Diffusion-LM Improves Controllable Text Generation

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

NeurIPS 2022

Max Mehta

Outline

- Motivation
- Related Works
- Approach
- Experiments
- Results
- Conclusion

Motivation

 Motivation

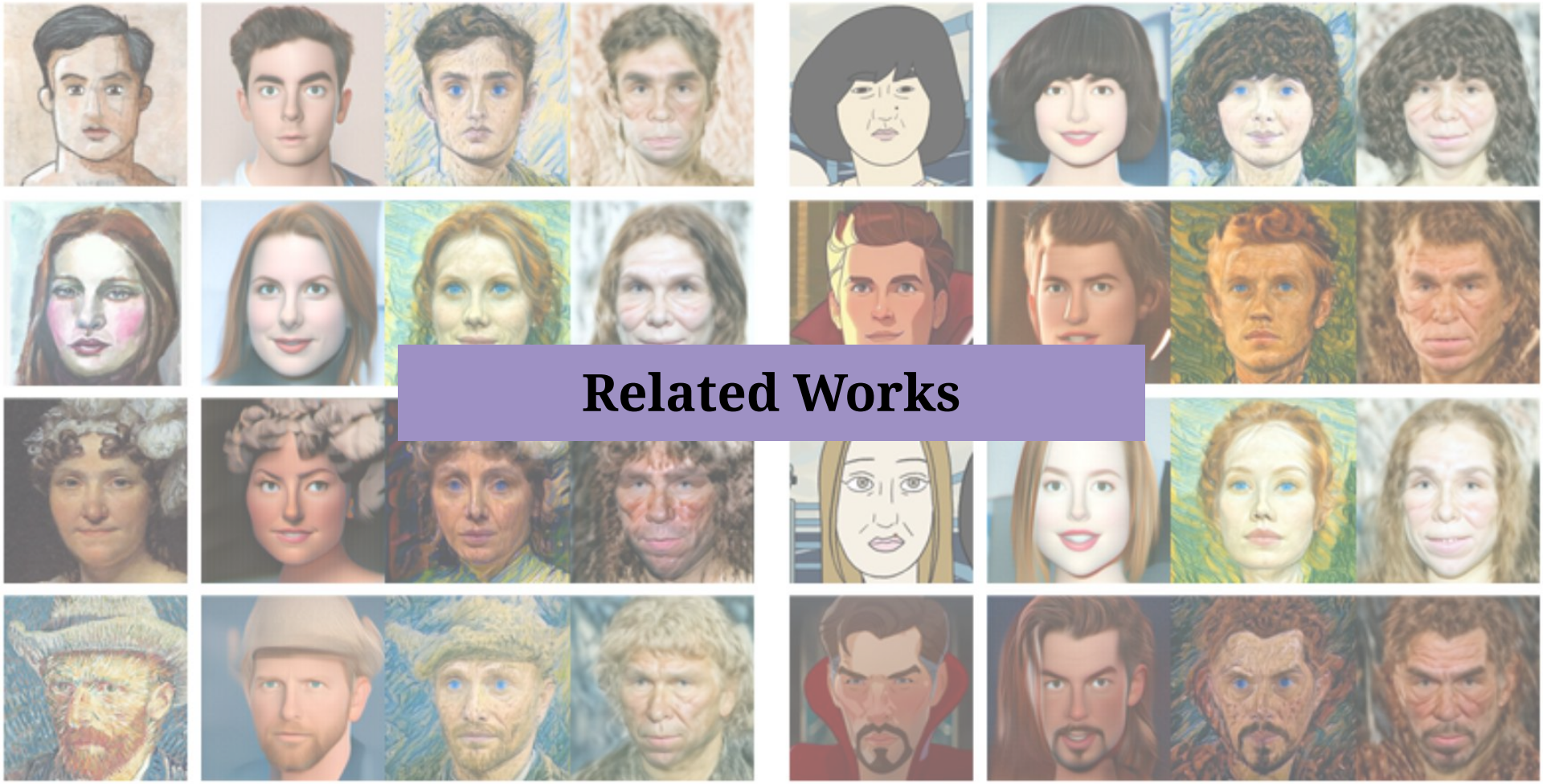
- ➔ Controlling the behavior of language models (LMs)
  - ◆ autoregressive (text generation) ⟶ controllable (real-world deployment)
  - ◆ simple sentence attributes (sentiment) ⟶ complex, fine-grained control (syntactic structure)

- ➔ Fine-tune with supervised data (control, text)
  - ◆ expensive
  - ◆ ❌ multiple controls

- ➔ Plug-and-play is the answer
  - ◆ LM frozen - external classifier (guides generation + satisfy control)
  - ◆ Guiding frozen autoregressive LM is hard - limited to sentiment or topic

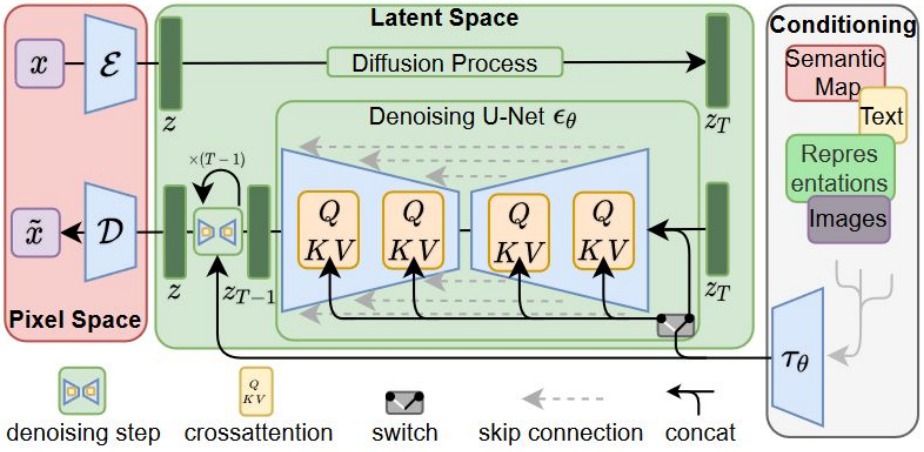
Proposal

- ➔ Diffusion-LM
  - non-autoregressive
  - continuous diffusion
- ➔ Continuous diffusion for discrete text (novel) - need modifications
  - embedding step & rounding step
- ➔ Six control tasks (4 classifier-guided + 2 classifier-free) - semantic & structure
  - individual control tasks + multiple classifier-guided controls
  - outperforms/on-par with prior plug-and-play & autoregressive LM



Related Works

Latent Diffusion



Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). High-Resolution Image Synthesis with Latent Diffusion Models. *ArXiv*. /abs/2112.10752

Denoising diffusion probabilistic models (DDPM)

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) + \sigma_t \boldsymbol{z}$$

Markov Chain

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. *ArXiv*. /abs/2006.11239

Denoising diffusion implicit models (DDIM)

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \sigma_t^2 \boldsymbol{z}, \quad (5)$$

where, $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t)$ is a the prediction of $\boldsymbol{x}_0$ at t given $\boldsymbol{x}_t$ and $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$:

$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) := \frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}}. \quad (6)$$

Non-Markovian
Faster
Deterministic
Random noise is zeroed

Song, J., Meng, C., & Ermon, S. (2020). Denoising Diffusion Implicit Models. ArXiv. /abs/2010.02502

Forward

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) := \mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \mathbf{I})$$

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{w}, \quad \boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

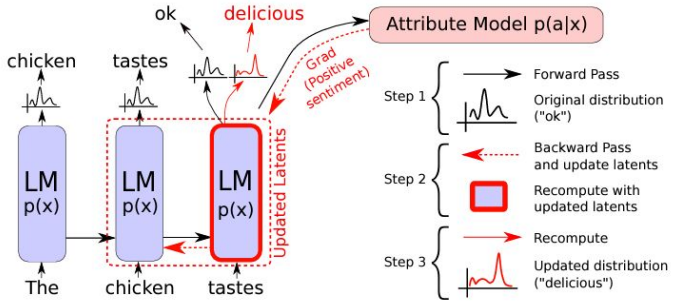
Reverse (Generative)

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}(\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)\mathbf{I})$$

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right)$$

noise approximation model (Trained)

Plug and Play Language Models



[–] The potato and cauliflower are both in season to make combo breads, mounds, or pads. For an added challenge, try some garlic mashed potatoes.
[Negative] The potato is a pretty bad idea. It can make you fat, it can cause you to have a terrible immune system, and it can even kill you. . .
[Positive] The potato chip recipe you asked for! We love making these, and I've been doing so for years. I've always had a hard time keeping a recipe secret. I think it's the way our kids love to eat them – so many little ones.
[Science] The potato was once thought to have no health problems and has been promoted as a nutritious food source since the mid-1800s, but recent reports indicate that it has many harmful health issues. In fact, researchers from Johns Hopkins University. . .
[Politics] [Positive] To conclude this series of articles, I will present three of the most popular and influential works on this topic. The first article deals with the role of women's political participation in building a political system that is representative of the will of the people.
[Politics] [Negative] To conclude, the most significant and lasting damage from the economic crisis in 2008 was that many governments, including those in the political center, lost power for the first time in modern history.

- ➔ Controllable text generation
- ➔ Pretrained LM + attribute classifiers
- ➔ Modular (no retraining needed)
- ➔ Good performance
  - high fluency (perplexity)
  - control (classifier accuracy)
  - human eval
- ➔ Simple coarse-grained control
  - sentiment
  - switching topic

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., & Liu, R. (2019). Plug and Play Language Models: A Simple Approach to Controlled Text Generation. *ArXiv*. /abs/1912.02164

Approach

Diffusion-LM

- Controllable text generation - sample w from a conditional distribution p(w | c)
  - w = [$w_1 \cdots w_n$] - sequence of discrete words
  - c - control variable
  - Goal: generate w that satisfies the control target c
- Diffusion-LM (continuous + non-autoregressive)
- Gaussian noise vectors ⟶ (denoise) ⟶ continuous latent representation ⟶ (denoise) ⟶ word vectors
  - gradient-based methods - complex control tasks

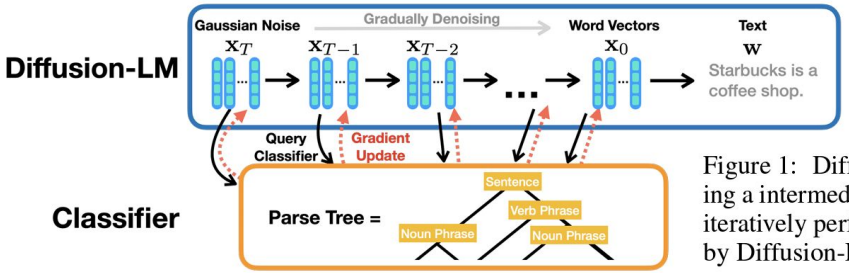


Figure 1: Diffusion-LM iteratively denoises a sequence of Gaussian vectors into word vectors, yielding a intermediate latent variables of decreasing noise level $\mathbf{x}_T \cdots \mathbf{x}_0$. For controllable generation, we iteratively perform gradient updates on these continuous latents to optimize for fluency (parametrized by Diffusion-LM) and satisfy control requirements (parametrized by a classifier).

Diffusion-LM Modifications

- Continuous diffusion for discrete text (novel) - need modifications
- Embedding function
  - discrete text ⟶ continuous space
  - end-to-end training objective (learn embeddings)
- Rounding method
  - vectors in embedding space ⟶ discrete text
  - re-parametrization + clamping

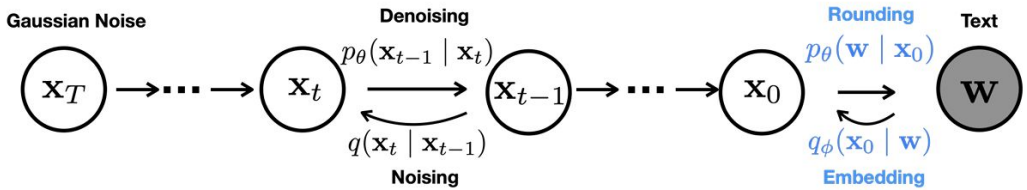


Figure 2: A graphical model representing the forward and reverse diffusion processes. In addition to the original diffusion models [12], we add a Markov transition between $\mathbf{x}_0$ and $\mathbf{w}$, and propose the embedding §4.1 and rounding §4.2 techniques.

End-to-end Training

➔ Embedding function

$$\text{EMB}(\mathbf{w}) = [\text{EMB}(w_1), \ldots, \text{EMB}(w_n)] \in \mathbb{R}^{nd}$$ sequence w of length n

◆ $\text{EMB}(w_i)$: word ⟶ vector in $\mathbb{R}^d$

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

➔ Training objective (DDPM) - stable

➔ MSE

➔ $\mu_\theta(x_t, t)$: predicted mean of $p_\theta(x_{t-1} \mid x_t)$ → reverse process

➔ Embedding Markov transition (forward)

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

➔ Rounding Markov transition (reverse)

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

➔ New training objective

$$\mathcal{L}^{\text{e2e}}_{\text{simple}}(\mathbf{w}) = \mathop{\mathbb{E}}_{q_\phi(\mathbf{x}_{0:T}|\mathbf{w})} \left[\mathcal{L}_{\text{simple}}(\mathbf{x}_0) + ||\text{EMB}(\mathbf{w}) - \mu_\theta(\mathbf{x}_1, 1)||^2 - \log p_\theta(\mathbf{w}|\mathbf{x}_0)\right]$$

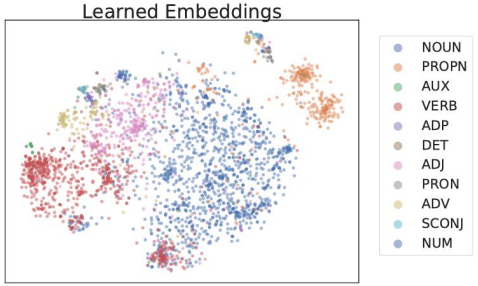


Figure 3: A t-SNE [41] plot of the learned word embeddings. Each word is colored by its POS.

Reducing Rounding Errors

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

- x_0 does not commit to a single word embedding
  - $L_{simple}(x_0)$ ❌ model the structure of x_0 well (constraint: $t \to 0$)
  - Solution: reparametrize $L_{simple}(x_0)$

$$\mathcal{L}_{\text{simple}}(\mathbf{x}_0) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{q(\mathbf{x}_t|\mathbf{x}_0)} ||\mu_\theta(\mathbf{x}_t, t) - \hat{\mu}(\mathbf{x}_t, \mathbf{x}_0)||^2 \quad \to \quad \mathcal{L}^{\text{e2e}}_{\mathbf{x}_0\text{-simple}}(\mathbf{x}_0) = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}_t} ||f_\theta(\mathbf{x}_t, t) - \mathbf{x}_0||^2$$

  - ➔ forced to predict x_0 in every term - x_0 centered at a single word embedding

- Clamping during decoding
  - Reparametrize reverse generation process

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}} f_\theta(\mathbf{x}_t, t) + \sqrt{1-\bar{\alpha}}\epsilon \quad \to \quad \mathbf{x}_{t-1} = \sqrt{\bar{\alpha}} \cdot \text{Clamp}(f_\theta(\mathbf{x}_t, t)) + \sqrt{1-\bar{\alpha}}\epsilon$$

Map $f_\theta(x_t,t)$ (estimate of x_0) to nearest word embedding - $f_\theta(x_t,t)$ centered at a single word embedding for every step

- ➔ $f_\theta(x_t,t)$ predictions more accurate + rounding errors reduced

Controlling text generation

- Control $x_{0:T}$ (continuous latent variables)
- Gradient update on x_{t-1} (time step t)

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

parametrized by Diffusion-LM

parametrized by NN classifier

- Classifier trained on latent variables
- Fluency regularization
  - generate fluent text
  - stochastic

$$\lambda \underbrace{\log p(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}_{\text{fluency}} + \underbrace{\log p(\mathbf{c} \mid \mathbf{x}_{t-1})}_{\text{control}}$$

λ - hyperparameter

- Multiple gradient steps
  - 3 steps of Adagrad update per diffusion step
  - ↑ computation - downsample diffusion steps (2000 ⟶ 200) - ↑ controlled generation speed

Controlling text generation

- Minimum Bayes Risk (MBR) decoding
  - single high-quality output
  - machine translation, sentence infilling

$$\hat{\mathbf{w}} = \text{argmin}_{\mathbf{w} \in S} \sum_{\mathbf{w}' \in S} \frac{1}{|S|} \underbrace{\mathcal{L}(\mathbf{w}, \mathbf{w}')}_{\text{negative BLEU score}}$$

Sample with minimum expected risk under L

→ aggregate samples S from Diffusion-LM

Experiments

Experiments

- Trained on 2 datasets

Dataset	Size	Composition
E2E	50K restaurant reviews	8 fields including food type, price, and customer rating
ROCStories	98K five-sentence stories	causal and temporal commonsense relations between daily events

- Diffusion-LM model
  - Transformer (80M parameters)
  - sequence length n = 64, diffusion steps T = 2000, square-root noise schedule
  - embedding dimension: d = 16 for E2E & d = 128 for ROCStories
  - decoding time diffusion steps T = 200 for E2E & T = 2000 for ROCStories

Experiments - Control Tasks

- 6 control tasks (4 classifier-guided + 2 classifier-free)
  - sample 200 control targets from val - 50 samples for each control target
- Fluency: generated text ⟶ teacher LM (GPT-2)
  - perplexity of generated text under the teacher LM - metric: lm-score (lower = better sample quality)

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.

Table 1: Example input control and output text for each control tasks.

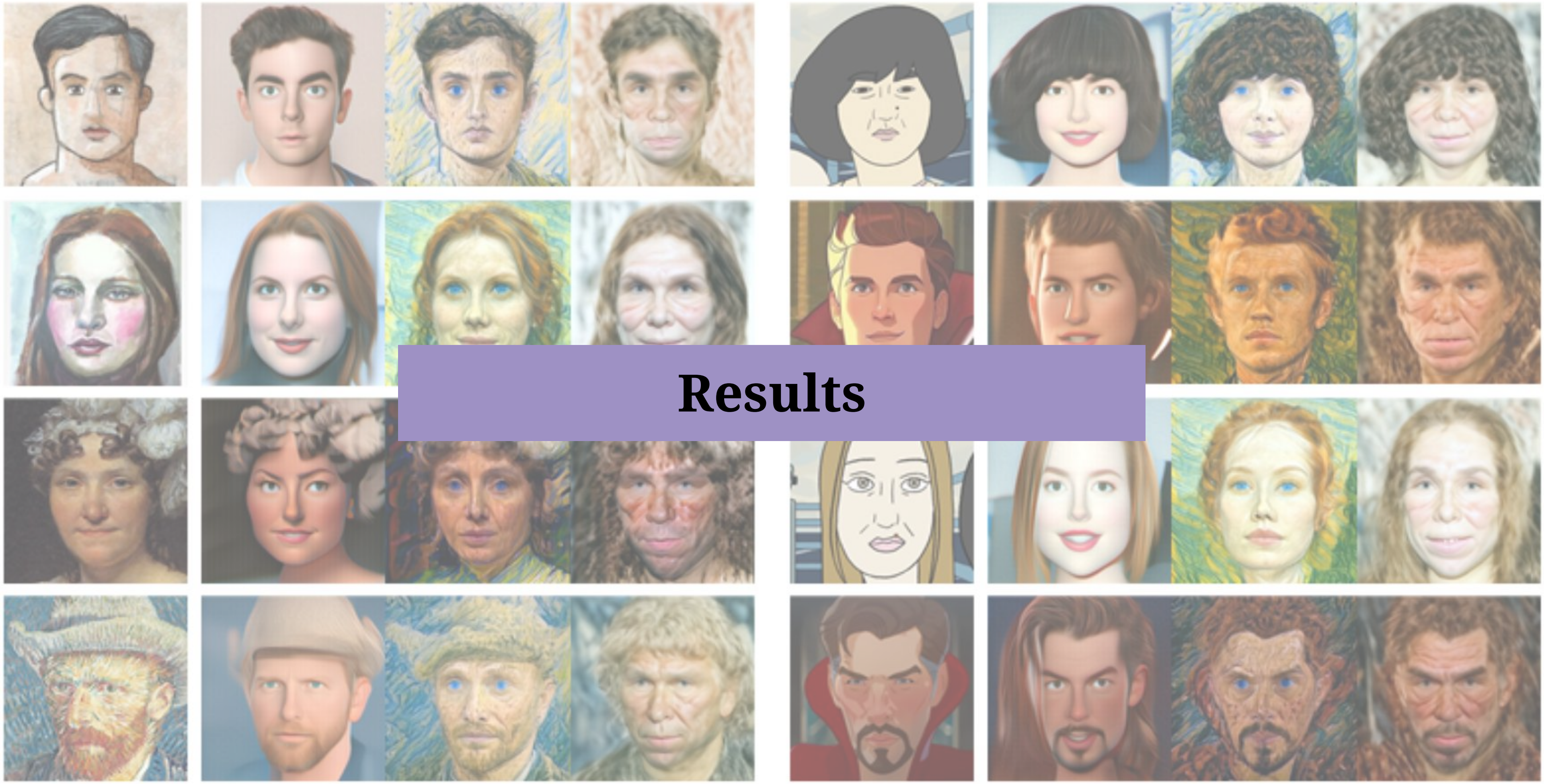
Experiments - Control Tasks

Control Task	Evaluation Method	Success Metric
Semantic Content	**Given:** field & value **Task:** generate a sentence where field=value	exact match of value
Parts-of-speech	**Given:** sequence of POS tags **Task:** generate sentence of same length with matching POS tags	word-level exact match
Syntax Tree	**Given:** syntactic parse tree **Task:** generate text with same syntactic parse	F1 scores from off-the-shelf parser
Syntax Spans	**Given:** (span, syntactic category) pair **Task:** generate text with same syntactic parse tree over span	fraction of spans that match exactly
Length	**Given:** length **Task:** generate a sequence of ±2 length	fraction of correct lengths
Infilling	**Given:** left context (O_1) & right context (O_2) from aNLG dataset **Task:** sentence that logically connects O_1 and O_2	automatic and human evaluation

Experiments - Baselines

Baseline	Description	Control Tasks
PPLM	Plug-and-play autoregressive LM trained from scratch on GPT-2 No positional information	Semantic Content
FUDGE	Plug-and-play autoregressive LM trained from scratch on GPT-2	Semantic Content, Parts-of-speech, Syntax Tree, Syntax Spans, Length
FT	Oracle conditional LM (fine-tuned GPT-2) FT-sample (sampling) & FT-search (beam search)	Semantic Content, Parts-of-speech, Syntax Tree, Syntax Spans, Length
DELOREAN	autoregressive LM left-to-right fluency	Infilling
COLD	energy-based model left-to-right & right-to-left fluency + coherence	Infilling
AR-infilling	train autoregressive LM from scratch with ROCStories preprocess: (O_1, O_{middle}, O_2) to (O_1, O_2, O_{middle})	Infilling

Results



Main Results

- ➔ Diffusion-LM on E2E and ROCStories
  - negative log-likelihood (NLL) - lower is better
  - underperforms autoregressive models
  - ↑ model & dataset size - better

- ➔ controllable generation is better for Diffusion-LM

Dataset	Small AR	Small Diffusion	Medium Diffusion
E2E	1.77	2.28	-
ROCStories	3.05	3.88	-
ROCStories (+GPT-J)	2.41	3.59	3.10

Table 7: Log-likelihood results

Controllable Text Generation Results

- ➔ Diffusion-LM outperforms baselines
  - ◆ Non-autoregressive nature = good for future planning (spans, length) & global structures (tree, POS)
  - ◆ coarse-to-fine representations = control on entire sequence ($t \longrightarrow T$) & individual tokens ($t \longrightarrow 0$)

	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

Table 2: 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.

Qualitative Results

- Syntax Tree
  - Diffusion-LM & FT do well (1m + ctrl), FUDGE deviates
  - Diffusion-LM > FT: correct for a failed span - no errors in suffix spans

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.

Table 3: Qualitative examples from the Syntax Tree control. The syntactic parse tree is linearized by nested brackets representing the constituents, and we use the standard PTB syntactic categories. Tokens within each span are represented as * . We color failing spans red and **bold** the spans of interest that we discuss in §7.1.

Composition of Controls

- ➔ Plug-and-play controllable generation = modular
  - ◆ generate from the intersection of multiple controls
  - ◆ Composition results: Diffusion-LM > FUDGE & FT

	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**

Table 4: In this experiment, we compose semantic control and syntactic control: Diffusion-LM achieves higher success rate (ctrl ↑) at some cost of fluency (lm ↓). Our method outperforms both FUDGE and FT-PoE (product of experts of two fine-tuned models) on control success rate, especially for the structured syntactic controls (i.e. syntactic parse tree and POS).

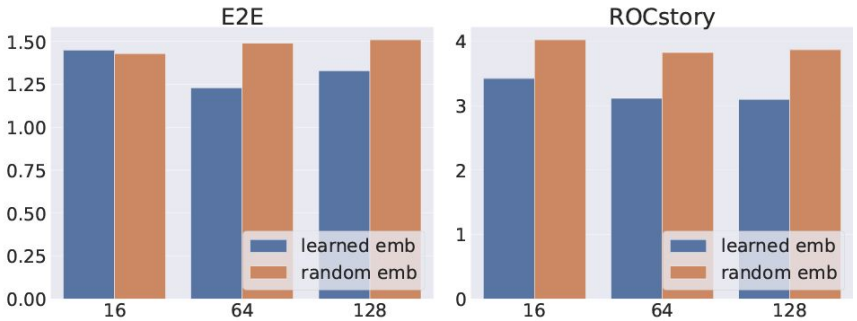
Infilling Results

- ➔ Diffusion-LM outperforms baselines (COLD & DELOREAN)
  - ◆ comparable to AR-infilling
  - ◆ ↑ automatic evaluation, ~ human evaluation

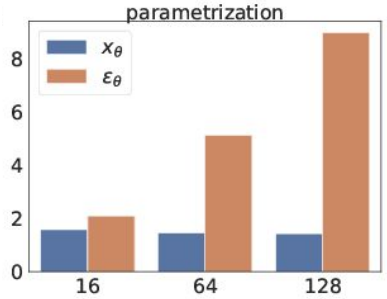
	Automatic Eval				Human Eval
	BLEU-4 ↑	ROUGE-L ↑	CIDEr ↑	BERTScore ↑	
Left-only	0.9	16.3	3.5	38.5	n/a
DELOREAN	1.6	19.1	7.9	41.7	n/a
COLD	1.8	19.5	10.7	42.7	n/a
Diffusion	**7.1**	**28.3**	**30.7**	**89.0**	$\mathbf{0.37}^{+0.03}_{-0.02}$
AR	6.7	27.0	26.9	**89.0**	$\mathbf{0.39}^{+0.02}_{-0.03}$

Table 5: For sentence infilling, Diffusion-LM significantly outperforms prior work COLD [31] and Delorean [30] (numbers taken from paper), and matches the performance of an autoregressive LM (AR) trained from scratch to do infilling.

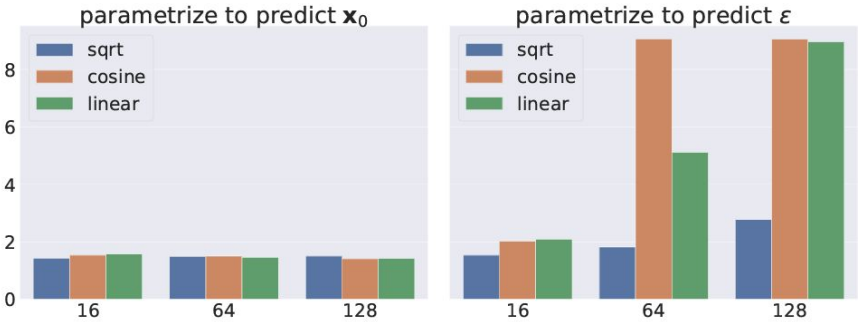
Ablation Studies



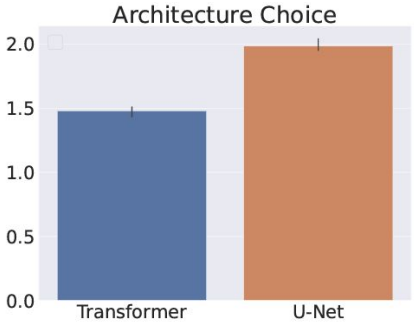
Learned v.s. Random Embeddings



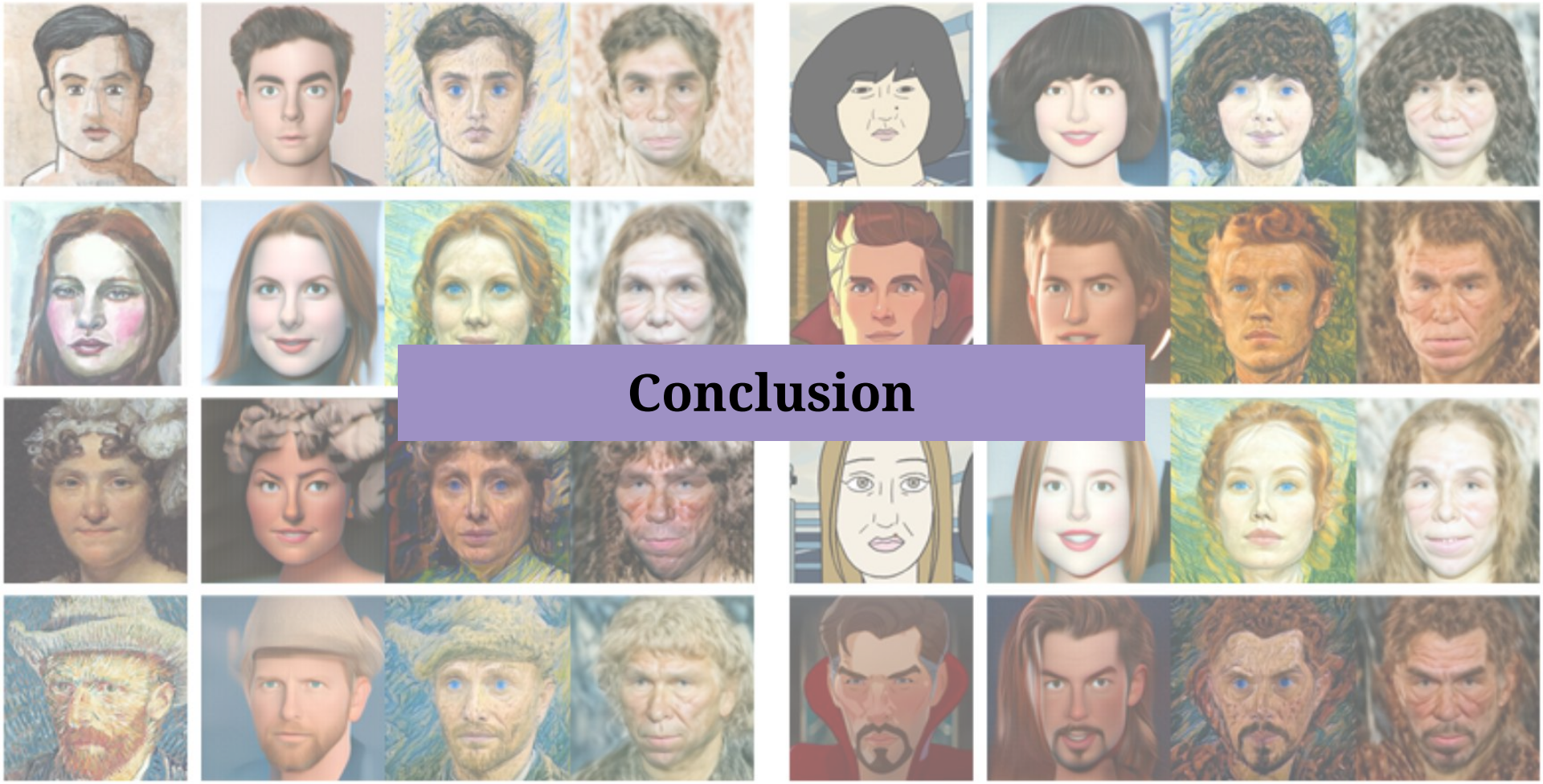
Objective Parametrization (x_0 vs noise term ϵ)



Noise schedule



Architecture



Conclusion

Conclusion

- ➔ Diffusion-LM
  - novel & controllable LM
  - continuous diffusion + non-autoregressive
  - complex fine-grained control tasks
- ➔ Success in 6 control tasks
  - doubles success rate of baselines
  - comparable to dedicated fine-tuning methods
- ➔ Limitations
  - higher perplexity
  - decoding is substantially slower (7x slower than autoregressive LMs)
  - training converges more slowly

Questions