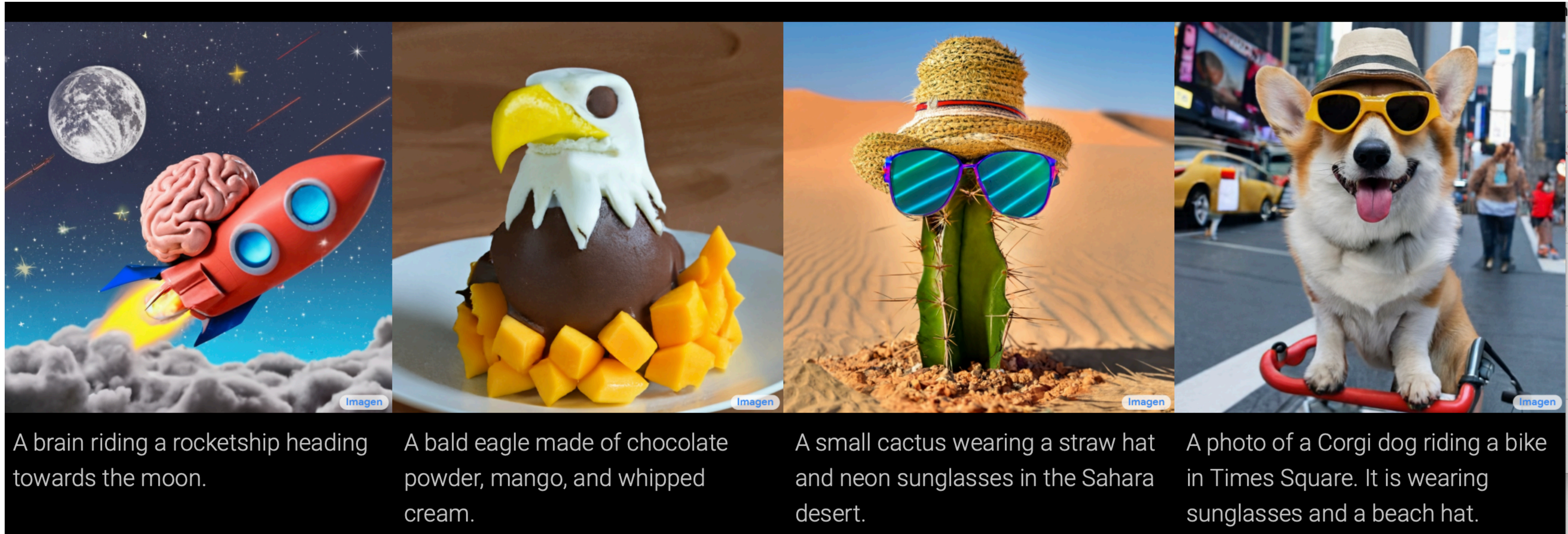
Diffusion-LM: Diffusion based Language Models

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!

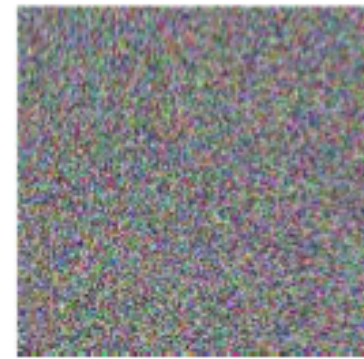
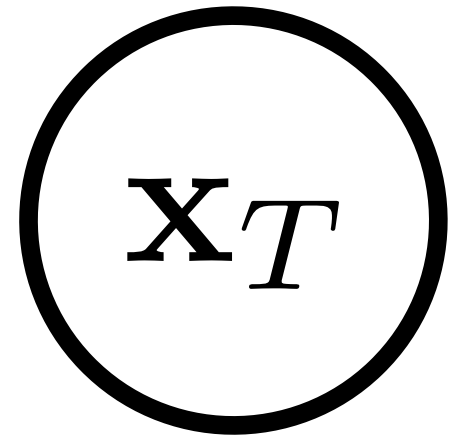


Diffusion Model for Images

Generative Process: $p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$

Diffusion Model for Images

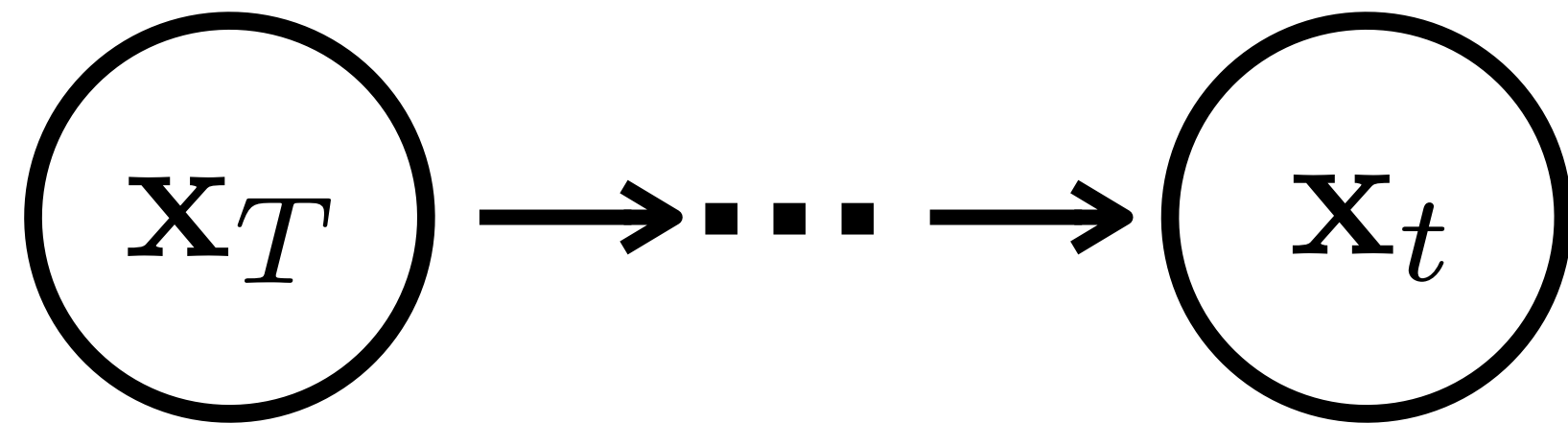
Generative Process: $p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$



Gaussian Noise

Diffusion Model for Images

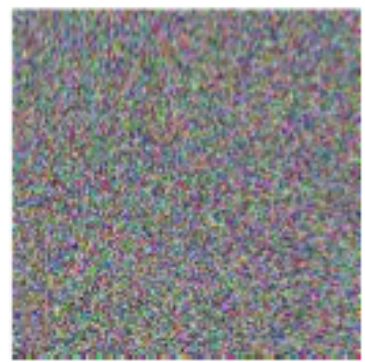
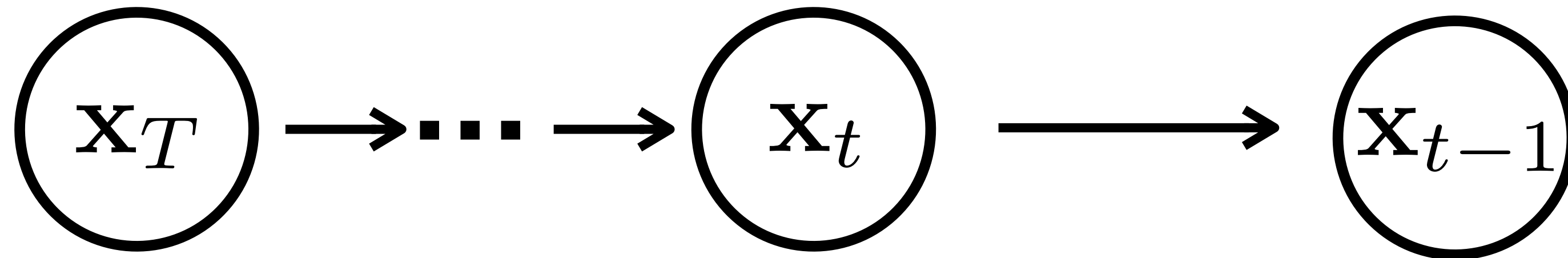
Generative Process: $p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$



Gaussian Noise

Diffusion Model for Images

Generative Process: $p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$

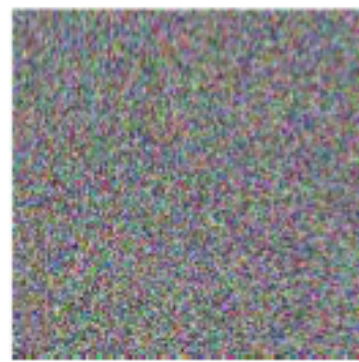
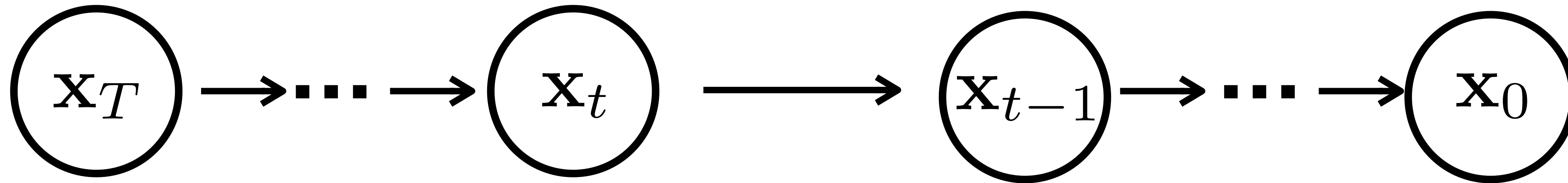


Gaussian Noise



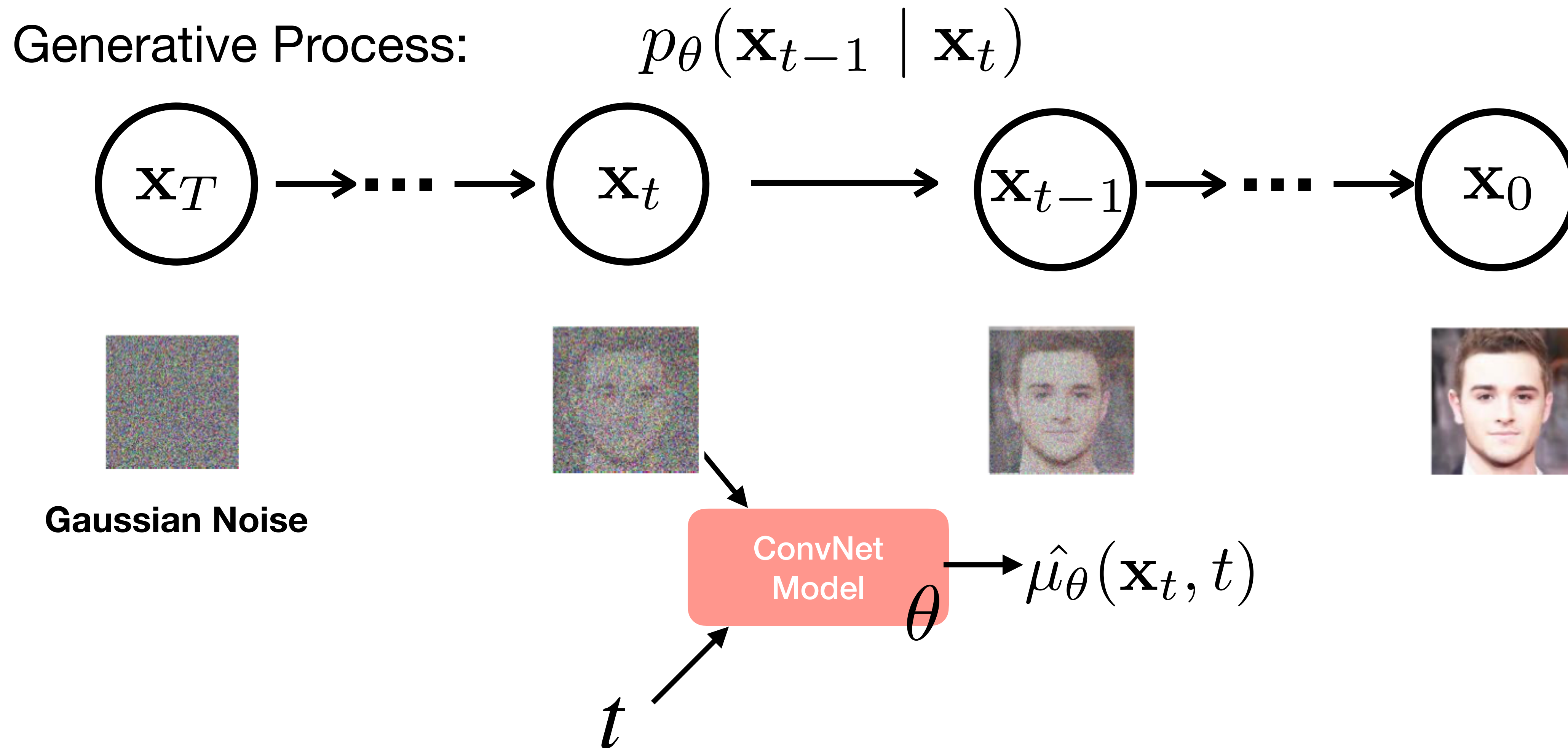
Diffusion Model for Images

Generative Process: $p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$



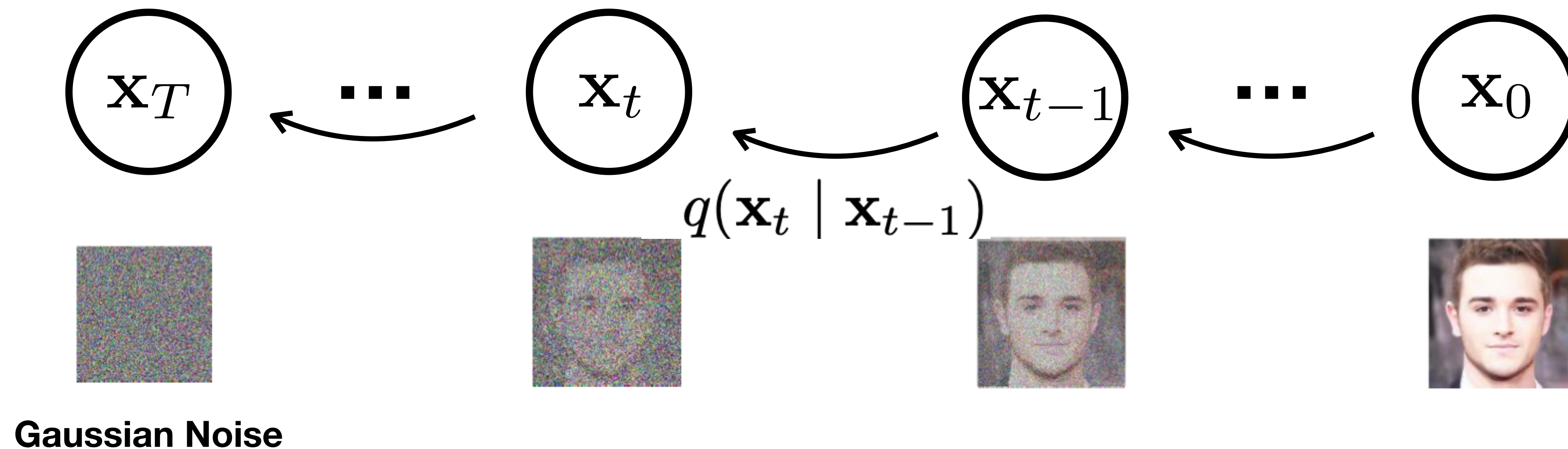
Gaussian Noise

Diffusion Model for Images



Diffusion Model for Images

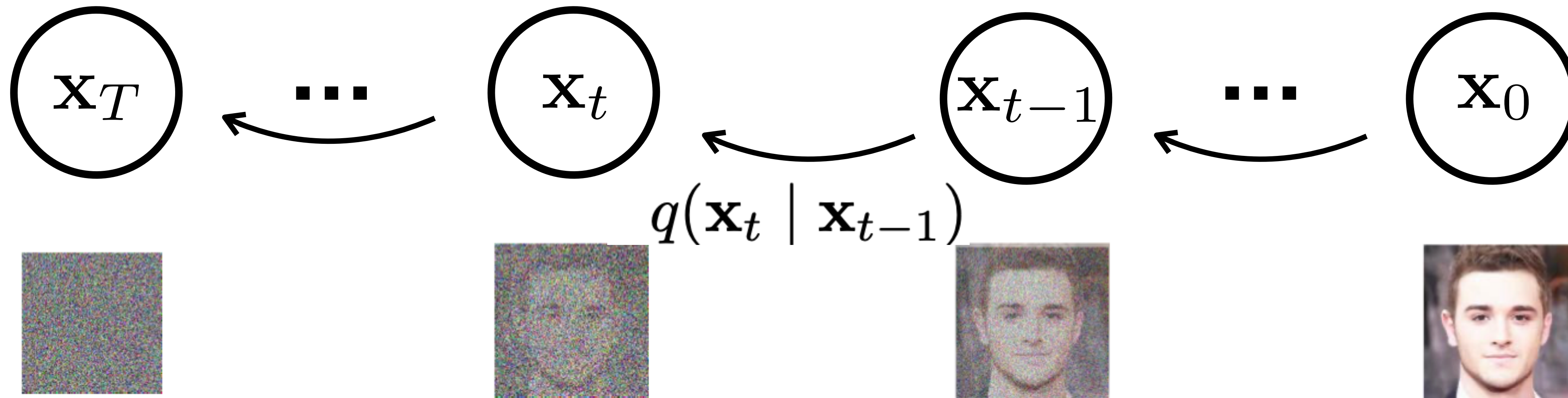
Training: Construct latent variables pairs, then apply supervised training.



Latents Construction: $q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$

Diffusion Model for Images

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



Gaussian Noise

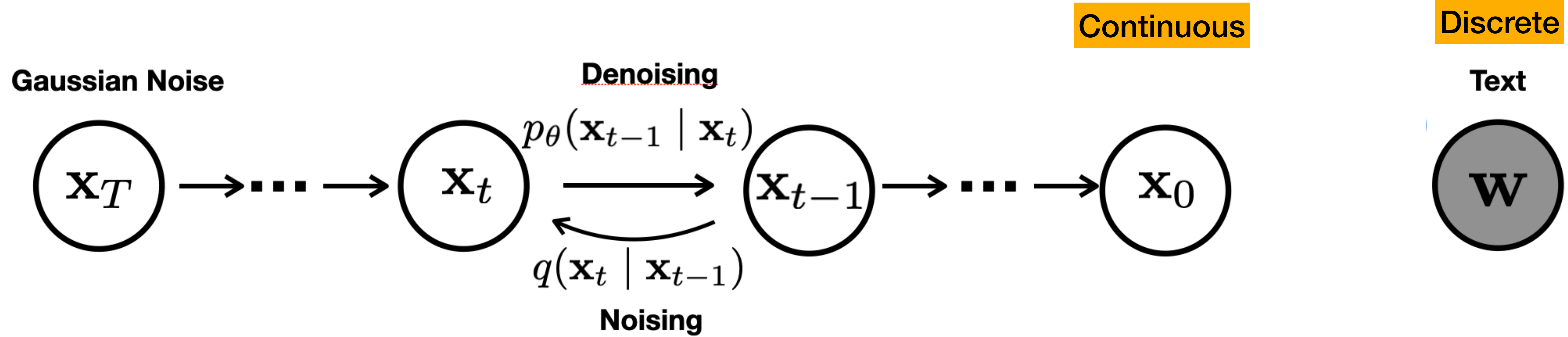
Supervised Training: $\mathcal{L}_{\text{simple}}(x_0) = \sum_{t=1}^T \mathbb{E}_{q(\mathbf{x}_t | \mathbf{x}_0)} \|\hat{\mu}_\theta(\mathbf{x}_t, t) - \mu_{t-1}(\mathbf{x}_t, \mathbf{x}_0)\|^2$

$$\mu_{t-1}(\mathbf{x}_t, \mathbf{x}_0) = \mathbb{E}_q[\mathbf{x}_{t-1} | x_t = \mathbf{x}_t, x_0 = \mathbf{x}_0]$$

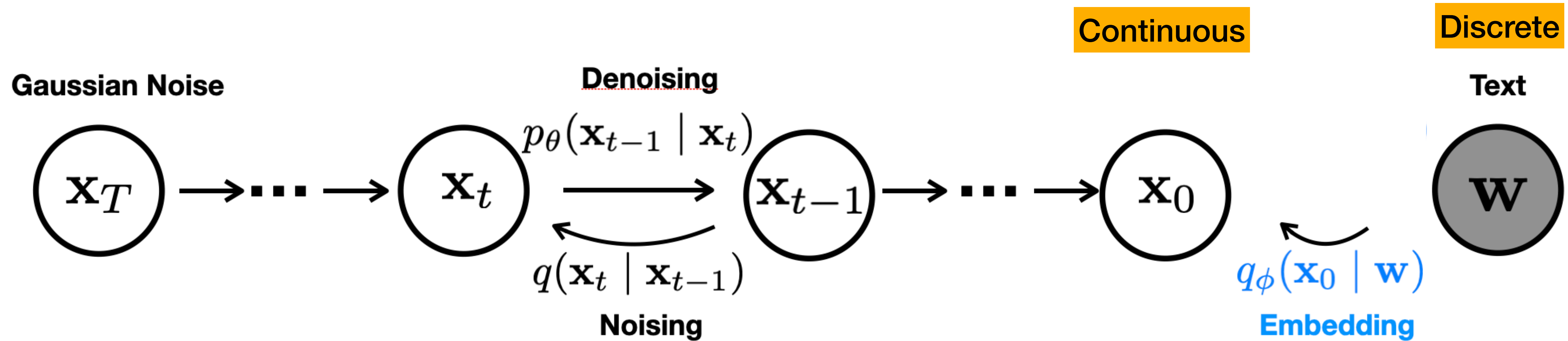
Diffusion-LM: Diffusion based Language Models

1. What is diffusion model?
2. Apply diffusion to text
3. Discussion
 1. Distinction btw text and images
 2. Compare with autoregressive LM

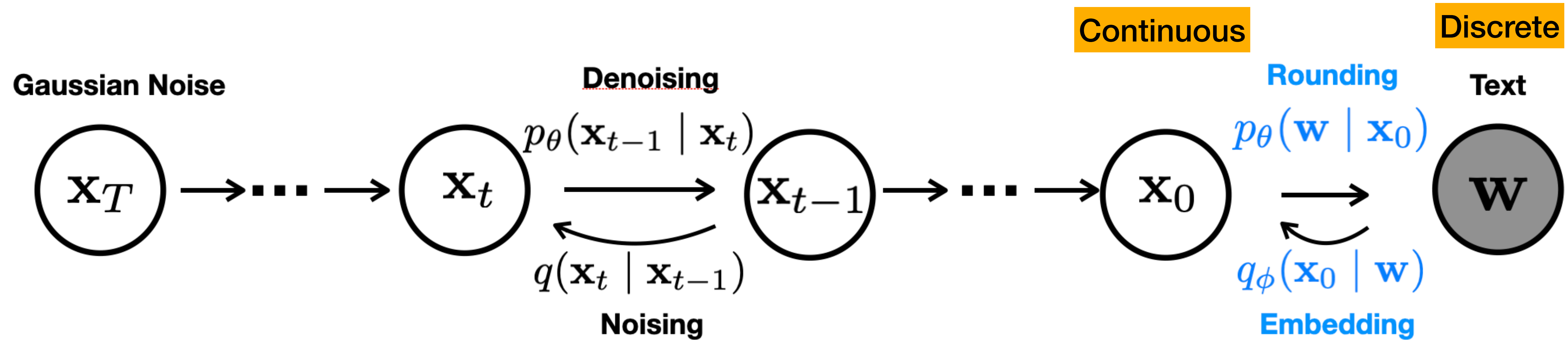
Diffusion Model for Discrete Text



Diffusion Model for Discrete Text



Diffusion Model for Discrete Text



Embedding

$\text{EMB}(w_i)$ maps each word to \mathbb{R}^d .

$$\text{EMB}(\mathbf{w}) = [\text{EMB}(w_1), \dots, \text{EMB}(w_n)] \in \mathbb{R}^{nd}$$

$$q_\phi(\mathbf{x}_0 | \mathbf{w}) = \mathcal{N}(\text{EMB}(\mathbf{w}), \sigma_0 I)$$

σ_0 is a hyper parameter (a small number)

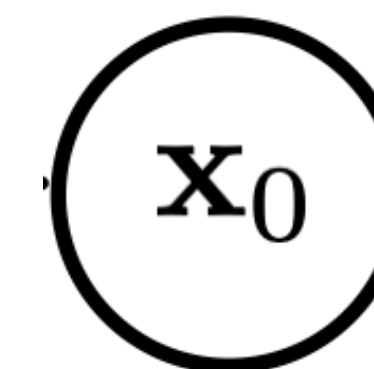
How to choose the embedding function $\text{EMB}(w_i)$

$$x_0 \in \mathbb{R}^{nd}$$

$$x_t \in \mathbb{R}^{nd}$$

$$x_T \in \mathbb{R}^{nd}$$

Continuous



Discrete

Text

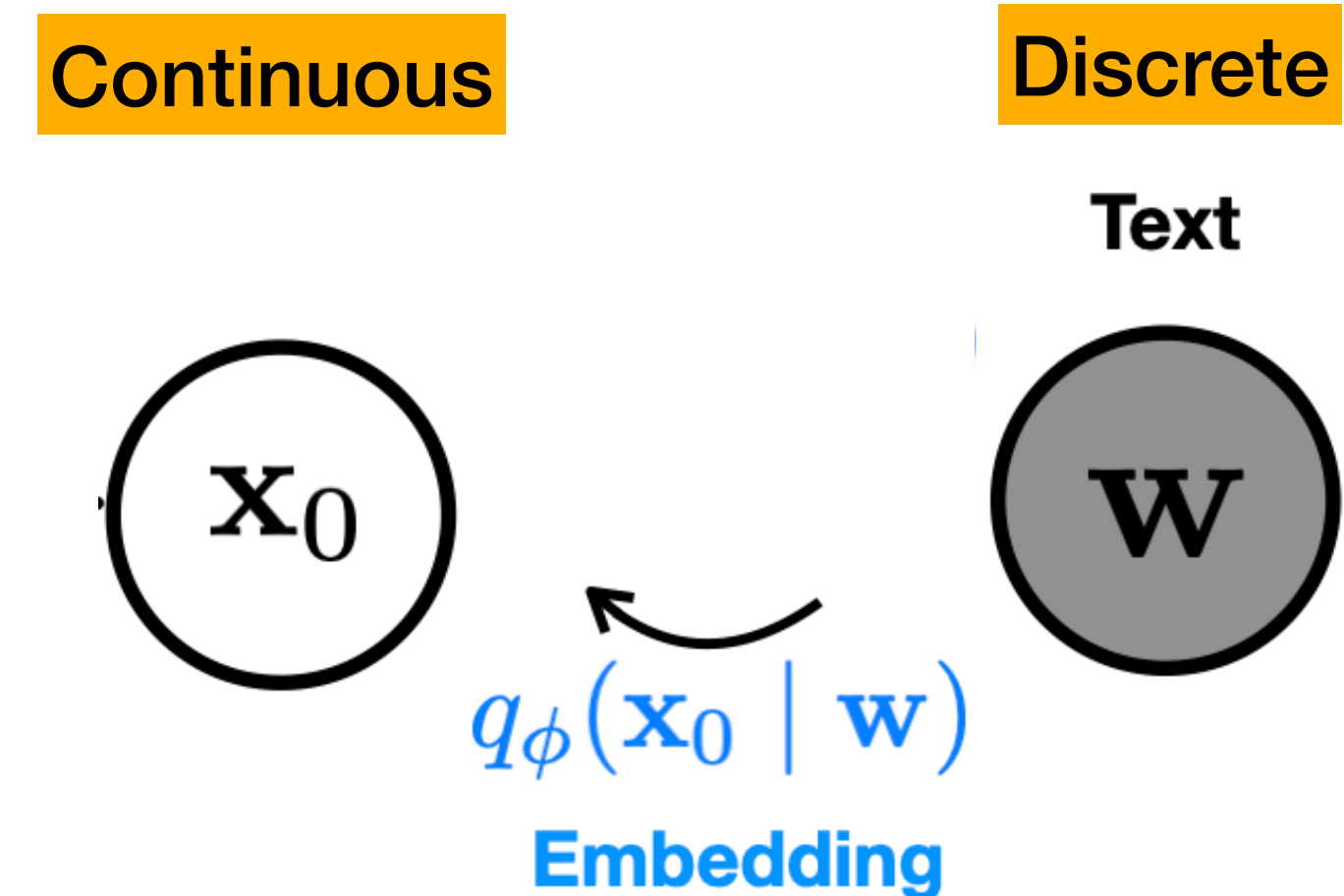


$q_\phi(\mathbf{x}_0 | \mathbf{w})$
Embedding

Embedding

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

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



$$x_0 \in \mathbb{R}^{nd}$$

$$x_t \in \mathbb{R}^{nd}$$

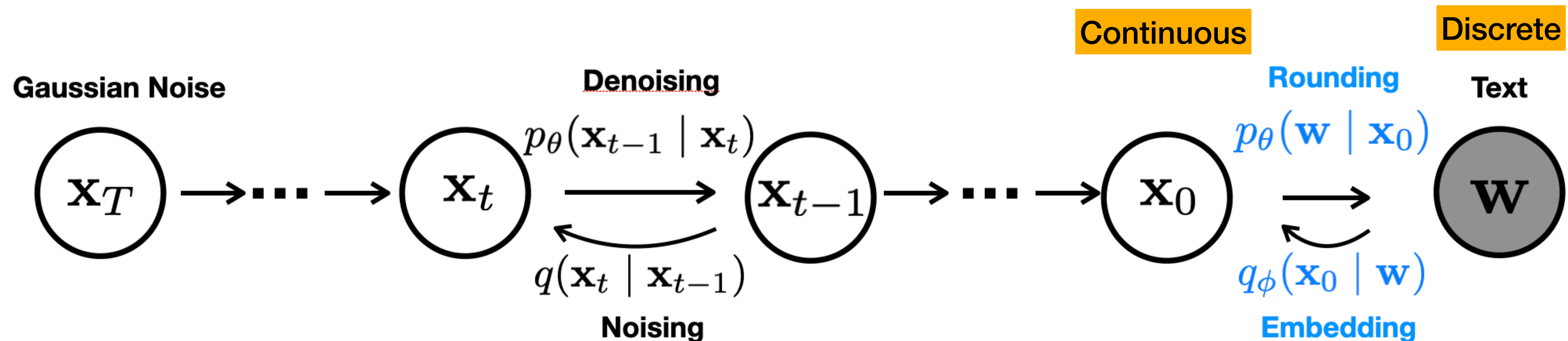
$$x_T \in \mathbb{R}^{nd}$$

Embedding: End-to-end Training

maximize probability of Embedding

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

reconstruction loss



$$x_0 \in \mathbb{R}^{nd}$$

$$x_t \in \mathbb{R}^{nd}$$

$$x_T \in \mathbb{R}^{nd}$$

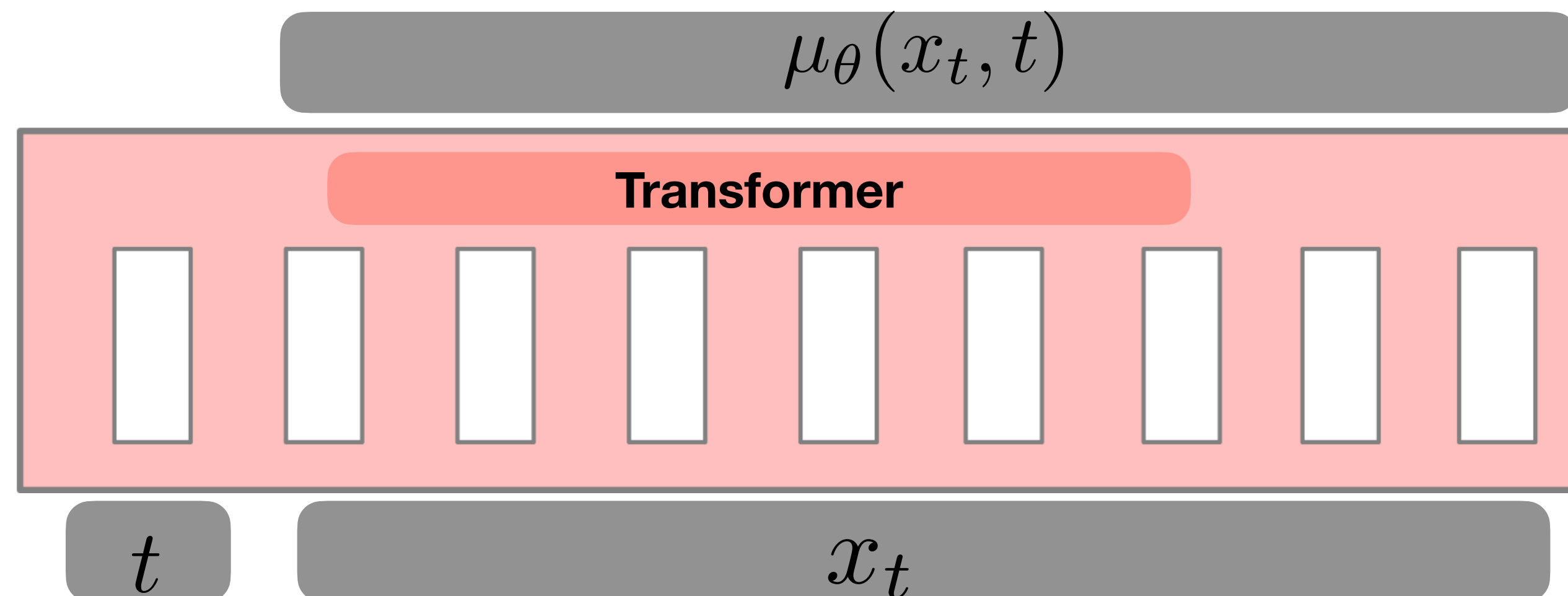
Embedding: End-to-end Training

maximize probability of Embedding

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

reconstruction loss

$$\mathcal{L}_{\text{simple}}(x_0) = \sum_{t=1}^T \mathbb{E}_{q(\mathbf{x}_t \mid \mathbf{x}_0)} \left\| \hat{\mu}_\theta(\mathbf{x}_t, t) - \mu_{t-1}(\mathbf{x}_t, \mathbf{x}_0) \right\|^2$$



$x_0 \in \mathbb{R}^{nd}$
 $x_t \in \mathbb{R}^{nd}$
 $x_T \in \mathbb{R}^{nd}$

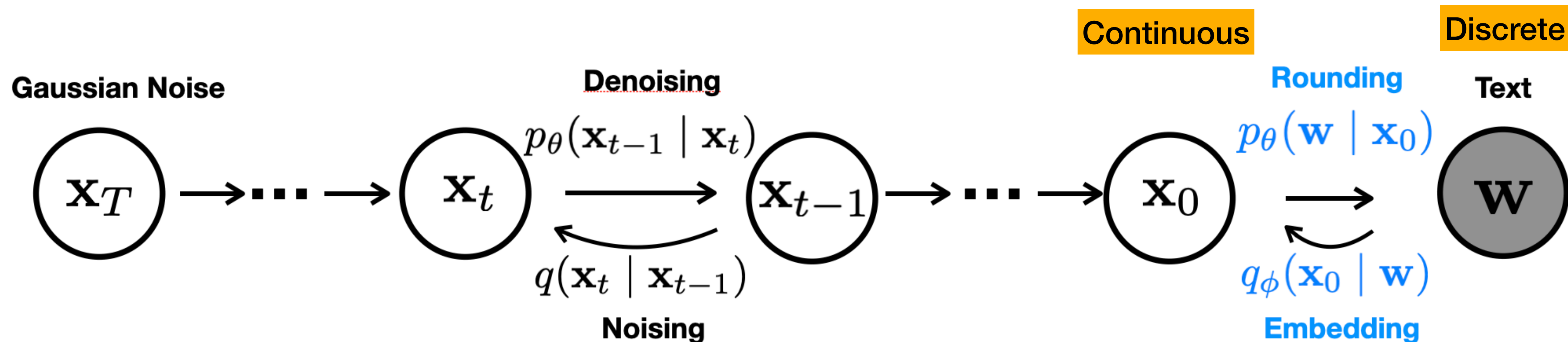
Embedding: End-to-end Training

maximize probability of Embedding

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

reconstruction loss

$$p_\theta(\mathbf{w} \mid \mathbf{x}_0) = \prod_{i=1}^n p_\theta(w_i \mid x_i)$$



$x_0 \in \mathbb{R}^{nd}$
 $x_t \in \mathbb{R}^{nd}$
 $x_T \in \mathbb{R}^{nd}$

Embedding: End-to-end Training

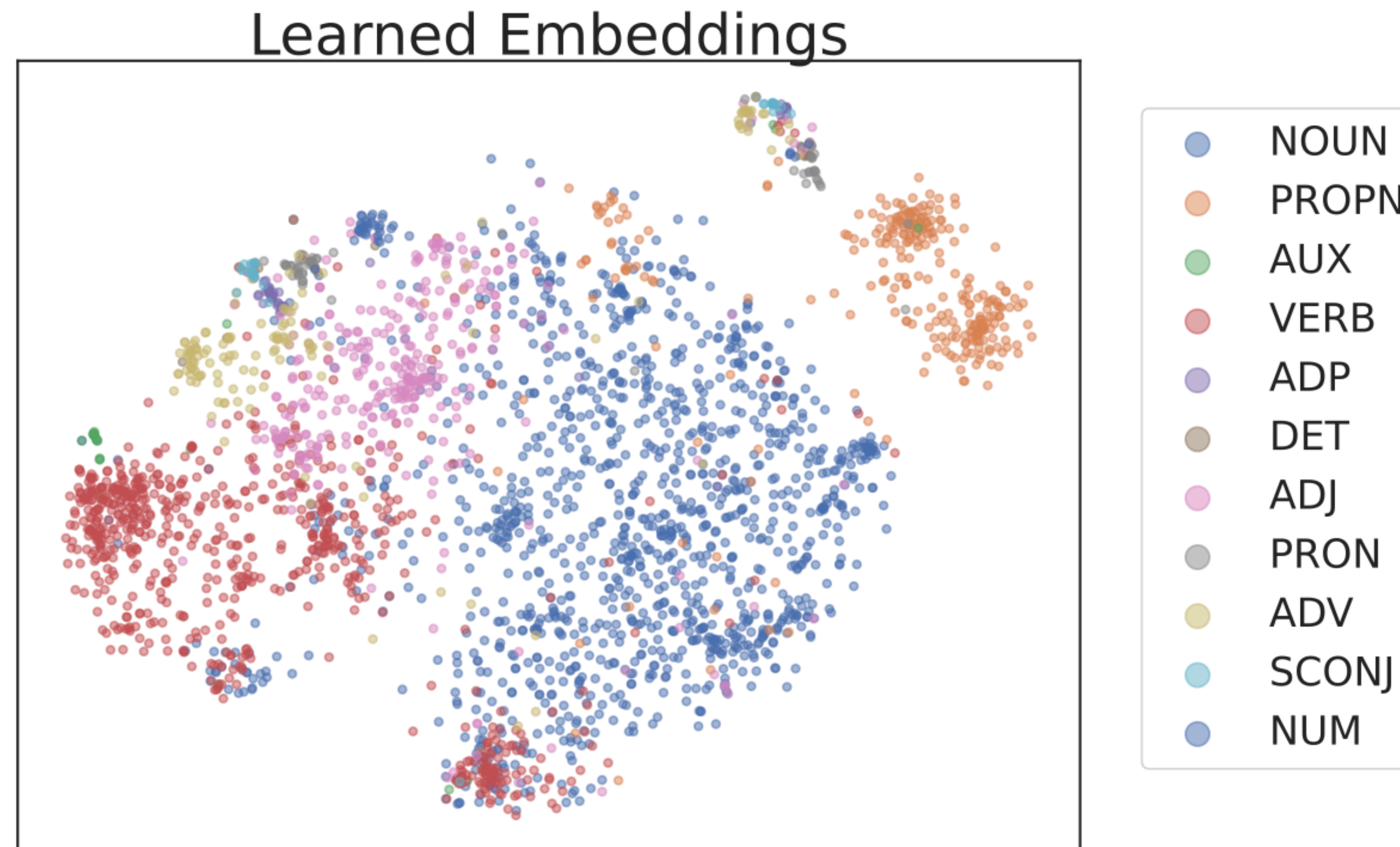


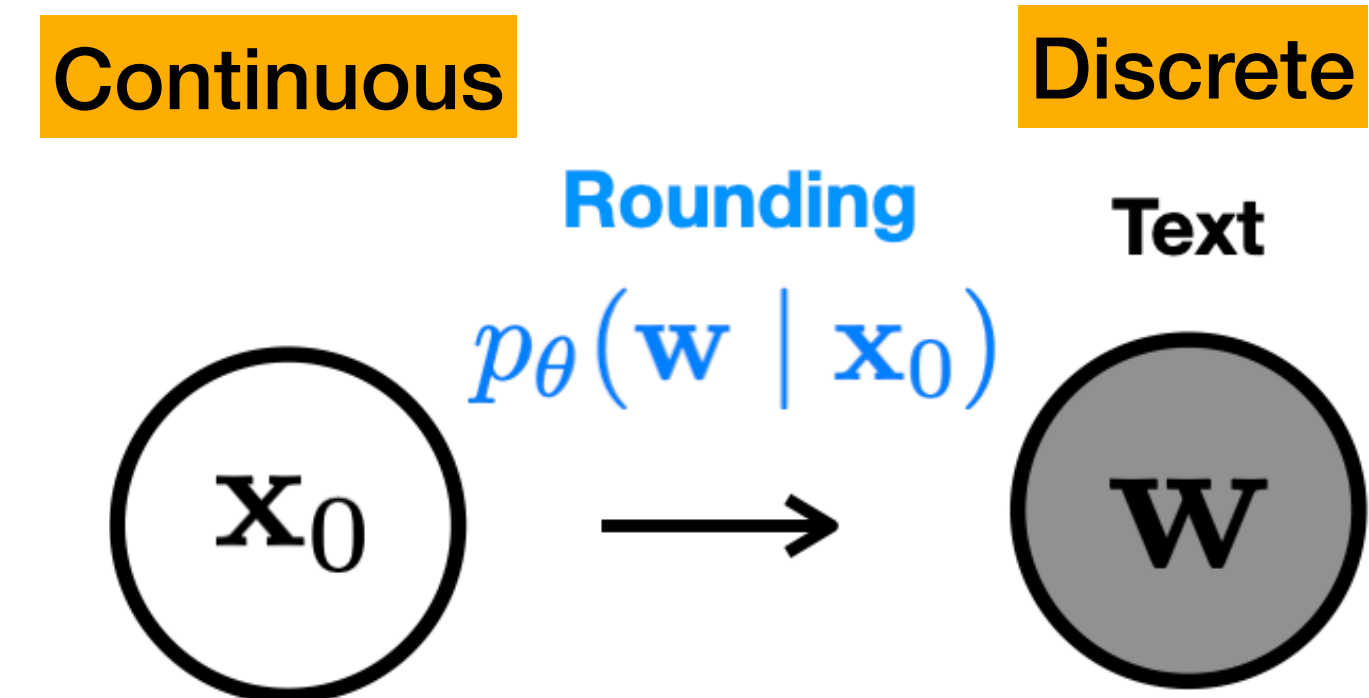
Figure 3: A t-SNE [41] plot of the learned word embeddings. Each word is colored by its POS.

Rounding

$$\hat{w}_i = \operatorname{argmax}_{w_i} p_{\theta}(w_i \mid x_0[i])$$


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

😓 Reality: there is still rounding error.



Reducing Rounding Error

$$\hat{w}_i = \operatorname{argmax}_{w_i} p_{\theta}(w_i \mid x_0[i])$$

 Ideally, the model should predict x_0 that lies exactly on a word embedding.

 Reality: there is still rounding error.

My ice cream is [BLANK].

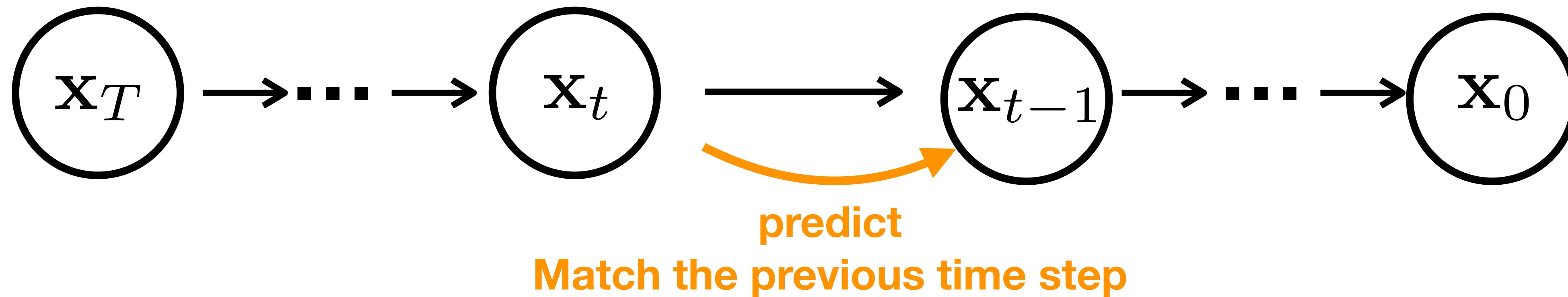
melting 

saving 

“melting” and “saving” are close in the embedding space.

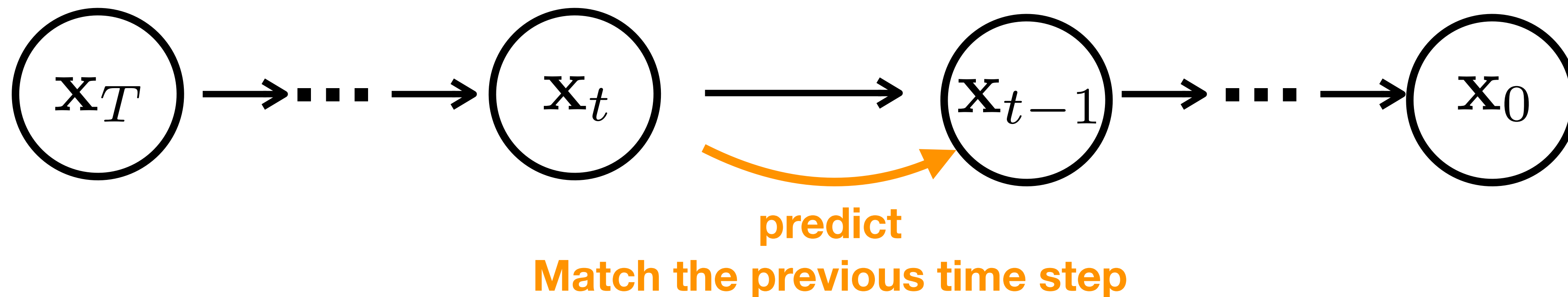
But they are not substitutable in this context.

Reducing Rounding Error (training time)



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

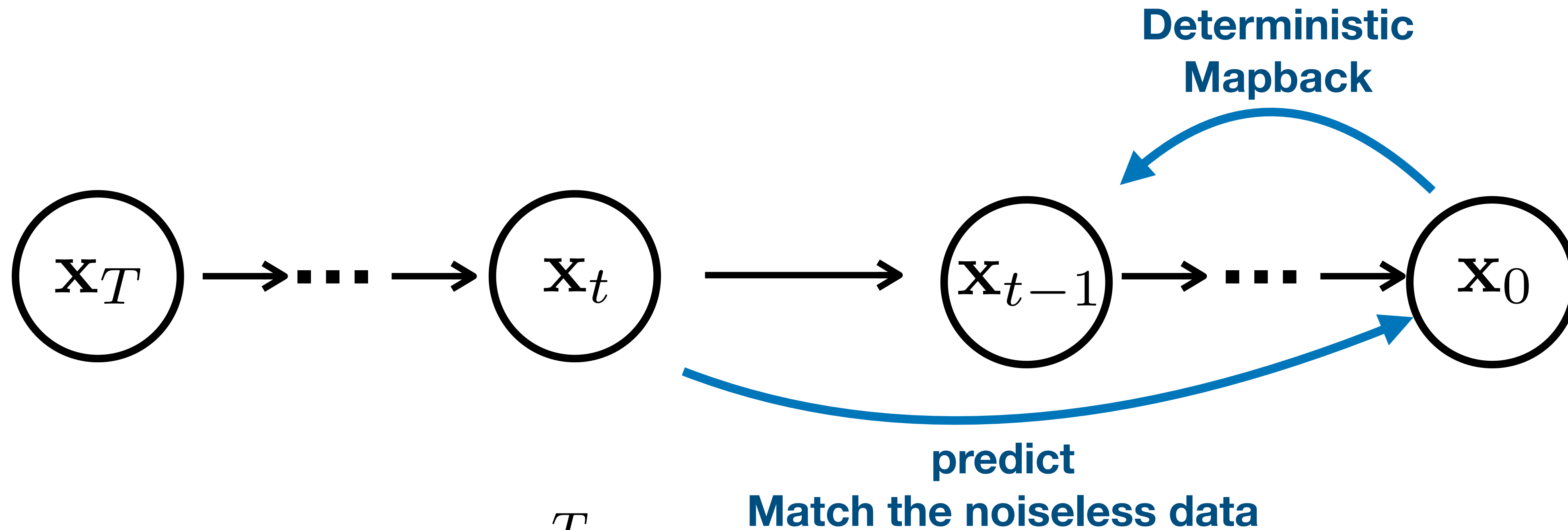
Reducing Rounding Error (training time)



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

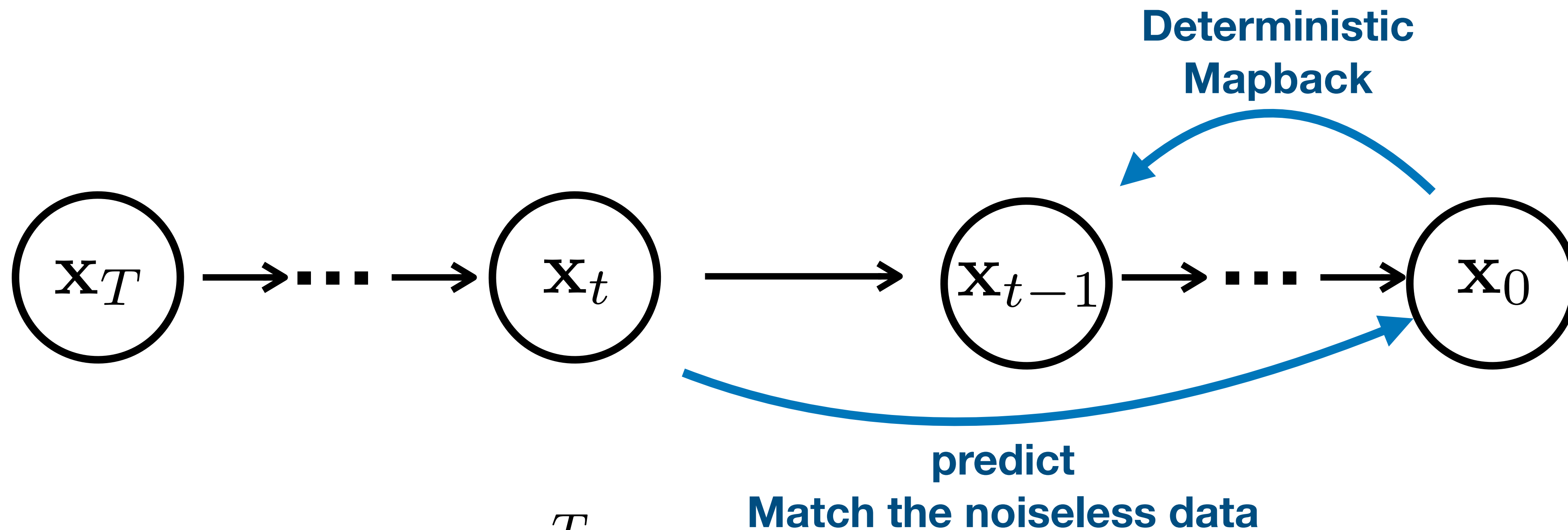
Intuition: Rounding happens all at the last step: $x_1 \rightarrow x_0$ which is hard and prone to error.

Reducing Rounding Error (training time)



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

Reducing Rounding Error (training time)



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

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

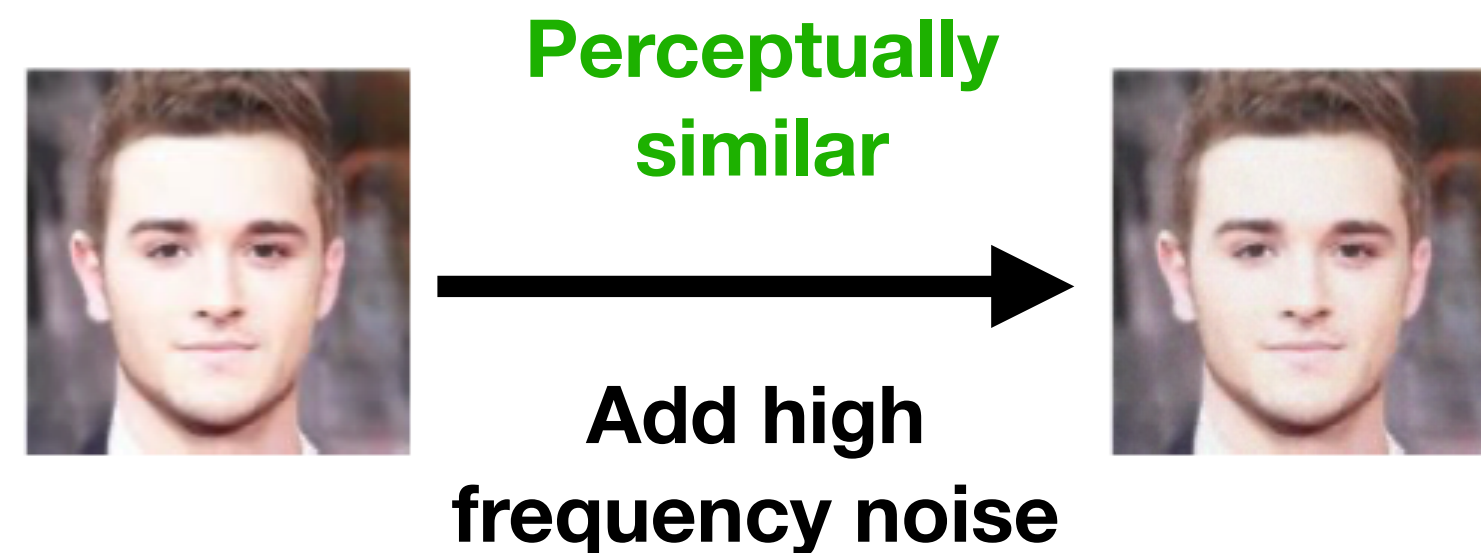
Discussion: Text v.s. Images

High frequency v.s. Low frequency

Low frequency → the large scale structure of the images

Medium frequency → the details.

High frequency → perceptually meaningless.



High noise levels: build up the low-frequency components.
Low noise levels: refine the high frequency details.

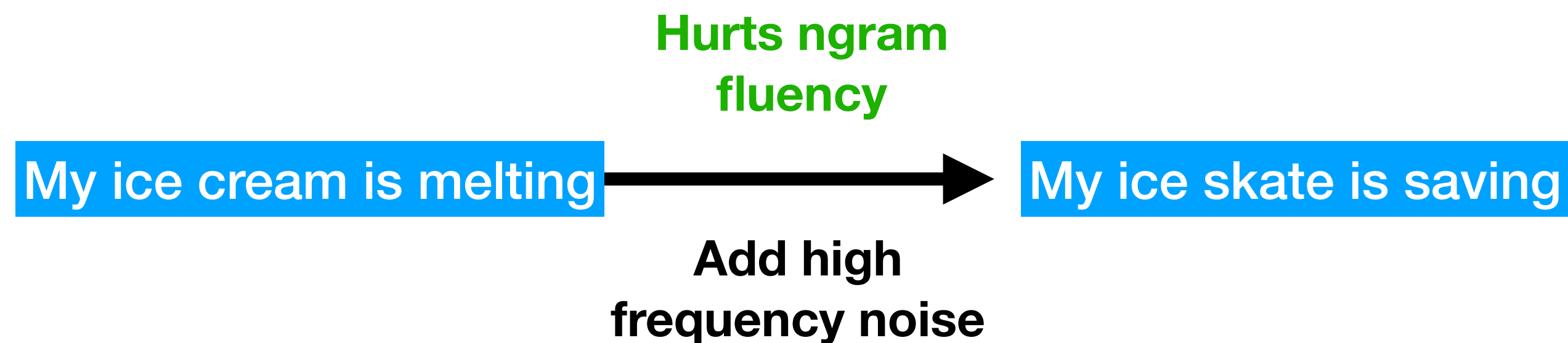
For text, what does high frequency even mean?

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 high frequency components matters a lot for text, because it relates to token prediction, and it's easy to notice bad ngrams.

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

Very different

How **autoregressive** LM produce long text:

Left-to-right.

More similar

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 > diffusion-LM

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 > diffusion-LM

Decoding Efficiency

Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM > diffusion-LM

Decoding Efficiency

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

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

Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM > diffusion-LM

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.

Discussion: Diffusion v.s. Autoregressive

Training Efficiency

Autoregressive LM > diffusion-LM

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

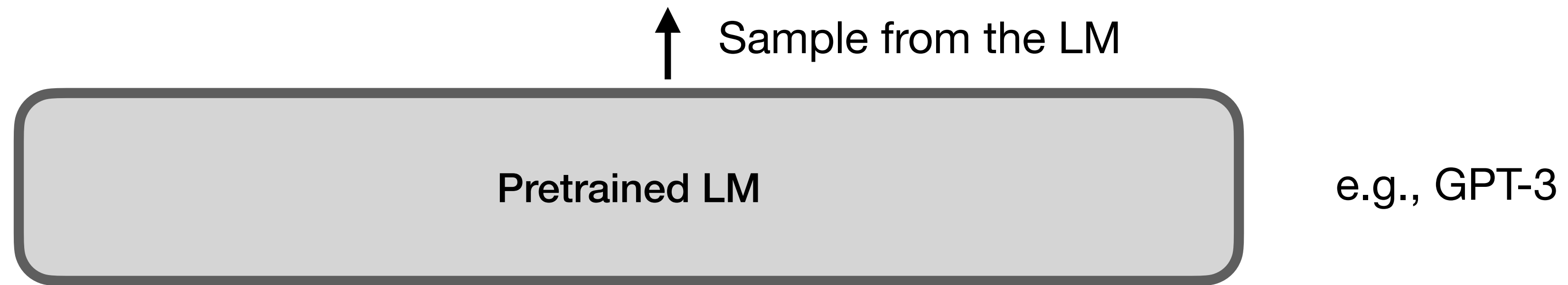
1. Flexible Generation Order

Diffusion-LM > Autoregressive LM

2. Controllable Text Generation

Diffusion-LM > Autoregressive LM (Spoiler)

Text Generation



Text Generation

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...

↑ Sample from the LM

Pretrained LM

e.g., GPT-3

Text Generation

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...

↑ Sample from the LM

Pretrained LM

e.g., GPT-3

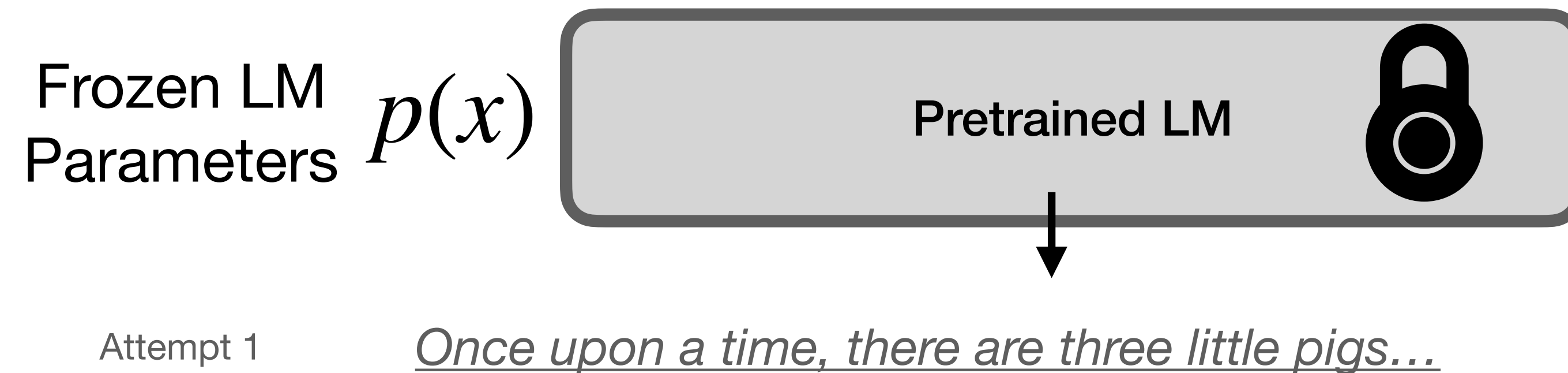
How to generate positive reviews about a coffee shop called Coupa?

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



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

$$p(c | x)$$

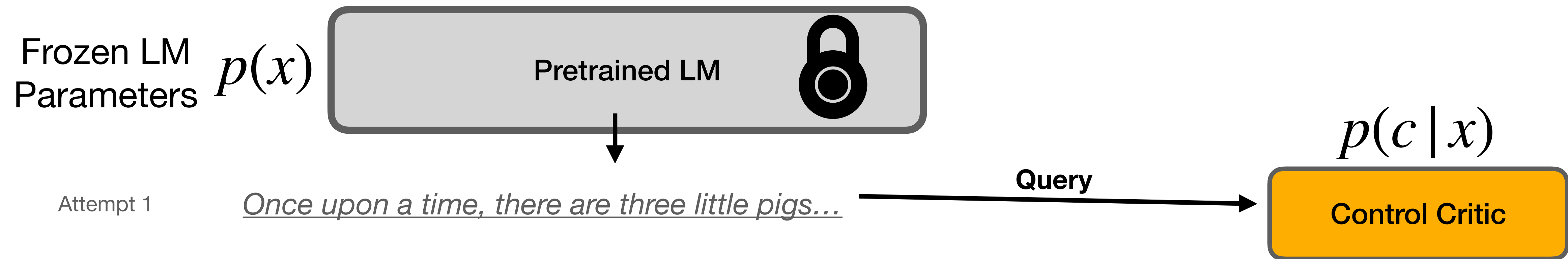
Control Critic

How to Control Text Generation?

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

Plug-and-play Control

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

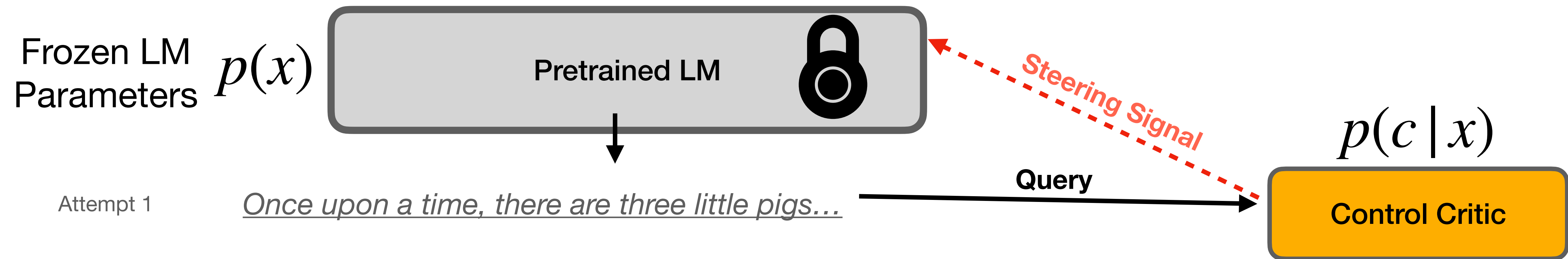


How to Control Text Generation?

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

Plug-and-play Control

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



How to Control Text Generation?

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

Plug-and-play Control

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

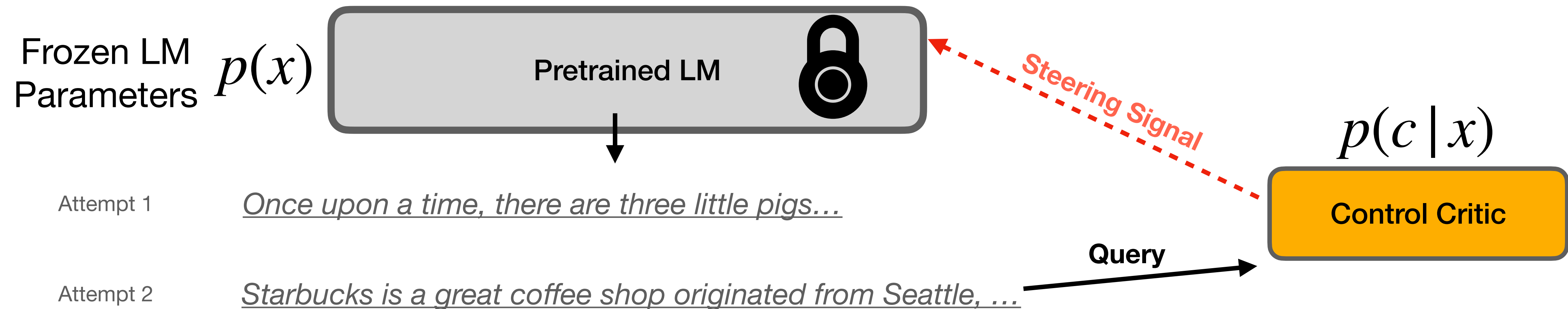


How to Control Text Generation?

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

Plug-and-play Control

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

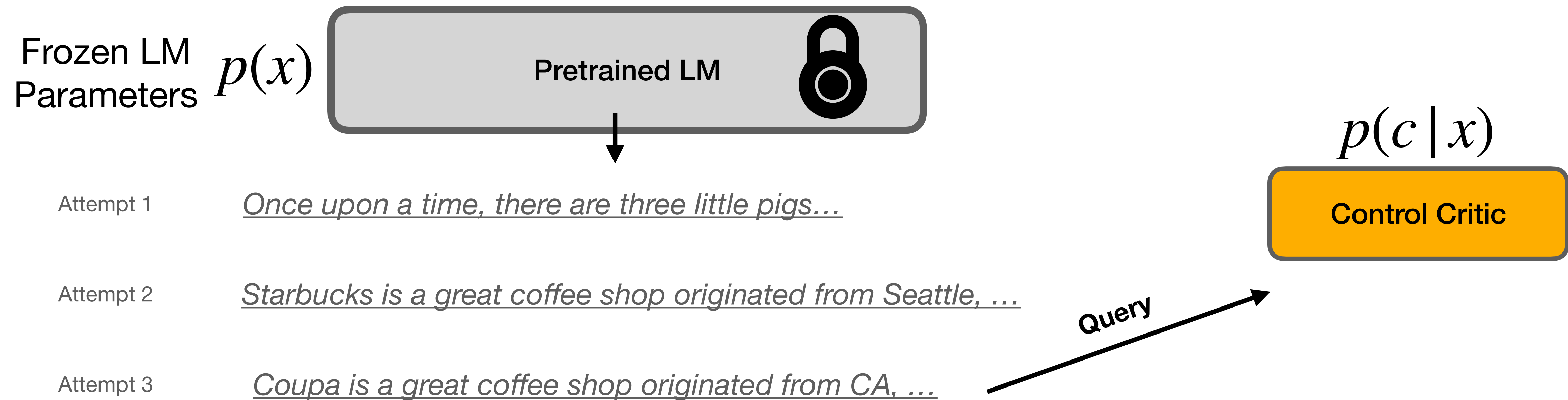


How to Control Text Generation?

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

Plug-and-play Control

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

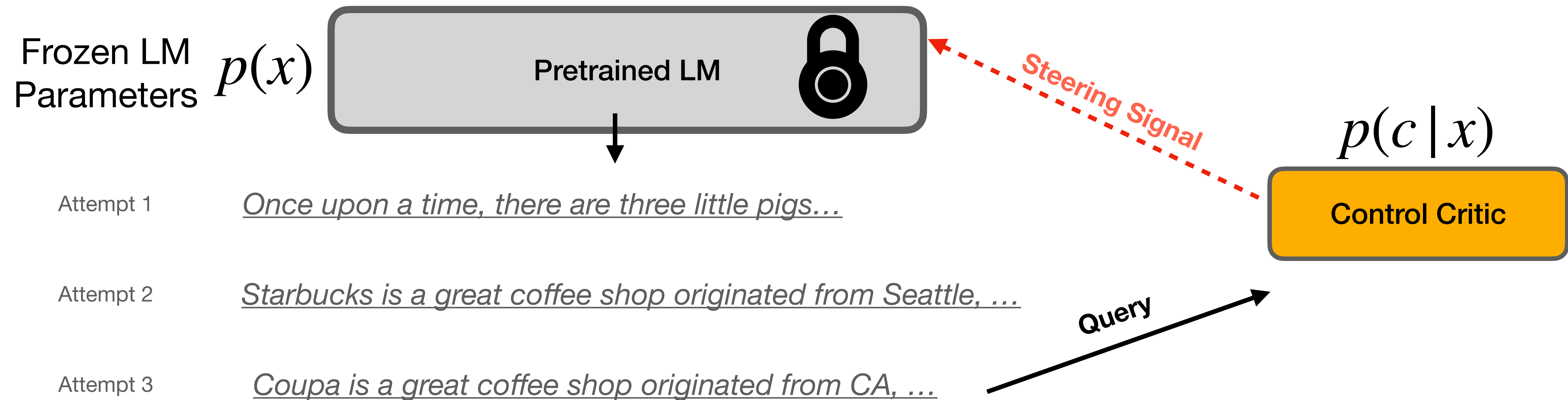


How to Control Text Generation?

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

Plug-and-play Control

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



How to Control Text Generation?

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

Plug-and-play Control

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



✓ 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

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

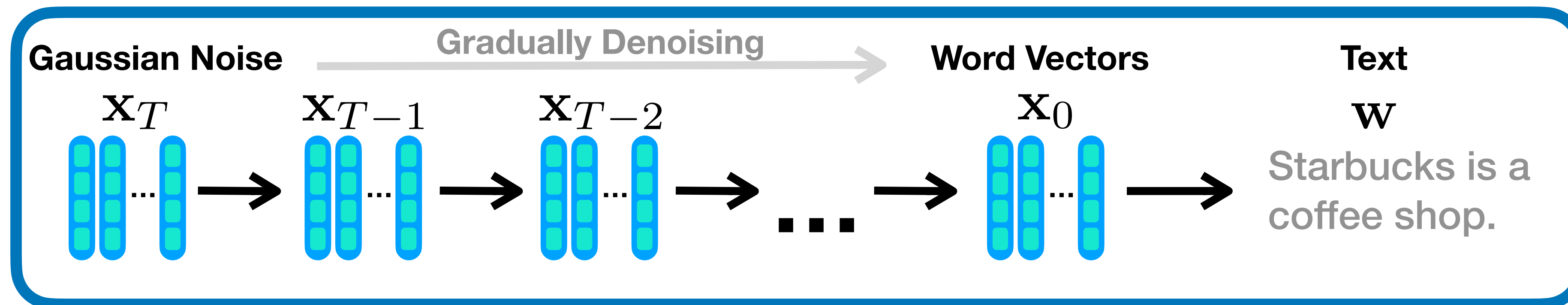


- ✓ lightweight. Frozen LM; only need to specify the classifier.
- ✓ Enables composition.

Controllable Generation

Goal: sample from $p(\mathbf{x}_{0:T}|\mathbf{c})$

Diffusion-LM

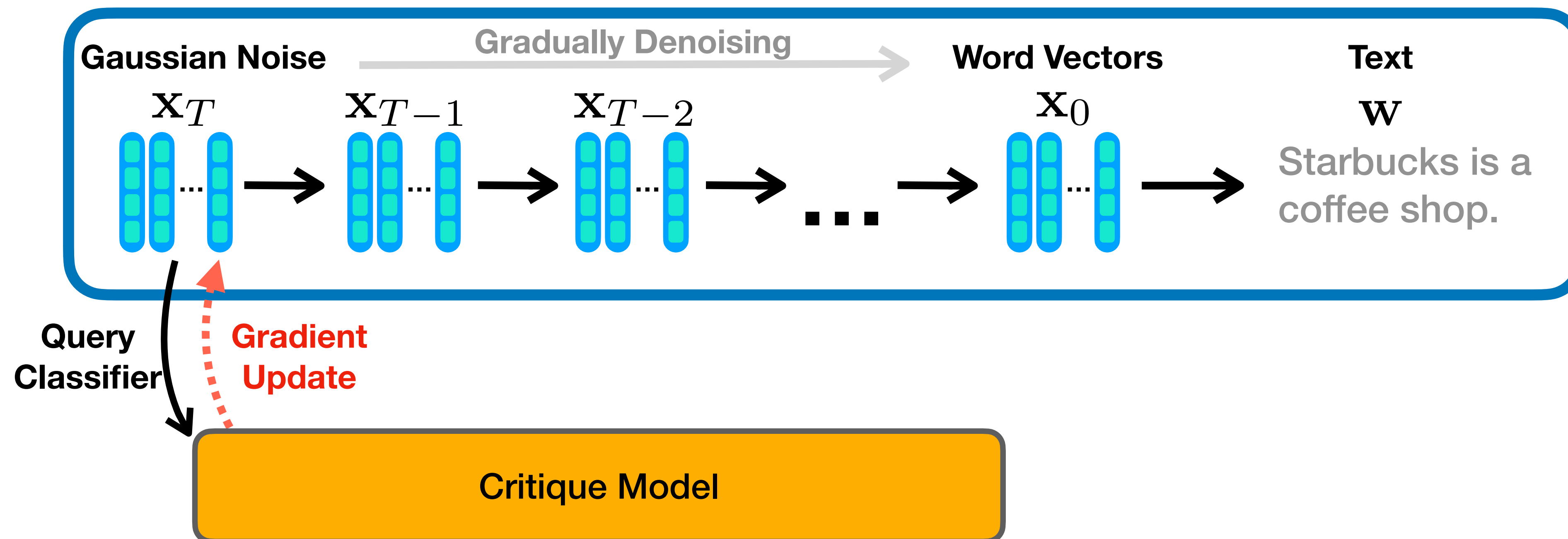


Critique Model

Controllable Generation

Goal: sample from $p(\mathbf{x}_{0:T}|\mathbf{c})$

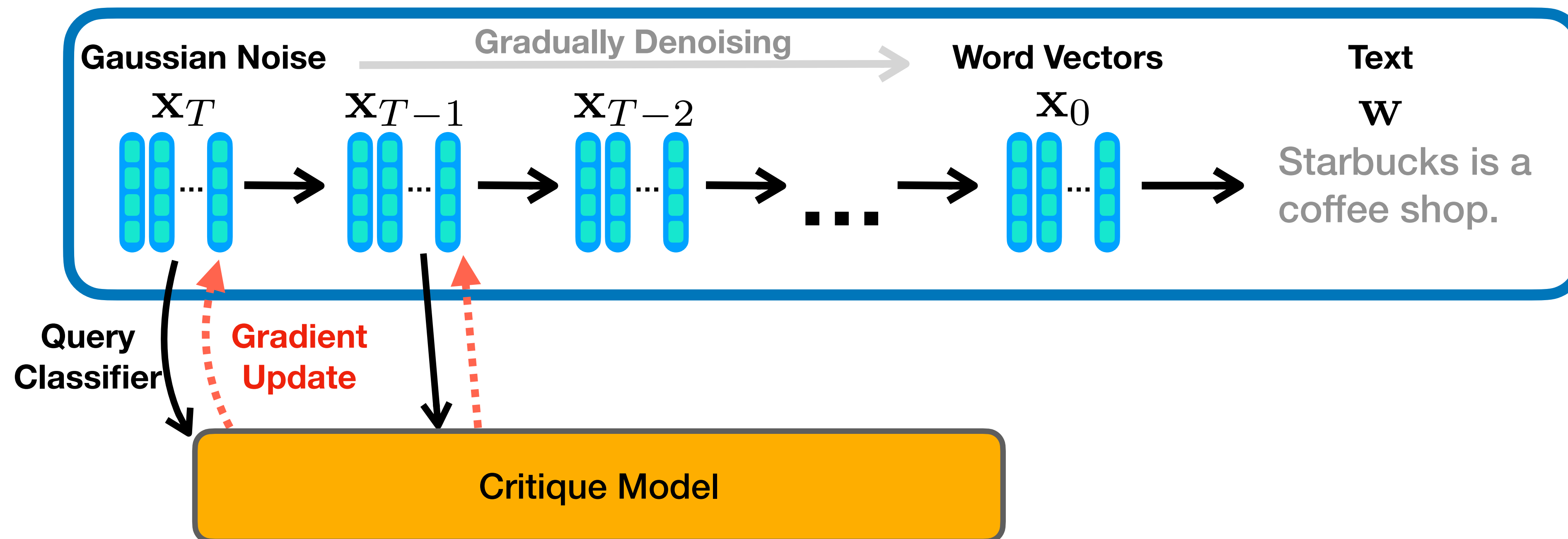
Diffusion-LM



Controllable Generation

Goal: sample from $p(\mathbf{x}_{0:T}|\mathbf{c})$

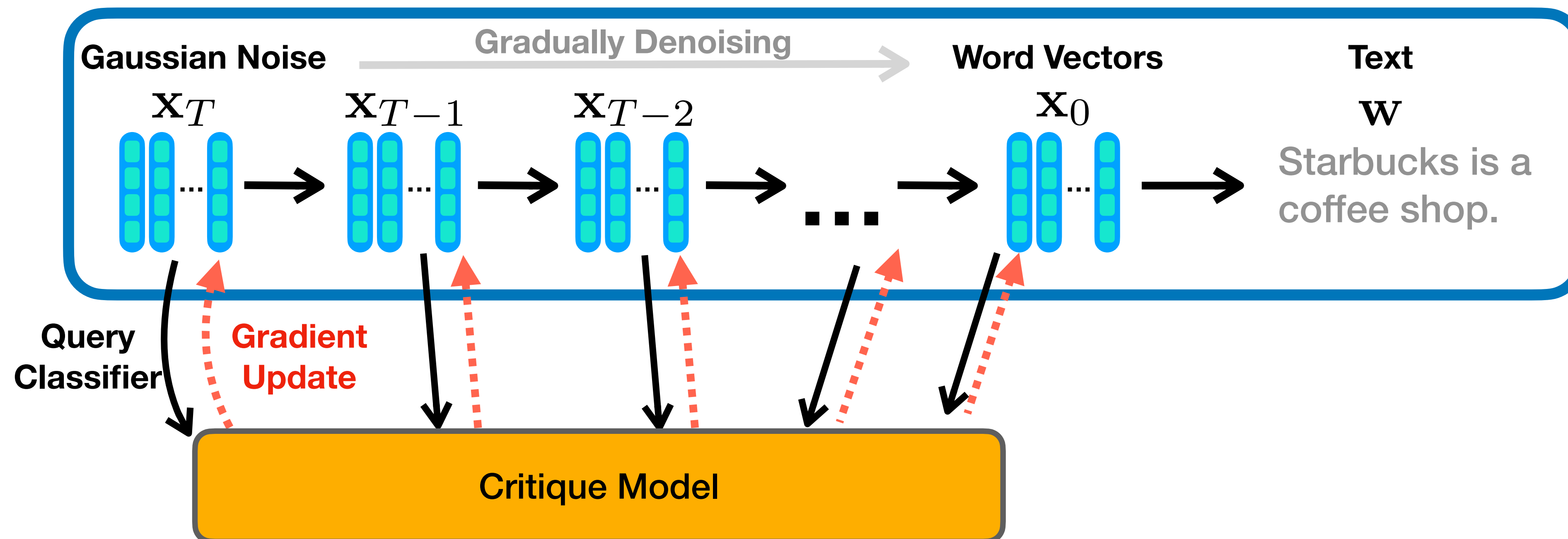
Diffusion-LM



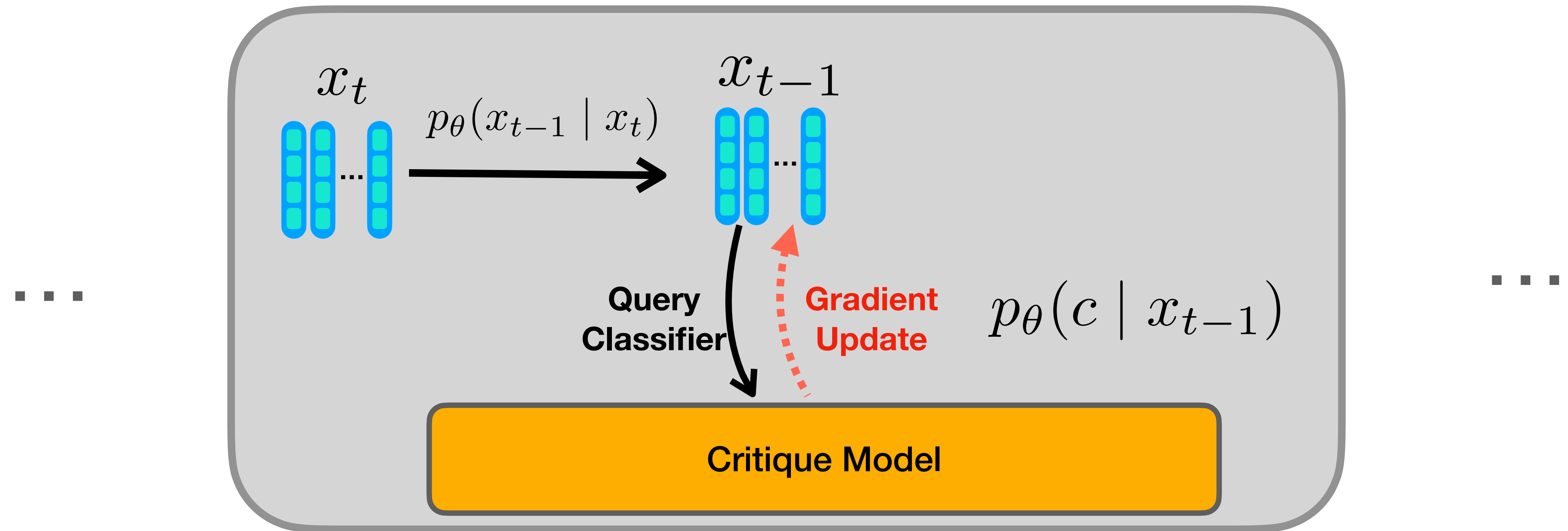
Controllable Generation

Goal: sample from $p(\mathbf{x}_{0:T}|\mathbf{c})$

Diffusion-LM



Controllable Generation



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

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

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:

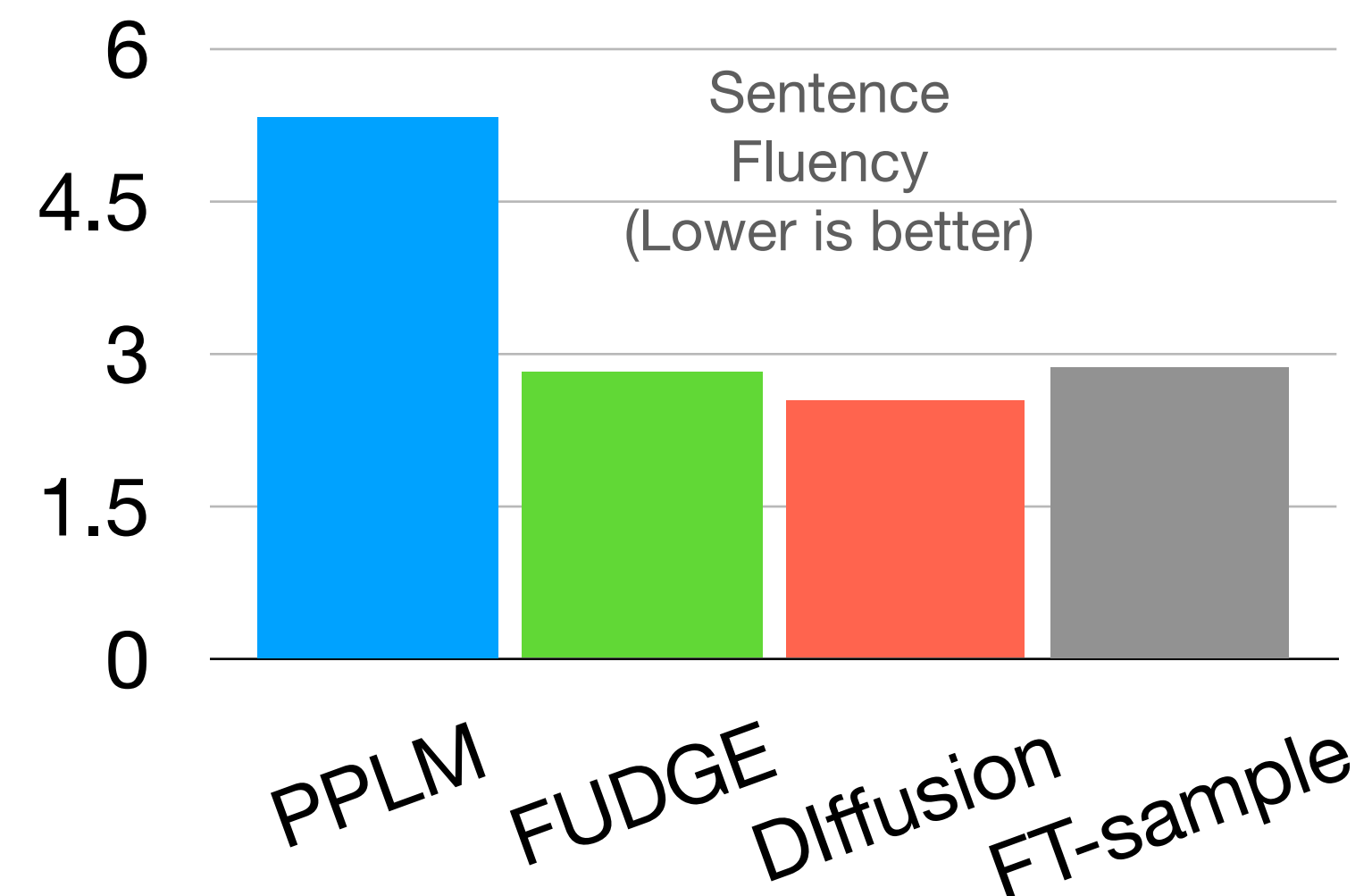
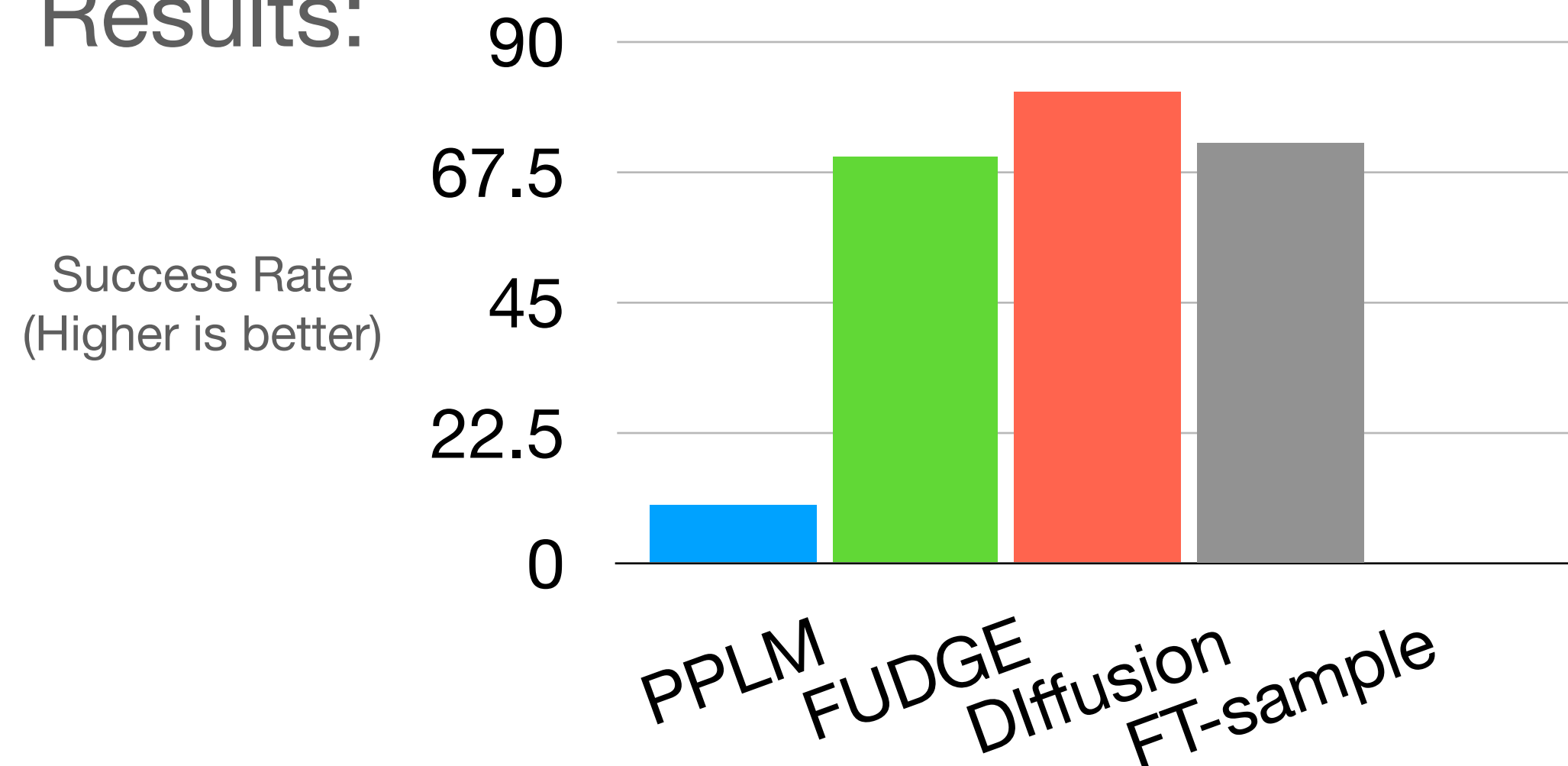
Semantic Content

Food = Japanese

Output Text

Browns Cambridge is good for Japanese food and also children friendly near The Sorrento .

Results:



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:

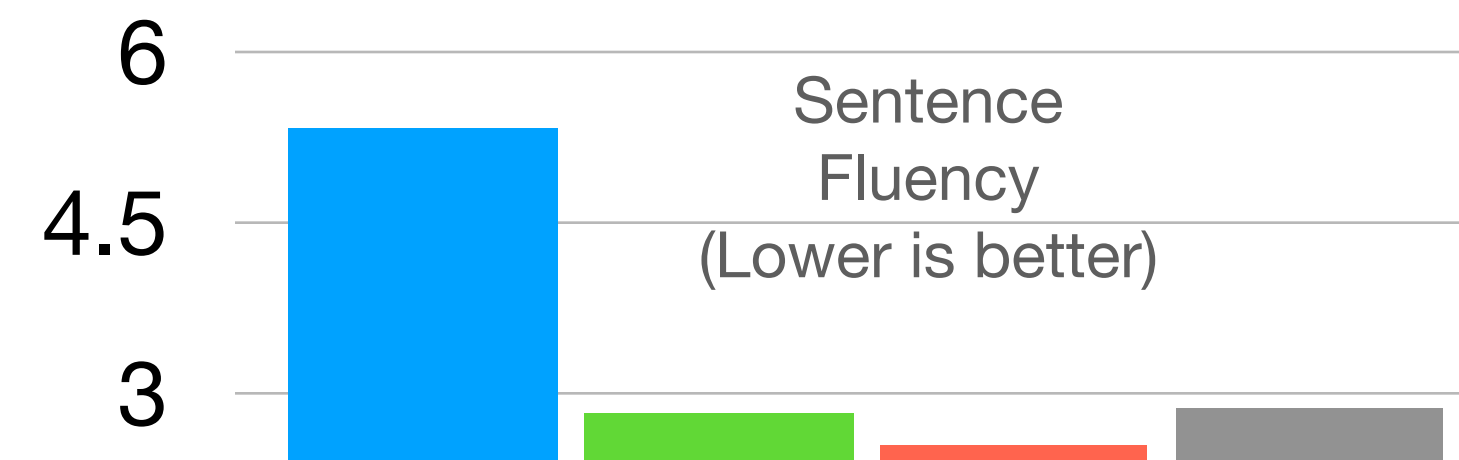
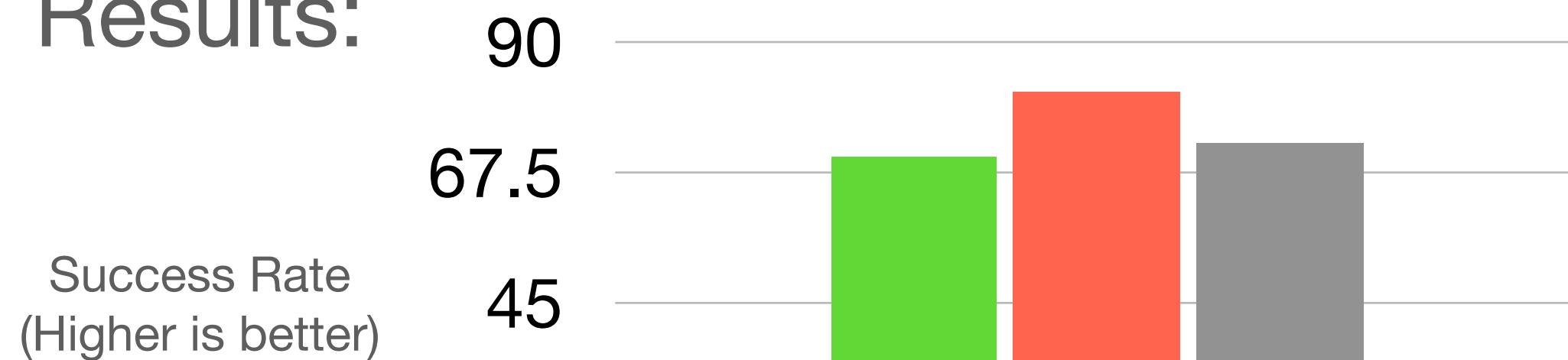
Semantic Content

Food = Japanese

Output Text

Browns Cambridge is good for Japanese food and also children friendly near The Sorrento .

Results:



Takeaway:

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

Thanks :)

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