Diffusion-LM

Improves **Controllable** Text Generation

NEURAL INFORMATION PROCESSING SYSTEMS 2022

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani,
Percy Liang, Tatsunori Hashimoto

Stanford University

Pretrained AR-LM

Pre-trained large Autoregressive Language Models (AR-LMs) can generate high-quality text, but in order to be applicable to everyday life, it is necessary to control the generated text.

It is common to use pairs of (control, text) to fine-tune the model

Issue

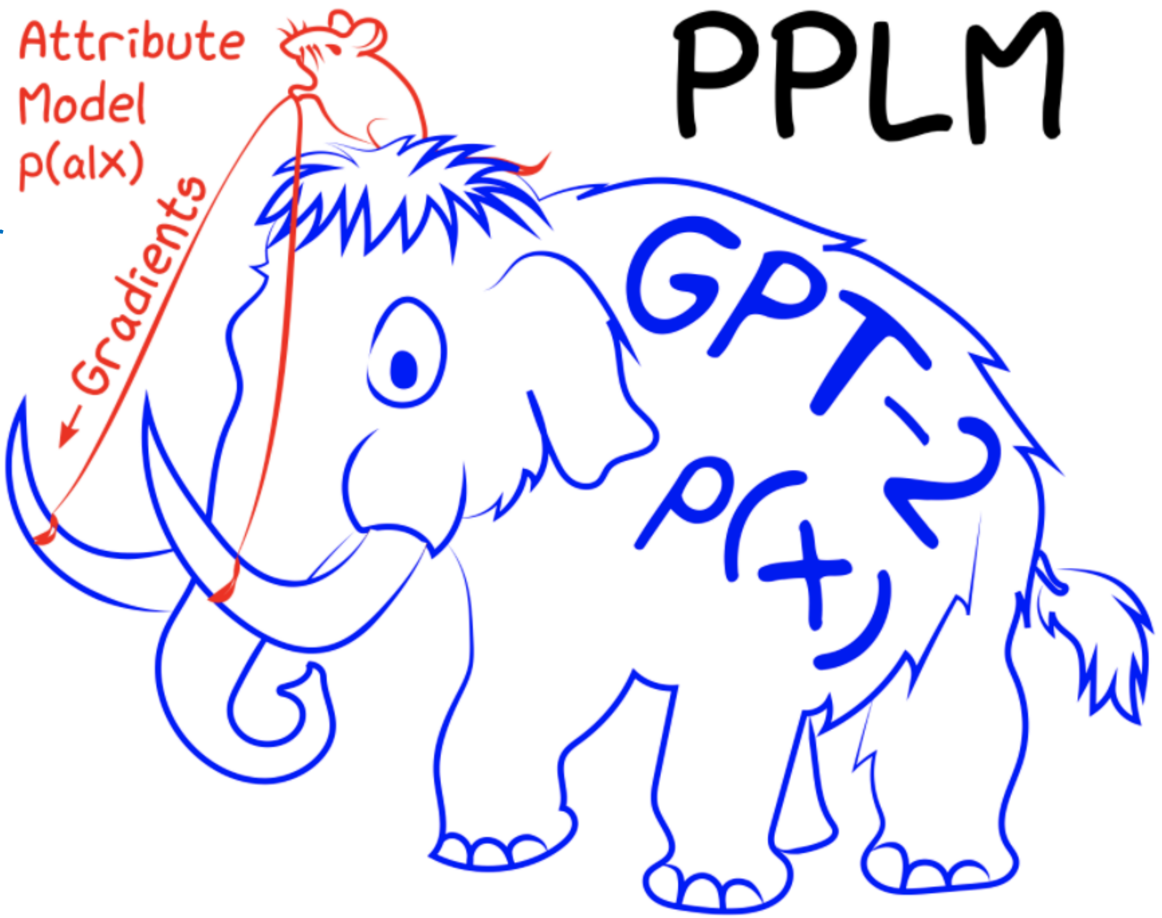
- Updating LM parameters for different tasks is expensive.
- Difficult to combine multiple conditions.
- The fixed generation order limits the models' flexibility in many controllable generation settings

Controllable

Plug & Play

$$p(x_{i+1}|x_{0:i}, a)$$
$$\propto p(a|x_{0:i+1})p(x_{i+1}|x_{0:i})$$

- A pretrained large LM is responsible for generating fluent text.
- Use additional small models to control the attributes of the generated text.
- No need to fine-tune the parameters of the LM.
- Only controls at the attribute level are available.



Controllable DPMs for Text

Diffusion probabilistic models (DPMs) have demonstrated great success in **continuous data domains**, producing images and audio that have state-of-the-art sample quality.

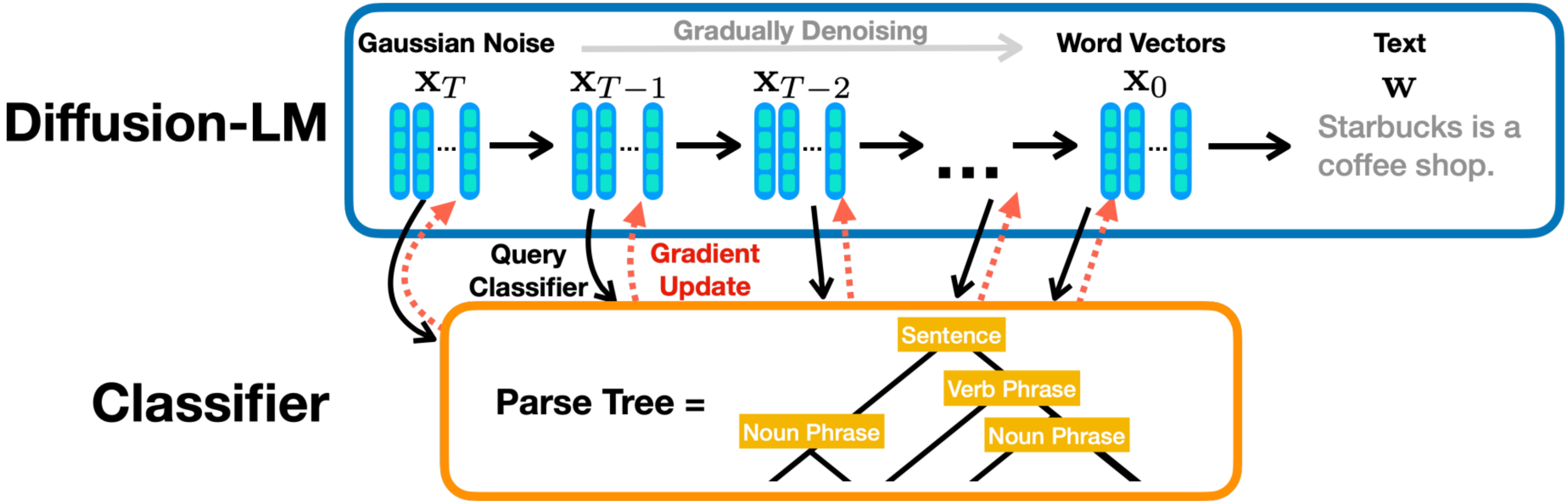
And can use gradient-based control methods for effective control. However...

Past Works in Text

- **Discrete** DPMs.
- Unable to use gradient-based control methods.

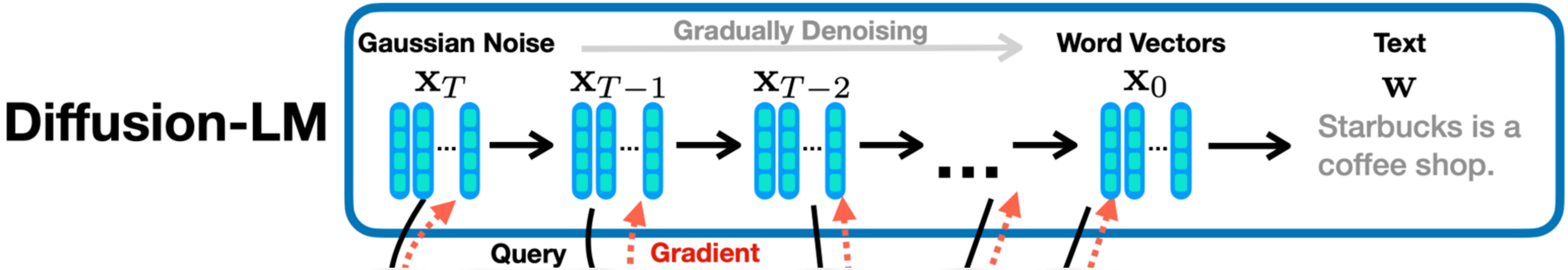
Diffusion-LM

Continuous DPMs for Text



Diffusion-LM

Continuous DPMs for Text

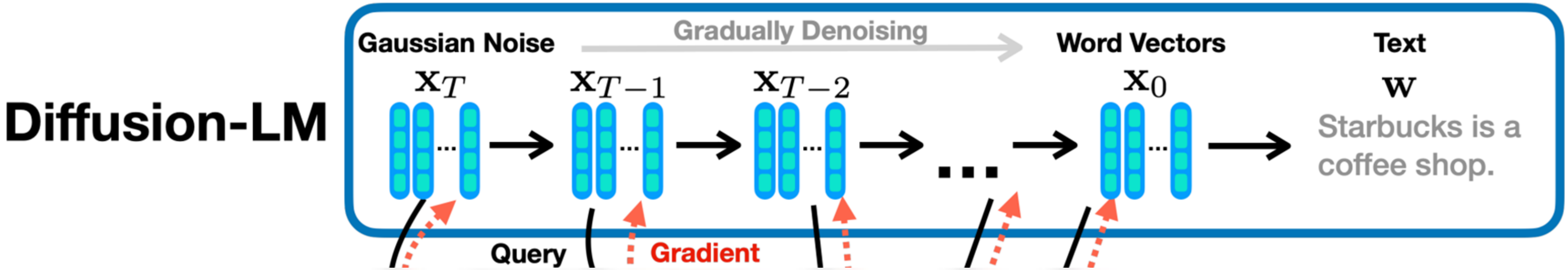


- Starts with a sequence of Gaussian noise vectors
- Incrementally denoises them into vectors corresponding to words.

$$\mathrm{x}_\mathrm{T} \sim \mathcal{N}(0, I) \in \mathbb{R}^{L\times d}$$

Diffusion-LM

Continuous DPMs for Text

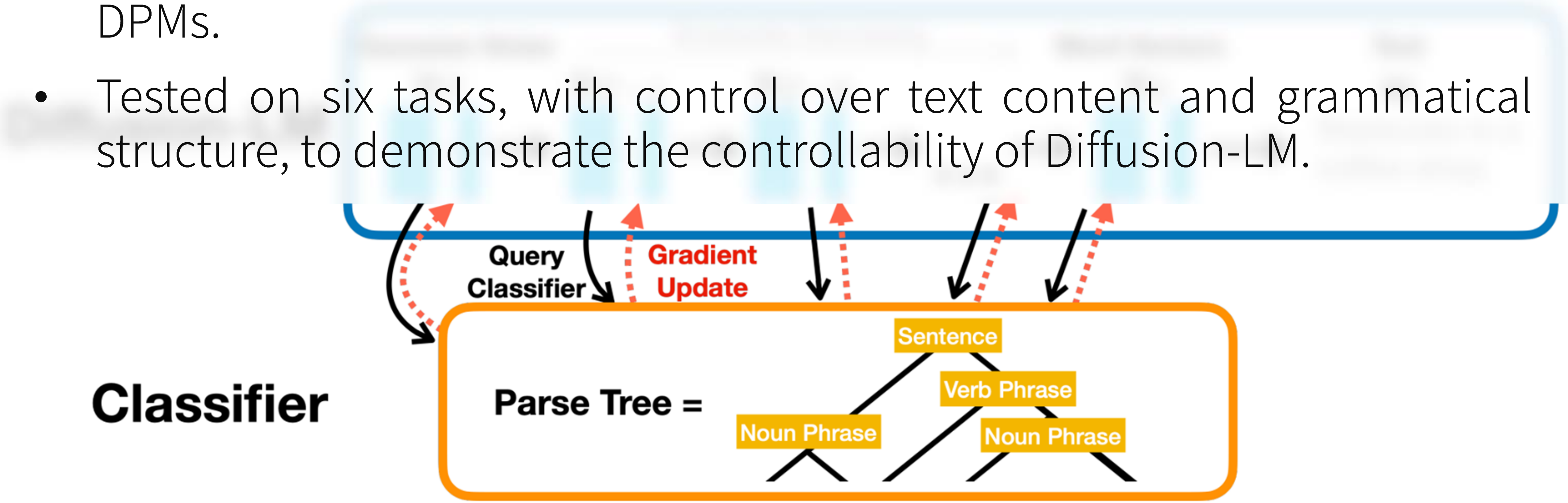


- Starts with a sequence of Gaussian noise vectors
- Incrementally denoises them into vectors corresponding to words.
- There is no fixed generation order so it can be used directly for infilling tasks.

Diffusion-LM

Gradient-based Control

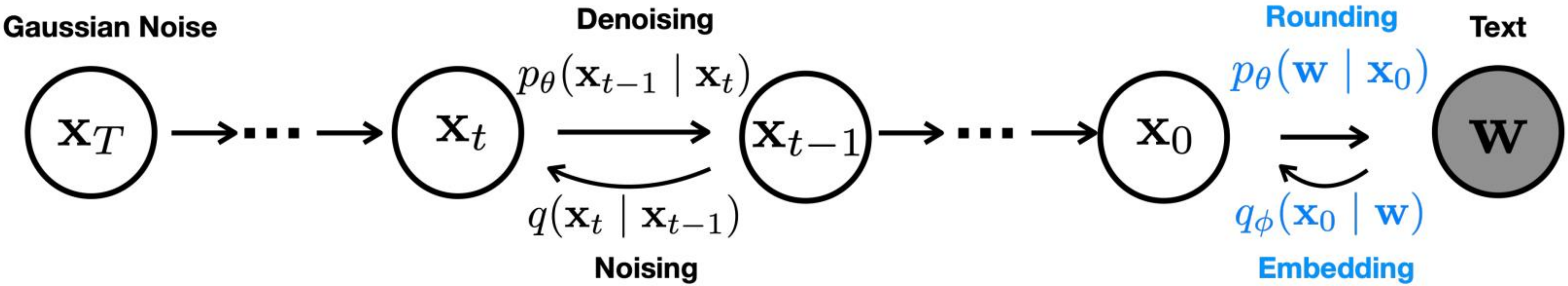
- Change sampling paths using Gradient-based Controls common to DPMs.
- Tested on six tasks, with control over text content and grammatical structure, to demonstrate the controllability of Diffusion-LM.



Diffusion-LM

Loss (MSE ver.)

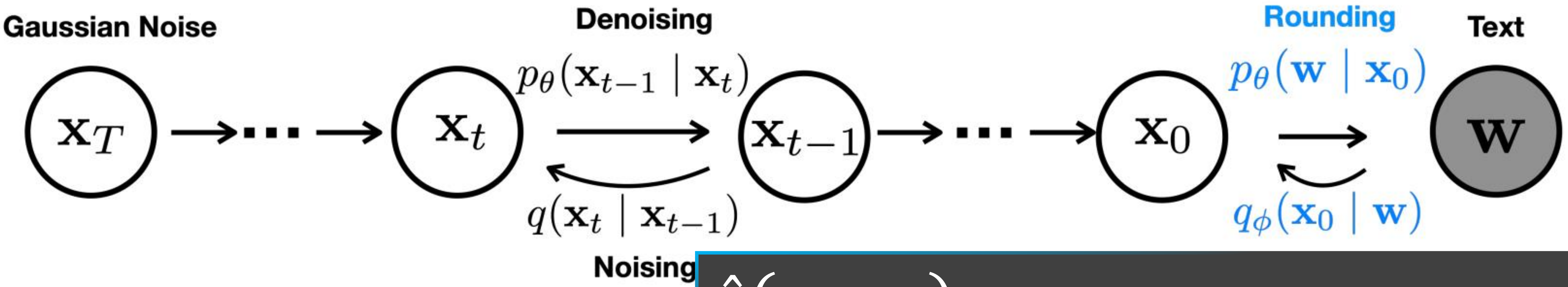
$$\mathcal{L}_{\text{simple}}^{\text{e2e}}(\mathrm{w}) = [\underbrace{\|\mu_\theta(\mathrm{x}_t, t) - \hat{\mu}(\mathrm{x}_t, \mathrm{x}_0)\|^2} - \underbrace{\log p_\theta(\mathrm{w}|\mathrm{x}_0)}]$$



Diffusion-LM

Loss (MSE ver.)

$$\mathcal{L}^{\mathrm{e2e}}_{\mathrm{simple}}(\mathrm{w}) = [\underbrace{||\mu_\theta(\mathrm{x}_t, t) - \hat{\mu}(\mathrm{x}_t, \mathrm{x}_0)||^2} - \underbrace{\log p_\theta(\mathrm{w}|\mathrm{x}_0)}]$$



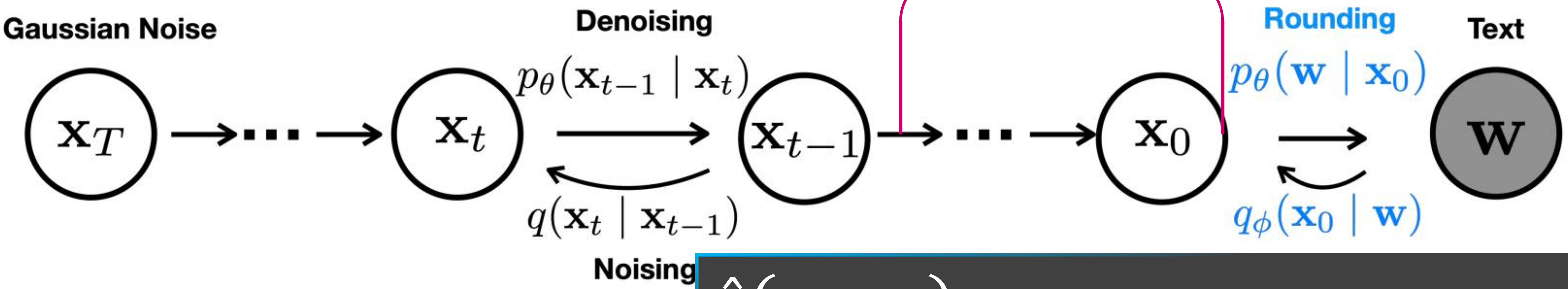
$$\hat{\mu}(\mathrm{x}_t, \mathrm{x}_0) = \mathrm{x}_{t-1}$$

Diffusion-LM

Rounding Errors

Empirically, the model fails to generate $\mathbf{x}_0$ that commits to a single word.

Only starting the denoising at t close to 0 can make $\mathbf{x}_0$ converge to the word embedding.



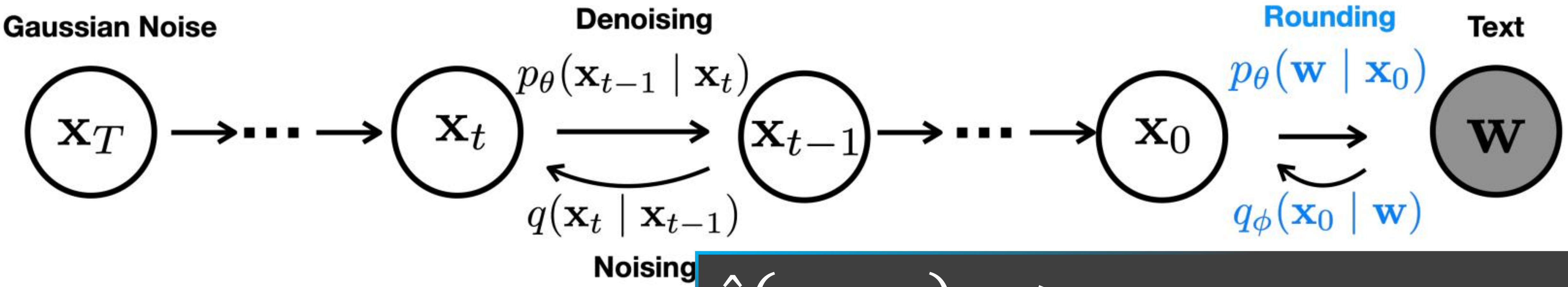
$$\hat{\mu}(\mathrm{x}_t, \mathrm{x}_0) = \mathrm{x}_{t-1}$$

Diffusion-LM

Reducing Rounding Errors

This objective quickly learn that x_0 should precisely centered at a word embedding.

$$\mathcal{L}^{\text{e2e}}_{\text{simple}}(\mathrm{w}) = \underbrace{[\|\mu_\theta(\mathrm{x}_t, t) - \mathrm{x}_0\|^2}_{} - \underbrace{\log p_\theta(\mathrm{w}|\mathrm{x}_0)]}_{}$$



$$\hat{\mu}(\mathrm{x}_t, \mathrm{x}_0) = \cancel{\mathrm{x}_{t-1}}\mathrm{x}_0$$

Controllable Generation

Reducing Rounding Errors

Sampling Algorithm

θ: parameter of Diffusion-LM, φ: parameter of Attribute Model

$\mathrm{x}_T \sim \mathcal{N}(0, I) \in \mathbb{R}^{L\times d}$

for t in {T,…,1}:

$\quad \mu_t \leftarrow \sqrt{\bar{a}_t} \cdot \mathrm{Clamp}(\mu_\theta(\mathrm{x}_t, t))$

$\quad \mathrm{x}_{t-1} \sim \mathcal{N}(\mu_t, 1 - \bar{a}_t) \in \mathbb{R}^{L\times d}$

$\quad$**for** k in {1, …, K}:

$\qquad \Delta \mathrm{x}_{t-1} \leftarrow \nabla_{\mathrm{x}_{t-1}}[\lambda \|\mu_t - \mathrm{x}_{t-1}\|^2 - \log p_\varphi(c|\mathrm{x}_{t-1})]$

$\qquad \mathrm{x}_{t-1} \leftarrow \mathrm{Adagrad}(\mathrm{x}_{t-1}, \Delta \mathrm{x}_{t-1})$

return $p_\theta(\mathrm{w}|\mathrm{x}_0)$

- **Replace** the predicted $\boldsymbol{x_0}$ with the **closest word embedding**.
- Applying the clamping trick to early diffusion steps with **t near T may be sub-optimal.**

Controllable Generation

Multiple Gradient Steps

Sampling Algorithm

θ: parameter of Diffusion-LM, φ: parameter of Attribute Model

$\mathrm{x}_T \sim \mathcal{N}(0, I) \in \mathbb{R}^{L \times d}$

for t in {T,…,1}:

$\mu_t \leftarrow \sqrt{\bar{a}_t} \cdot \mathrm{Clamp}\big(\mu_\theta(\mathrm{x}_t, t)\big)$

$\mathrm{x}_{t-1} \sim \mathcal{N}(\mu_t, 1 - \bar{a}_t) \in \mathbb{R}^{L \times d}$

for k in {1, …, K}:

Increase control strength

$\Delta \mathrm{x}_{t-1} \leftarrow \nabla_{\mathrm{x}_{t-1}}\left[\lambda \|\mu_t - \mathrm{x}_{t-1}\|^2 - \log p_\varphi(c|\mathrm{x}_{t-1})\right]$

$\mathrm{x}_{t-1} \leftarrow \mathrm{Adagrad}(\mathrm{x}_{t-1}, \Delta \mathrm{x}_{t-1})$

return $p_\theta(\mathrm{w}|\mathrm{x}_0)$

Controllable Generation

Fluency Regularization

Sampling Algorithm

θ: parameter of Diffusion-LM, φ: parameter of Attribute Model

$\mathrm{x}_T \sim \mathcal{N}(0, I) \in \mathbb{R}^{L \times d}$

for t in {T,…,1}:

$\quad \mu_t \leftarrow \sqrt{\bar{a}_t} \cdot \mathrm{Clamp}\big(\mu_\theta(\mathrm{x}_t, t)\big)$

$\quad \mathrm{x}_{t-1} \sim \mathcal{N}(\mu_t, 1 - \bar{a}_t) \in \mathbb{R}^{L \times d}$

$\quad$ **for** k in {1, …, K}:

$\quad\quad \Delta \mathrm{x}_{t-1} \leftarrow \nabla_{\mathrm{x}_{t-1}}\left[\lambda \|\mu_t - \mathrm{x}_{t-1}\|^2 - \log p_\varphi(c|\mathrm{x}_{t-1})\right]$

maintain fluency

$\quad\quad \mathrm{x}_{t-1} \leftarrow \mathrm{Adagrad}(\mathrm{x}_{t-1}, \Delta \mathrm{x}_{t-1})$

return $p_\theta(\mathrm{w}|\mathrm{x}_0)$

Controllable Generation

Fluency Regularization

Sampling Algorithm

θ: parameter of Diffusion-LM, φ: parameter of Attribute Model

$\mathrm{x}_T \sim \mathcal{N}(0, I) \in \mathbb{R}^{L\times d}$

for t in {T,…,1}:

$\mu_t \leftarrow \sqrt{\bar{a}_t} \cdot \mathrm{Clamp}\big(\mu_\theta(\mathrm{x}_t, t)\big)$

$\mathrm{x}_{t-1} \sim \mathcal{N}(\mu_t, 1-\bar{a}_t) \in \mathbb{R}^{L\times d}$

for k in {1, …, K}:

$\Delta\mathrm{x}_{t-1} \leftarrow \nabla_{\mathrm{x}_{t-1}}\left[\lambda\|\mu_t - \mathrm{x}_{t-1}\|^2 - \log p_\varphi(c|\mathrm{x}_{t-1})\right]$

$\mathrm{x}_{t-1} \leftarrow \mathrm{Adagrad}(\mathrm{x}_{t-1}, \Delta\mathrm{x}_{t-1})$

return $\boxed{p_\theta(\mathrm{w}|\mathrm{x}_0)}$

Decoding by Minimum Bayes Risk

Control Tasks

Examples

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.

Training Dataset. E2E & ROCStories. **Challenge.** ROCStories > E2E.

Control Tasks

Semantic Content

input (Semantic Content)	food : Japanese
output text	Browns Cambridge is good for Japanese food and also children friendly near The Sorrento .

Given a **field** and **value**, generate a sentence that covers field=value.

- Classifier-Based
- Train an autoregressive LM (GPT-2 small architecture) to predict the (field, value) pair.
  <value> ← **AR-LM(**<sentence> <eos> <field> <eos>**)**

Control Tasks

Parts-of-speech

Generate a sequence of words of the same length whose POS tags match the target.

input (Parts-of-speech)	PROPN AUX DET ADJ NOUN NOUN VERB ADP DET NOUN ADP DET NOUN PUNCT
output text	Zizzi is a local coffee shop located on the outskirts of the city .

- Classifier-Based
- BERT-base architecture

parametrized by a parts-of-speech tagger, which estimates the probability of the target POS sequence conditioned on the latent variables.

Control Tasks

Syntax Tree

Given a target syntactic parse tree, generate text whose syntactic parse matches the given parse.

input (Syntax Tree)	(TOP (S (NP (*) (*) (*)) (VP (*) (NP (NP (*) (*))))))
output text	The Twenty Two has great food

- Classifier-Based
- Transformer-based constituency parser

Constituency parsing with a self-attentive encoder, ACL, 2018

Control Tasks

Syntax Spans

Given a target (span, syntactic category) pair, generate text whose parse tree over span [i, j] matches the target syntactic category.

input (Syntax Spans)	(7, 10, VP)
output text	Wildwood pub serves multicultural dishes and is ranked 3 stars

- Classifier-based
- Use the same parser trained for the syntax tree

Control Tasks

Length & Infilling

Length: Given a target length and generate a sequence with a length within ± 2 of the target.

Infilling: Given a left context and a right context, and generate a sentence to connect them.

- Classifier-Free

input (Length)	14
output text	Browns Cambridge offers Japanese food located near The Sorrento in the city centre .
input (left context)	My dog loved tennis balls.
input (right context)	My dog had stolen every one and put it under there.
output text	One day, I found all of my lost tennis balls underneath the bed.

Control Tasks

Infilling Algorithm

Algorithm 1 Infilling Algorithm

prefix, suffix
infix length
T: number of diffusion steps
p_θ : trained diffusion model
q_θ : forward diffusion process

$x_T(\text{infix}) \sim \text{Gaussian}(0, \text{I})$
for $t \in \{T-1, \ldots, 0\}$ **do**

$$x_{t+1}(\text{prefix}) \sim q_\theta(x_{t+1} \mid \text{EMB}(\text{prefix}))$$
$$x_{t+1}(\text{suffix}) \sim q_\theta(x_{t+1} \mid \text{EMB}(\text{suffix}))$$
$$x_{t+1} = [x_{t+1}(\text{prefix}), x_{t+1}(\text{infix}), x_{t+1}(\text{suffix})]$$
$$x_t \sim p_\theta(x_t \mid x_{t+1})$$

return $\text{Round}(x_0(\text{infix}))$

Main Results

Classifier-Guided

	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

Both PPLM and FUDGE are plug-and-play controllable generation approaches based on AR-LMs trained from scratch using the GPT-2 small architecture.

lm-score. Feed the generated text to a teacher LM (i.e., a carefully fine-tuned GPT-2 model) and report the perplexity of generated text under the teacher LM.

Main Results

Classifier-Guided

	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

	Semantic Content + Syntax Tree			Semantic Content + Parts-of-speech		
	semantic ctrl ↑	syntax ctrl ↑	lm ↓	semantic ctrl ↑	POS ctrl ↑	lm ↓
FUDGE	61.7	15.4	3.52	64.5	24.1	3.52
Diffusion-LM	**69.8**	**74.8**	5.92	**63.7**	**69.1**	3.46
FT-PoE	61.7	29.2	**2.77**	29.4	10.5	**2.97**

lm-score. Feed the generated text to a teacher LM (i.e., a carefully fine-tuned GPT-2 model) and report the perplexity of generated text under the teacher LM.

Main Results

Infilling

	Automatic Eval				Human Eval
	BLEU-4 ↑	ROUGE-L ↑	CIDEr ↑	BERTScore ↑	
Left-only	0.9	16.3	3.5	38.5	n/a
DELOREAN	1.6	19.1	7.9	41.7	n/a
COLD	1.8	19.5	10.7	42.7	n/a
Diffusion	**7.1**	**28.3**	**30.7**	**89.0**	$\mathbf{0.37}^{+0.03}_{-0.02}$
AR	6.7	27.0	26.9	**89.0**	$\mathbf{0.39}^{+0.02}_{-0.03}$

AR-infilling. $x_{i+1} \leftarrow \text{AR-LM}(x_{0:l}; x_{r:\text{end}}; x_{l+1:i}), l < i < r$

Conclusions

This study proposes a **novel text generation model** based on a **Continuous Diffusion Model** and exhibits good performance on six different control tasks.

Advantages

- High flexibility and quality
- Achieved performance similar to or even better than FT-AR without fine-tune.

Drawbacks

- Higher perplexity.
- Decoding is substantially slower.
- Training converges more slowly.

Q & A

Thanks for your attention.

- Doubt

  The GPT used in the experiment is not a model pre-trained with a large amount of data, which is different from the research environment of PPLM.

- Next Version

  DiffuSeq: **Sequence to Sequence** Text Generation with Diffusion Models

- Prediction

  In the near future, there will be research institutions developing large-scale pre-trained Diffusion-based Language Models. (Recorded on 2022/11/07)

Additional Information

PPLM

$$p(x_{i+1}|x_{0:i}, a) \propto p(a|x_{0:i+1})p(x_{i+1}|x_{0:i})$$

$$x_{i+1}, H_{i+1} \leftarrow \mathrm{LM}\left(x_i, H_i + \Delta H_i^{(T)}\right)$$

$$\Delta H_i^{(t+1)} \leftarrow \Delta H_i^{(t)} + \alpha \frac{\nabla_{\Delta H_i} \log p\left(a \middle| H_i + \Delta H_i^{(t)}\right)}{\left\| \nabla_{\Delta H_i} \log p\left(a \middle| H_i + \Delta H_i^{(t)}\right) \right\|^{\gamma}}$$