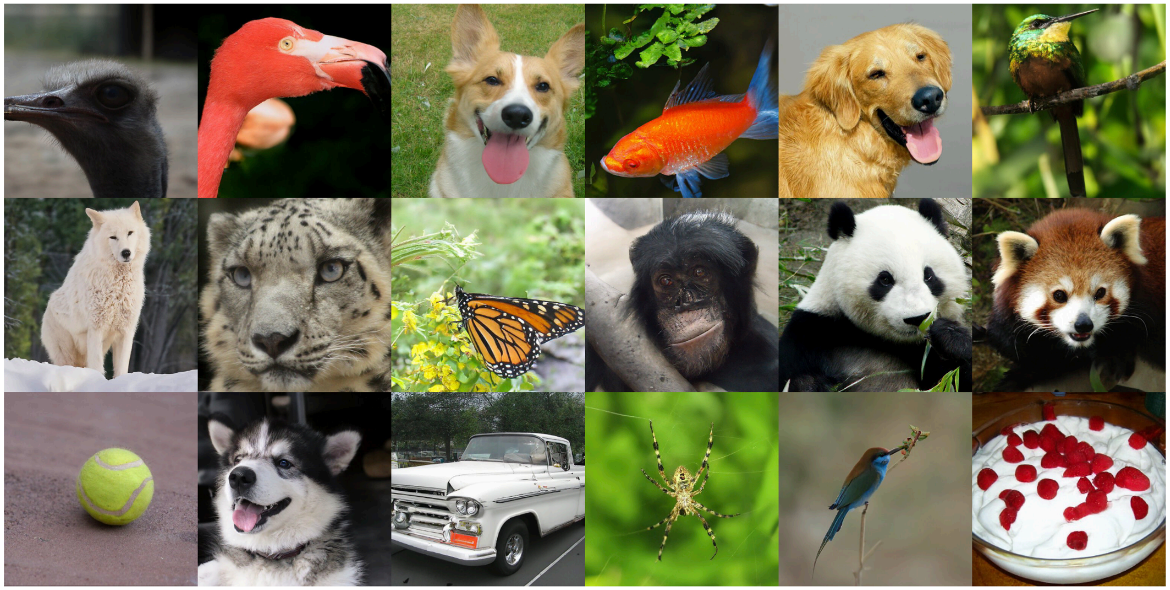
# Diffusion-LM Improves Controllable Text Generation

Xiang Lisa Li, **John Thickstun**, Ishaan Gulrajani, Percy Liang, Tatsunori B. Hashimoto
Stanford University





# Motivating Diffusion-LM



(Dhariwal and Nichol, 2021)

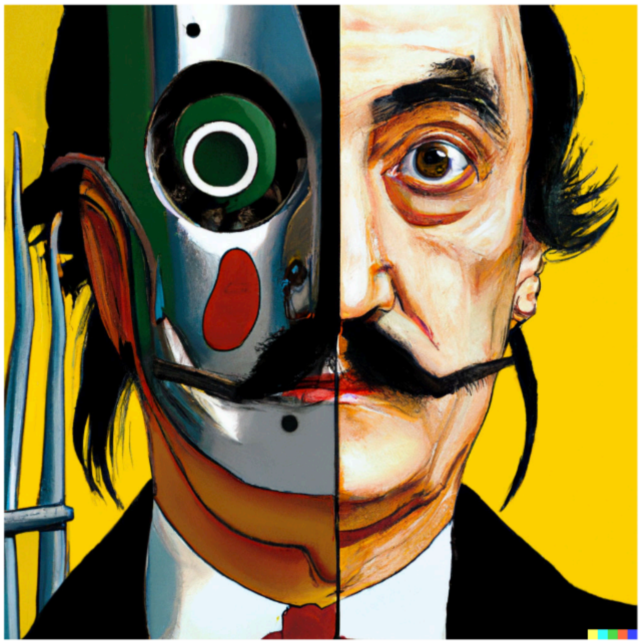
Diffusion models are now dominant in vision. Are they also good for language?

# Motivating Diffusion-LM: Classifier Guidance

- Diffusion models can be easily and convincingly steered using a probabilistic scoring function (e.g., a classifier).
- Analogous to "plug-and-play" language modeling (Dathathri et al., 2020).
- This post-hoc conditioning seems compelling:
  - train one general-purpose (expensive) generative model.
  - steer it for your specialized task at inference time.

(Dhariwal and Nichol, 2021)

# Motivating Diffusion-LM



vibrant portrait painting of Salvador Dalí with a robotic half face

a shiba inu wearing a beret and black turtleneck

a close up of a handpalm with leaves growing from it

(Ramesh et al., 2022)

Classifier Guidance vs. Prompting: competing or complementary paradigms?

# Talk Outline

- Constructing Diffusion-LM
    - ★ The Standard Diffusion Model (Ho et. al., 2020)
    - ★ Learning Word Embeddings (End-to-End Training)
    - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
    - ★ The Clamping Trick and Other Heuristics
    - ★ Classifier-Guided Control
- Experiments
- Discussion

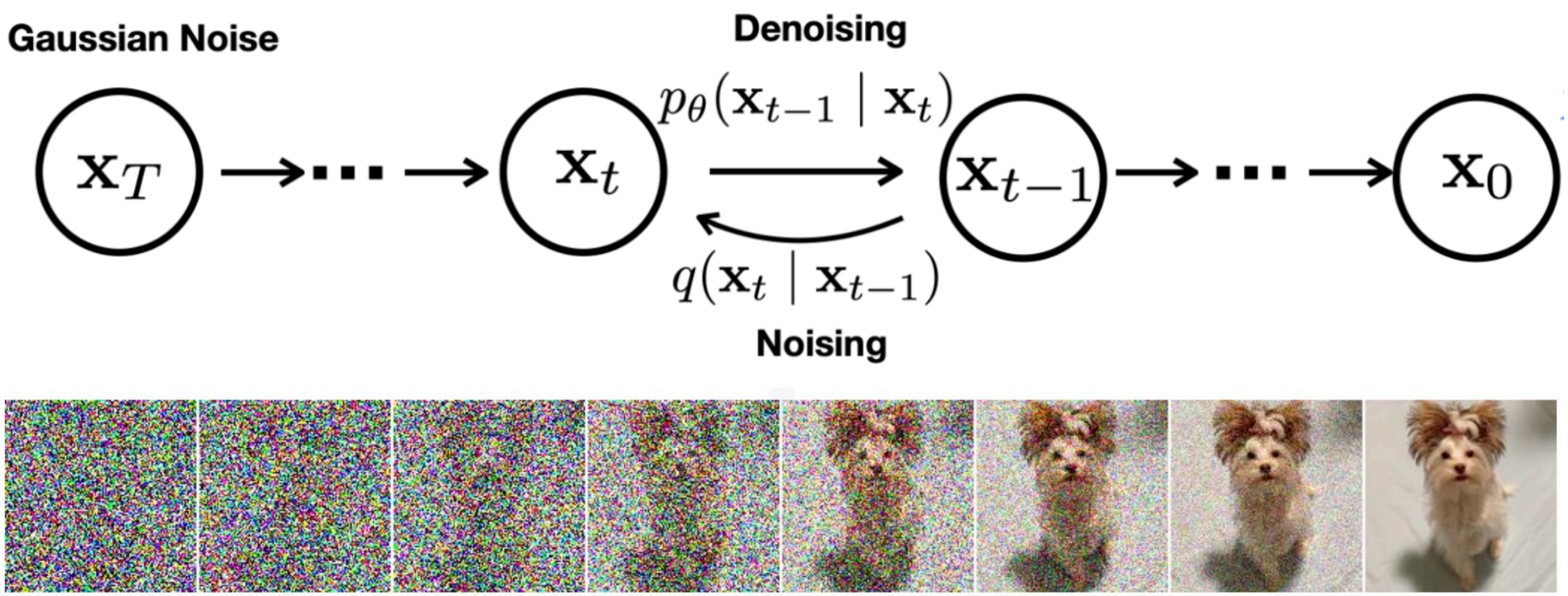
# Talk Outline

- **Constructing Diffusion-LM**
    - ★ The Standard Diffusion Model (Ho et. al., 2020)
    - ★ Learning Word Embeddings (End-to-End Training)
    - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
    - ★ The Clamping Trick and Other Heuristics
    - ★ Classifier-Guided Control
- Experiments
- Discussion

# Talk Outline

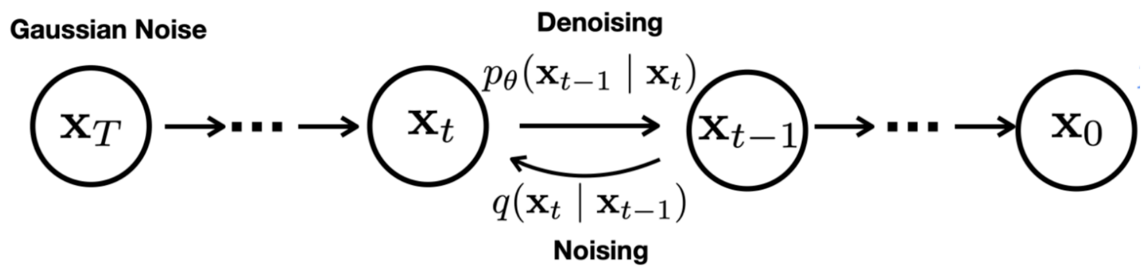
- **Constructing Diffusion-LM**
  - ★ **The Standard Diffusion Model (Ho et. al., 2020)**
  - ★ Learning Word Embeddings (End-to-End Training)
  - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
  - ★ The Clamping Trick and Other Heuristics
  - ★ Classifier-Guided Control
- Experiments
- Discussion

# Denoising Diffusion Probabilistic Models




- Learn to generate data by progressive denoising.

# Denoising Diffusion Probabilistic Models



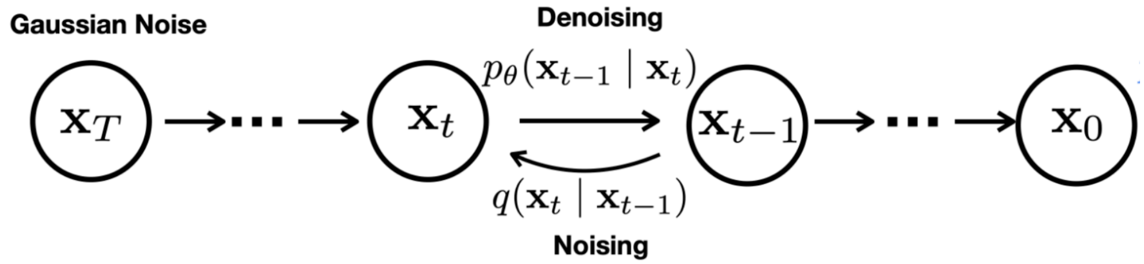

- **Forward Process.** Given *noise schedule* $\beta_1, \ldots, \beta_T$ (hyper-parameters):

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t I\right).$$

- **Reverse Process.** Learn to denoise with parameters $\theta$:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\right).$$

# Optimizing a Diffusion Model



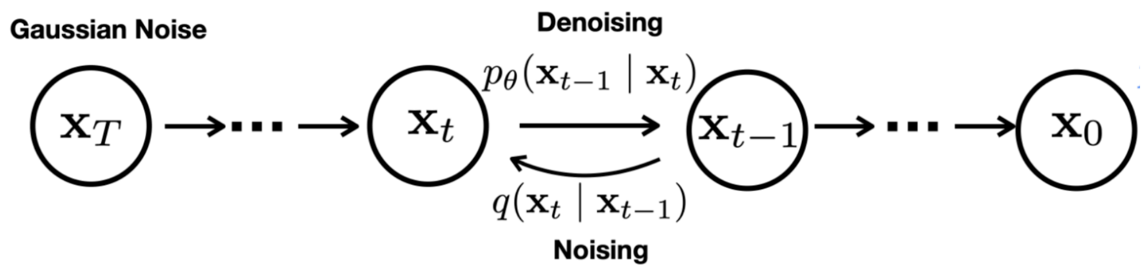

- Optimize like a VAE (the usual variational lower bound):

$$-\log p_\theta(\mathbf{x}_0) = -\log \int_{\mathbf{x}_{1:T}} p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T} = -\log \mathop{\mathbb{E}}_{\mathbf{x}_{1:T}\sim q}\left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]$$

$$\leq \mathop{\mathbb{E}}_{\mathbf{x}_{1:T}\sim q}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathop{\mathbb{E}}_{\mathbf{x}_{1:T}\sim q}\left[-\log q(\mathbf{x}_T) - \sum_{t=1}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right].$$

- Many small steps $\beta_t$ make the Gaussian approximation $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ valid.

# In What Sense is this Denoising?




- Rewrite the variational objective (algebra; see Ho et al., 2020; Appendix A):

$$\mathop{\mathbb{E}}_{\mathbf{x}_{1:T}\sim q}\left[D(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p(\mathbf{x}_T)) + \sum_{t=1}^{T} D(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right].$$

- Want to make $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ look like the posterior distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$.

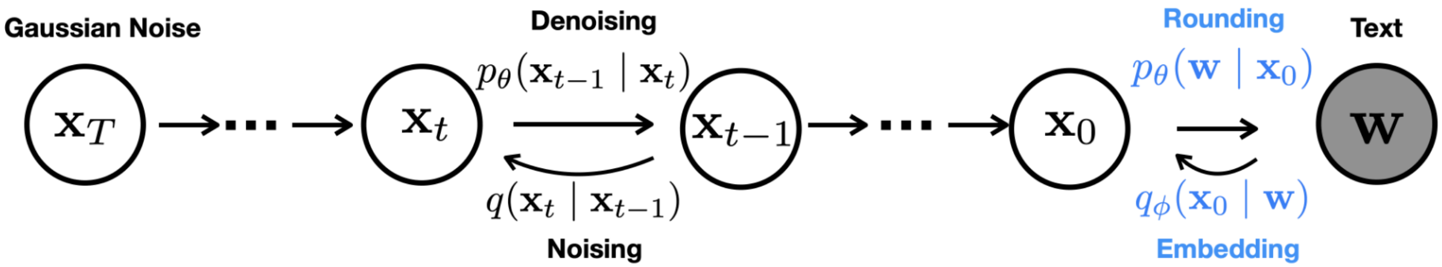
# Talk Outline

- **Constructing Diffusion-LM**
    - ★ The Standard Diffusion Model (Ho et. al., 2020)
    - ★ Learning Word Embeddings (End-to-End Training)
    - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
    - ★ The Clamping Trick and Other Heuristics
    - ★ Classifier-Guided Sampling
- Experiments
- Discussion

# Talk Outline

- **Constructing Diffusion-LM**
    - ★ The Standard Diffusion Model (Ho et. al., 2020)
    - ★ **Learning Word Embeddings (End-to-End Training)**
    - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
    - ★ The Clamping Trick and Other Heuristics
    - ★ Classifier-Guided Sampling
- Experiments
- Discussion

# Learning Word Embeddings




- Standard DDPM assumes inputs $\mathrm{x}_0$ are continuous.
- Language model inputs $\mathrm{w}$ are discrete tokens.
- Use the re-parameterization trick to learn word embeddings $q_\phi(\mathbf{x}_t|\mathbf{w})$:

$$-\log p_\theta(\mathbf{x}_0) \le \mathop{\mathbb{E}}_{\substack{\mathbf{x}_{1:T}\sim q \\ \mathbf{x}_0 \sim q_\phi}} \left[ -\log q(\mathbf{x}_T) - \sum_{t=1}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} - \log p_\theta(\mathbf{w}|\mathbf{x}_0) \right].$$

# Are These Learned Embeddings Meaningful?

- Learning the embedding seems important.
- Random embeddings performed poorly.
- Pre-trained embeddings from an AR model also performed poorly.


A t-SNE plot of learned embeddings, colored according to POS.

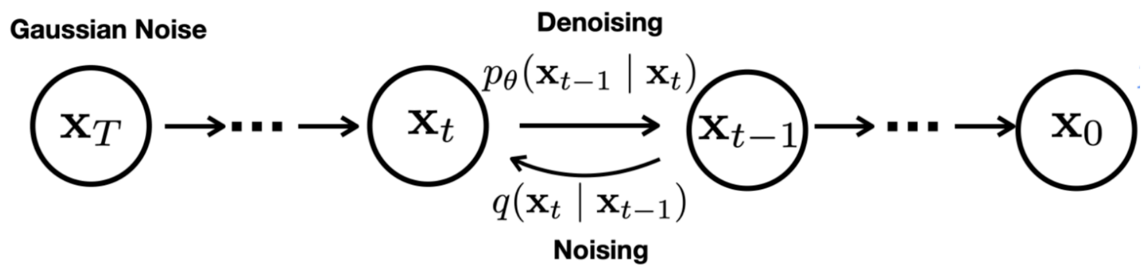
# Talk Outline

- **Constructing Diffusion-LM**
    - ★ The Standard Diffusion Model (Ho et. al., 2020)
    - ★ Learning Word Embeddings (End-to-End Training)
    - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
    - ★ The Clamping Trick and Other Heuristics
    - ★ Classifier-Guided Sampling
- Experiments
- Discussion

# Talk Outline

- **Constructing Diffusion-LM**
  - ★ The Standard Diffusion Model (Ho et. al., 2020)
  - ★ Learning Word Embeddings (End-to-End Training)
  - ★ **Predicting the Noiseless Embeddings**
- Sampling from Diffusion-LM
  - ★ The Clamping Trick and Other Heuristics
  - ★ Classifier-Guided Sampling
- Experiments
- Discussion

# Predicting the Noiseless Embeddings




- We want to minimize terms $D(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$, where

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\right).$$

- Closed form for the posteriors: $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t I)$,

  where $\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = r_t \mathbf{x}_t + s_t \mathbf{x}_0$.

- And $r_t, s_t, \tilde{\beta}_t$ are constants derived from the noise schedule $\beta_1, \ldots, \beta_T$.

# Predicting the Noiseless Embeddings

- Directly parameterize $\mu_\theta(\mathbf{x}_t, t)$ to approximate $\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = r_t\mathbf{x}_t + s_t\mathbf{x}_0$ ?

- But we already know $\mathbf{x}_t$.

- Ho et al., 2020: write $\mathbf{x}_0 = \mathbf{x}_t - \boldsymbol{\varepsilon}_t$, predict $\boldsymbol{\varepsilon}_t \approx \boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t)$ and reparameterize

$$\mu_\theta(\mathbf{x}_t, t) = (r_t + s_t)\mathbf{x}_t - s_t\boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t).$$

- Li et al., 2022: predict $\mathbf{x}_0 \approx f_\theta(\mathbf{x}_t, t)$ and reparameterize

$$\mu_\theta(\mathbf{x}_t, t) = r_t\mathbf{x}_t + s_t f_\theta(\mathbf{x}_t, t).$$

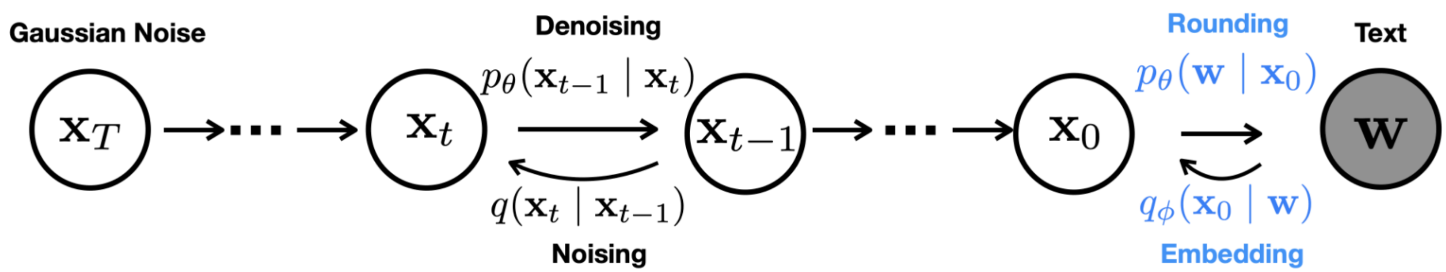
# Talk Outline

- Constructing Diffusion-LM
  - ★ The Standard Diffusion Model (Ho et. al., 2020)
  - ★ Learning Word Embeddings (End-to-End Training)
  - ★ Predicting the Noiseless Embeddings
- **Sampling from Diffusion-LM**
  - ★ The Clamping Trick and Other Heuristics
  - ★ Classifier-Guided Sampling
- Experiments
- Discussion

# Talk Outline

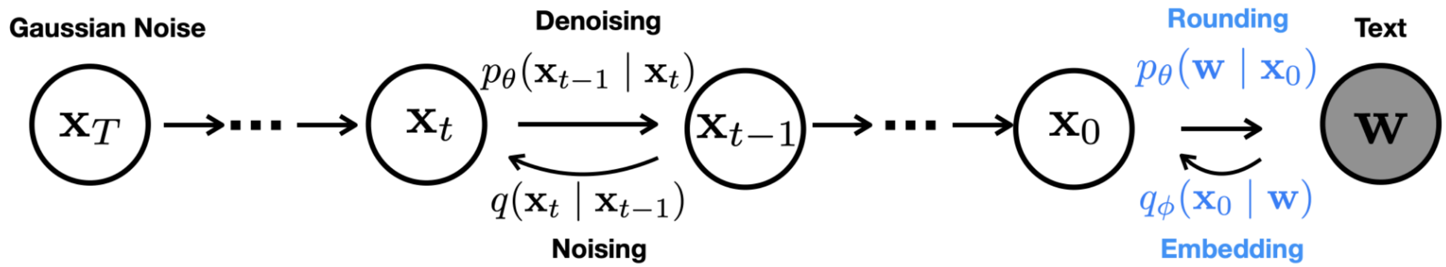
- Constructing Diffusion-LM
    - ★ The Standard Diffusion Model (Ho et. al., 2020)
    - ★ Learning Word Embeddings (End-to-End Training)
    - ★ Predicting the Noiseless Embeddings
- **Sampling from Diffusion-LM**
    - ★ **The Clamping Trick and Other Heuristics**
    - ★ Classifier-Guided Sampling
- Experiments
- Discussion

# Sampling from Diffusion-LM



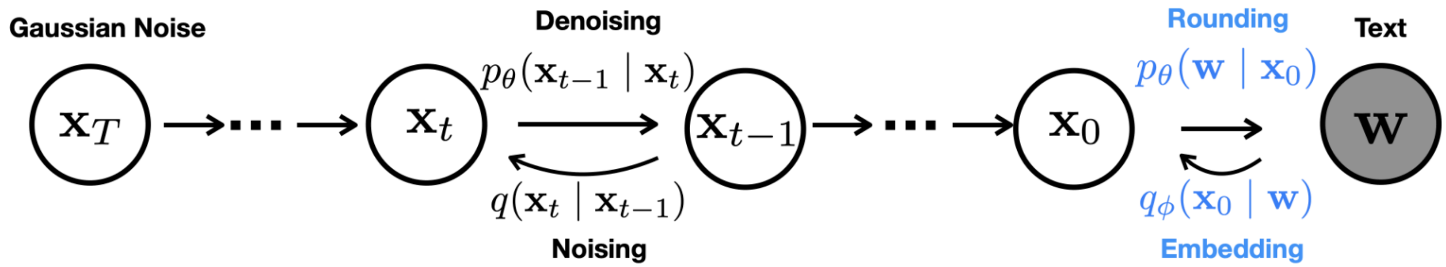

- Sample from a Gaussian $\mathbf{x}_T \sim \mathcal{N}(0, I)$ and then just follow the chain.
- Sampling Diffusion-LM requires $T$ (diffusion steps) model calls.
- In contrast, sampling AR models requires $L$ (sequence length) model calls.
- In our experiments, $T \gg L$ so sampling is slow.

# Sampling Heuristics




- Analogous AR sampling heuristics can be applied to Diffusion-LM.
- Temperature sampling: reduce the noise at each sampling step.
- Nucleus sampling: truncate the tails of the Gaussian noise.
- Minimum Bayes Risk (MBR) decoding.

# The Clamping Trick




- At each step we predict $\mathbf{x}_0 \approx f_\theta(\mathbf{x}_t, t)$.
- *The clamping trick*: instead predict the nearest embedding in the dictionary:

$$\mathbf{x}_0 \approx \mathrm{Clamp}(f_\theta(\mathbf{x}_t, t)).$$

- This seems to nudge Diffusion-LM to commit to tokens earlier.

# Talk Outline

- Constructing Diffusion-LM
    - ★ The Standard Diffusion Model (Ho et. al., 2020)
    - ★ Learning Word Embeddings (End-to-End Training)
    - ★ Predicting the Noiseless Embeddings
- **Sampling from Diffusion-LM**
    - ★ The Clamping Trick and Other Heuristics
    - ★ Classifier-Guided Sampling
- Experiments
- Discussion

# Talk Outline

- Constructing Diffusion-LM
    - ★ The Standard Diffusion Model (Ho et. al., 2020)
    - ★ Learning Word Embeddings (End-to-End Training)
    - ★ Predicting the Noiseless Embeddings
- **Sampling from Diffusion-LM**
    - ★ The Clamping Trick and Other Heuristics
    - ★ **Classifier-Guided Sampling**
- Experiments
- Discussion

# Classifier-Guided Sampling

- Given: labeled data pairs $(\mathbf{x}, \mathbf{c})$ describing the desired attribute $\mathbf{c}$.
- Goal: sample from the posterior distribution $p_\theta(\mathbf{x}_{0:T}|\mathbf{c}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})$.
- Each term can be rewritten as $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \propto p_\theta(\mathbf{c}|\mathbf{x}_{t-1}, \mathbf{x}_t) p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$.
- The term $p_\theta(\mathbf{c}|\mathbf{x}_{t-1}, \mathbf{x}_t)$ is a classifier. And $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is Diffusion-LM.
- In practice, the classifier doesn't need to see both $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$.
- Train the classifier $p_\theta(\mathbf{c}|\mathbf{x}_{t-1})$ on noisy data.

# Langevin Dynamics

- Goal: sample from the posterior $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})$.

- Langevin Dynamics: define a Markov chain

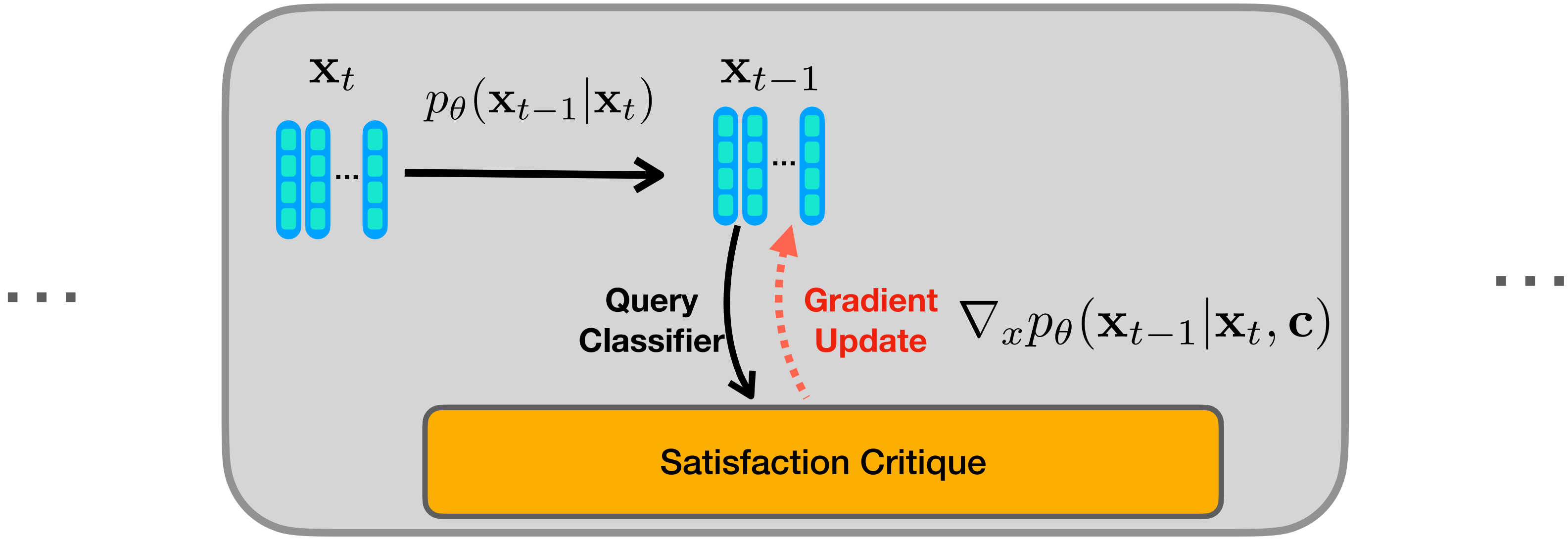
$$\mathbf{x}_{t-1}^{(i+1)} = x_{t-1}^{(i)} - \eta \nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) + \sqrt{2\eta}\varepsilon_i, \text{ where } \varepsilon_i \sim \mathcal{N}(0, I).$$

- For small $\eta$ , as $i \to \infty$, $D(\mathbf{x}_{t-1}^{(i+1)} \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})) \to 0$. The gradient is:

$$\nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) = \underbrace{\boxed{\nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{c}|\mathbf{x}_{t-1})}}_{\text{Classifier Score}} + \underbrace{\boxed{\nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t).}}_{\text{Diffusion-LM}}$$

- In practice, init $\mathbf{x}_{t-1}^{(0)} \sim p_\theta(\mathbf{x}_{t-1}, \mathbf{x}_t^{(1)})$ (warmstart) and take just one step.

# Iterative Gradient-Based Control




- Conceptually similar to PPLM (Dathathri et al., 2020).
- But control is applied coarse-to-fine, instead of left-to-right.

# Talk Outline

- Constructing Diffusion-LM
    - ★ The Standard Diffusion Model (Ho et. al., 2020)
    - ★ Learning Word Embeddings (End-to-End Training)
    - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
    - ★ The Clamping Trick and Other Heuristics
    - ★ Classifier-Guided Sampling
- Experiments
- Discussion

# Talk Outline

- Constructing Diffusion-LM
  - ★ The Standard Diffusion Model (Ho et. al., 2020)
  - ★ Learning Word Embeddings (End-to-End Training)
  - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
  - ★ The Clamping Trick and Other Heuristics
  - ★ Classifier-Guided Sampling
- **Experiments**
- Discussion

# Datasets

- Two datasets: **E2E** and **ROCStories**.

- **E2E**. 50k restaurant reviews. Sample text: “Browns Cambridge is good for Japanese food and also children friendly near The Sorrento.”

- **ROCStores**. 98k short stories. Sample text: "Jennifer has a big exam tomorrow. She got so stressed, she pulled an all-nighter. She went into class the next day, weary as can be. Her teacher stated that the test is postponed for next week. Jennifer felt bittersweet about it.”

- Small datasets: scaling up Diffusion-LM is an open problem.

# Control Tasks

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

- Six controllable generation tasks.

# Baselines

- For classifier-guided control:
  - PPLM (Dathathri et al., 2020).
  - FUDGE (Yang and Klein, 2021).
  - Finetuning (skyline).
- For infilling:
  - DELOREAN (Qin et al., 2020).
  - COLD (Qin et al., 2021).
  - Finetuning (skyline).

# A Qualitative Example

| | |
|---|---|
| Syntactic Parse | ( S ( S ( NP * ) ( VP * ( NP ( NP * * ) ( VP * **( NP ( ADJP * * ) * )** ) ) ) ) ) * ( S ( NP * * * ) ( VP * ( ADJP ( ADJP * ) ) ) ) ) ) |
| FUDGE | Zizzi is a cheap restaurant . **[incomplete]** |
| Diffusion-LM | Zizzi is a pub providing **family friendly Indian food** Its customer rating is low |
| FT | Cocum is a Pub serving **moderately priced meals** and the customer rating is high |
| Syntactic Parse | ( S ( S ( VP * ( PP * ( NP * * ) ) ) ) * **( NP * * * )** ( VP * ( NP ( NP * * ) ( SBAR ( WHNP * ) ( S ( VP * ( NP * * ) ) ) ) ) ) ) * ) |
| FUDGE | In the city near The Portland Arms is a coffee and fast food place named The Cricketers which is not family - friendly with a customer rating of 5 out of 5 . |
| Diffusion-LM | Located on the riverside , **The Rice Boat** is a restaurant that serves Indian food . |
| FT | Located near The Sorrento, **The Mill** is a pub that serves Indian cuisine. |

- FUDGE and Finetuning (FT) deviate after a few tokens (exposure bias).
- Diffusion-LM is robust to local failures to apply the control.

# Talk Outline

- Constructing Diffusion-LM
  - ★ The Standard Diffusion Model (Ho et. al., 2020)
  - ★ Learning Word Embeddings (End-to-End Training)
  - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
  - ★ The Clamping Trick and Other Heuristics
  - ★ Classifier-Guided Sampling
- Experiments
- Discussion

# Talk Outline

- Constructing Diffusion-LM
  - ★ The Standard Diffusion Model (Ho et. al., 2020)
  - ★ Learning Word Embeddings (End-to-End Training)
  - ★ Predicting the Noiseless Embeddings
- Sampling from Diffusion-LM
  - ★ The Clamping Trick and Other Heuristics
  - ★ Classifier-Guided Sampling
- Experiments
- **Discussion**

# Limitations

- Decoding is much slower than AR Transformer models: 2000 diffusion steps versus (short) sequence-length AR steps.
- Training seems to converge more slowly than for AR models.
- Diffusion-LM has higher perplexity than comparably-sized AR models.
- Similar challenges for diffusion models of images have been observed and overcome, so there is reason to be hopeful!

# Coarse-to-Fine Generation

- Coarse-to-fine generation seems helpful for control (versus autoregressive generation) because the control target can be incorporated globally into the plan for generating text.

- You can use Langevin dynamics to sample coarse-to-fine from AR models fine-tuned in noisy data (Jayaram and Thickstun, 2021)

- A mystery: I spent several months trying to adapt these methods to text, but couldn't get the sampling to work very well.

- Maybe coarse-to-fine isn't the only thing that's going right here?

# Non-Autoregressive Language Generation

- Previously non-autoregressive open-ended language generation has seemed difficult (GAN, VAE).
- Adapting the DDPM recipe to text required some alterations, but if similarly-scoped alterations would make VAE's work well for text it seems like this would have happened by now.
- What (if anything) is different about diffusion models?

# Continuous vs. Discrete Models

- NLP folks spend a lot of time trying to convert their discrete data into continuous representations.
- Vision folks spend a lot of time trying to convert their continuous data into discrete representations.
  - VQ-VAE2 (Razavi et al., 2019)
  - VQ-GAN (Esser et al., 2021)
  - Parti (Yu et al., 2022)
- What is going on here?

# Thank You!

**Paper:** https://arxiv.org/abs/2205.14217

**Code:** https://github.com/XiangLi1999/Diffusion-LM