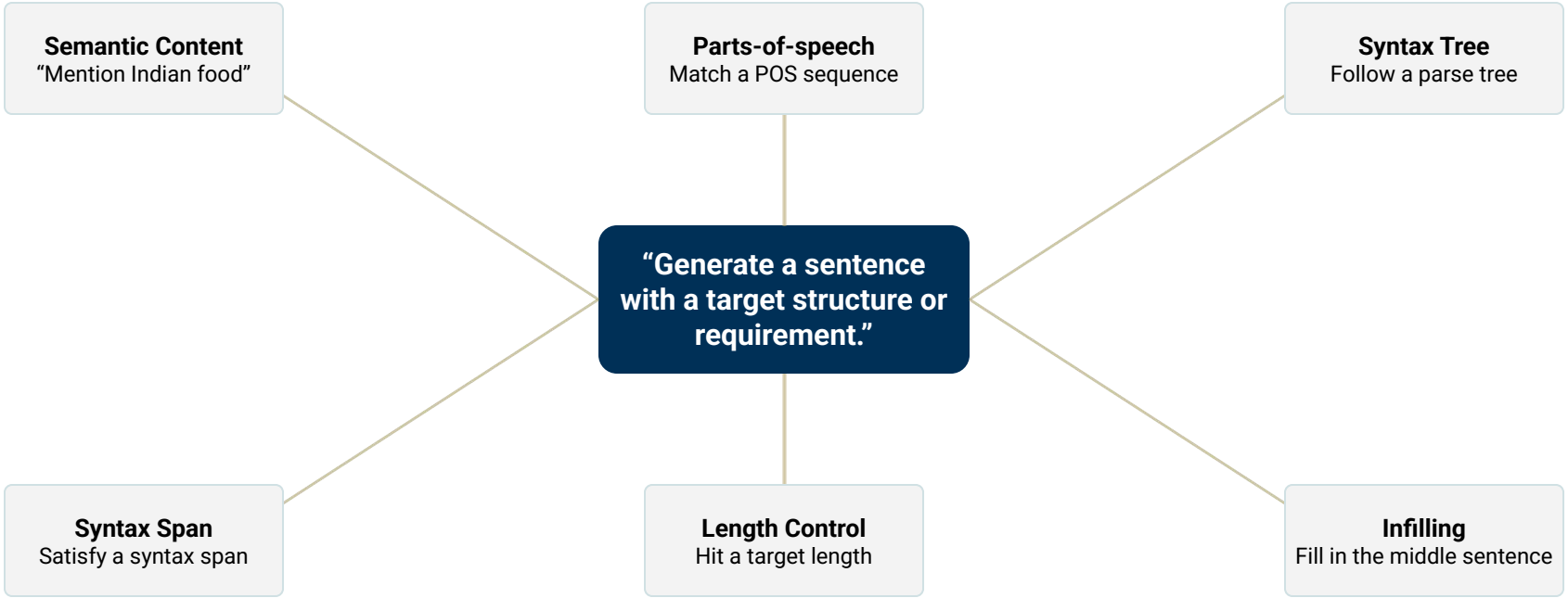
Diffusion-LM: Improves Controllable Text Generation

Discrete Diffusion for Fine-Grained Language Control

Akshay Raj & How-Shing Wang
CS 8803: LLM | April 8, 2026

NeurIPS 2022
Authors: Li et. al

Controllable Text Generation



One model, many kinds of control

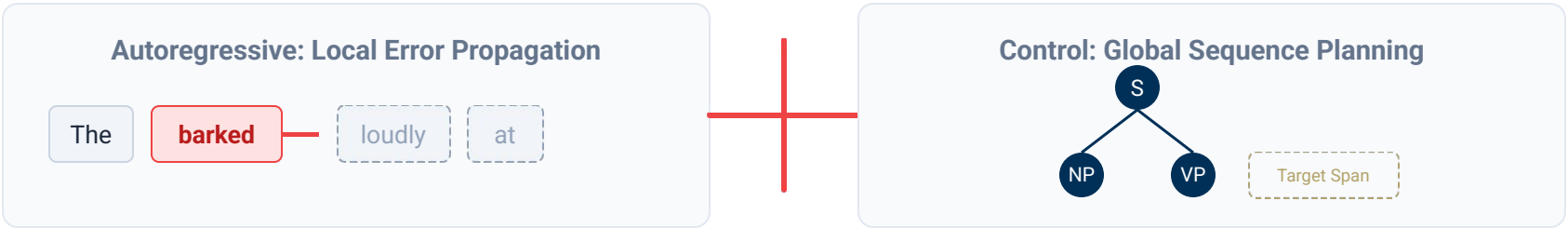
Why are Standard LMs Hard to Control?

Autoregressive LM

- Left-to-right processing
- Local next-token decisions
- Past mistakes are hard to fix

Complex Control Needs

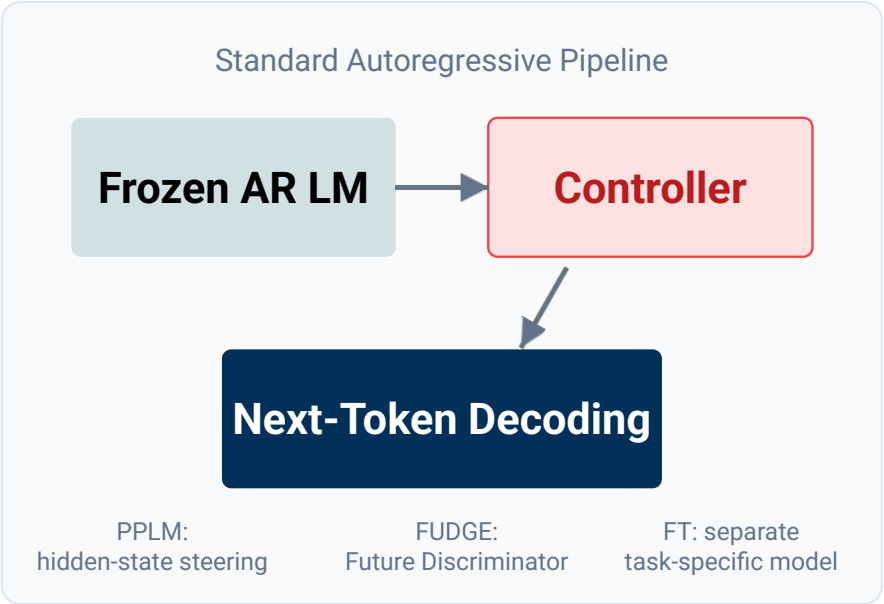
- Global structure (Parse trees)
- Future planning
- Right-context awareness



Local generation is a poor fit for global constraints.

Controlling Frozen Autoregressive LMs

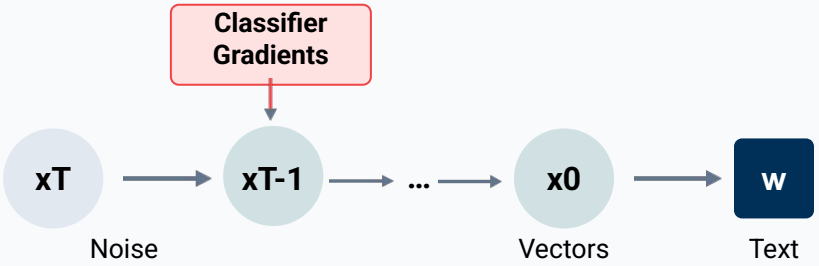
- **PPLM:** Gradient updates on hidden activations to steer next tokens
- **FUDGE:** Future discriminator reweights token predictions during decoding
- **Fine-tuning Oracle:** Retrain or update parameters for each specific control task



Good for simple attributes, weak on structured/global constraints.

Generate Text Through Continuous Diffusion

- Start from Gaussian noise (xT)
- Gradually denoise into word vectors (x0)
- Round vectors to words (w)
- Steer the latent sequence with classifier gradients



Classifier applies gradients on intermediate latents
Final rounded sentence at right

Key Idea: Control the entire sequence, not just the next token

Formalizing Controllable Generation

Goal: Generate text w satisfying control c

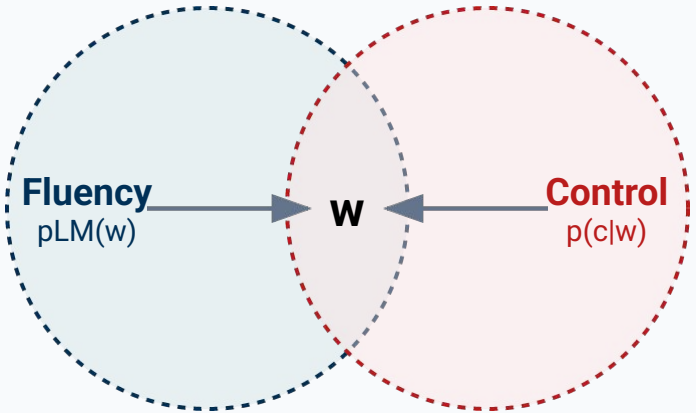
Want: p(w | c)

Plug-and-play view:

$$p(w \mid c) \propto p_{LM}(w)\, p(c \mid w)$$

pLM(w) = fluency

p(c | w) = control satisfaction



Good Output =
Fluent + Controlled

Example Control: "mention Indian food"

✓ "The restaurant serves Indian food in the city complex..." → **Good:** High Fluency + Satisfies Control

✗ "Indian food Indian food Indian food..." → **Bad:** Low Fluency

✗ "The restaurant has a cozy atmosphere and friendly staff..." → **Bad:** No Control Satisfaction

Diffusion Background

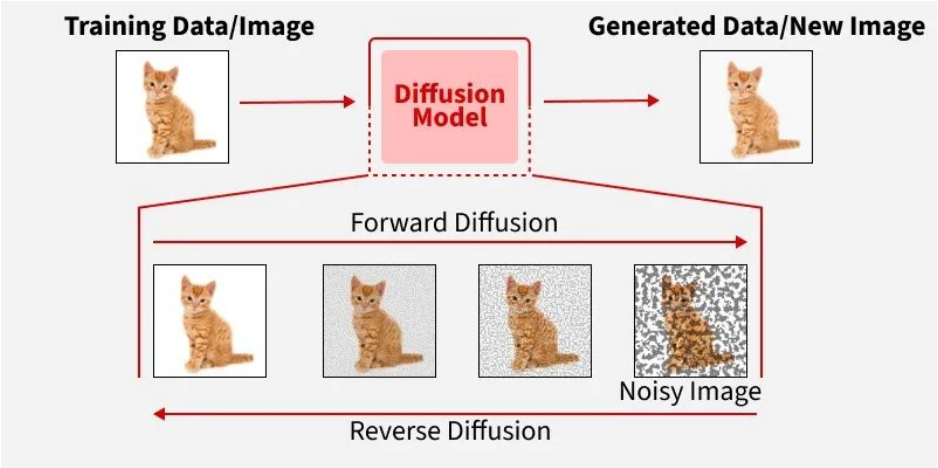
Two stages:

Forward process

- Add Gaussian noise step by step

Reverse process

- Learn to denoise step by step



In this paper: noise → word vectors → text

Why Text Is Harder than Images

Text is discrete

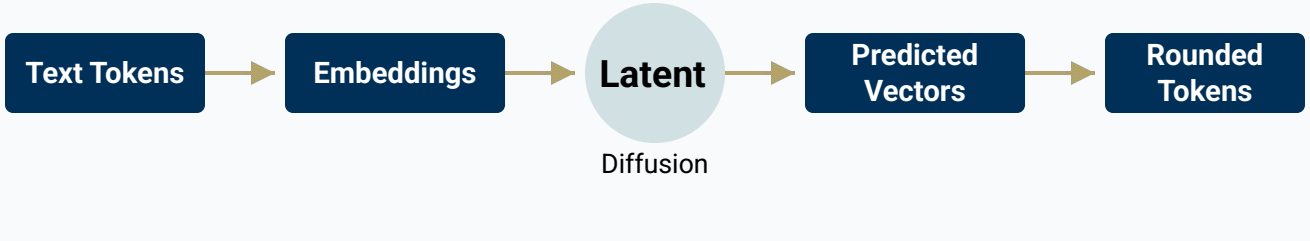
- Cannot directly diffuse tokens

Need embeddings

- Words must be mapped to vectors

Need rounding

- Vectors must be mapped back to tokens



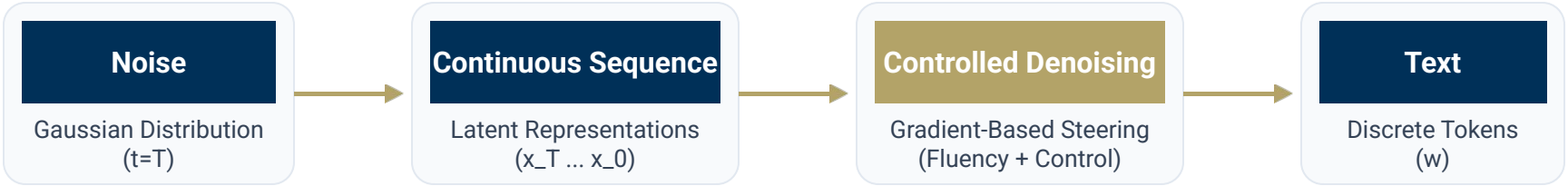
Text needs both an embedding step and a rounding step

Main Challenge: Global Control over Discrete Text

- Constraints depend on entire sequence
- Autoregressive models optimize locally
- Text is discrete → hard to optimize

We need global optimization over the entire sequence

The Paper’s Approach: Diffusion for Controllable Generation



- **Full-sequence representation:** Enables global coordination over the entire output
- **Iterative refinement:** Gradually reduces noise to produce high-quality samples
- **Gradient-based control:** Flexible plug-and-play steering via external classifiers

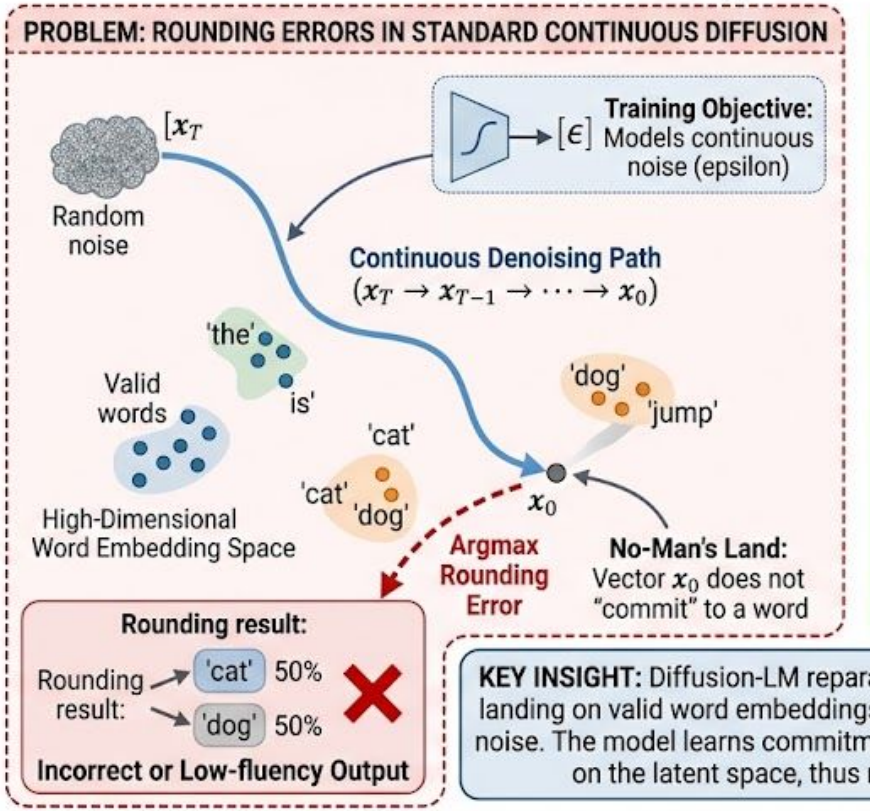
End-to-End Training

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

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

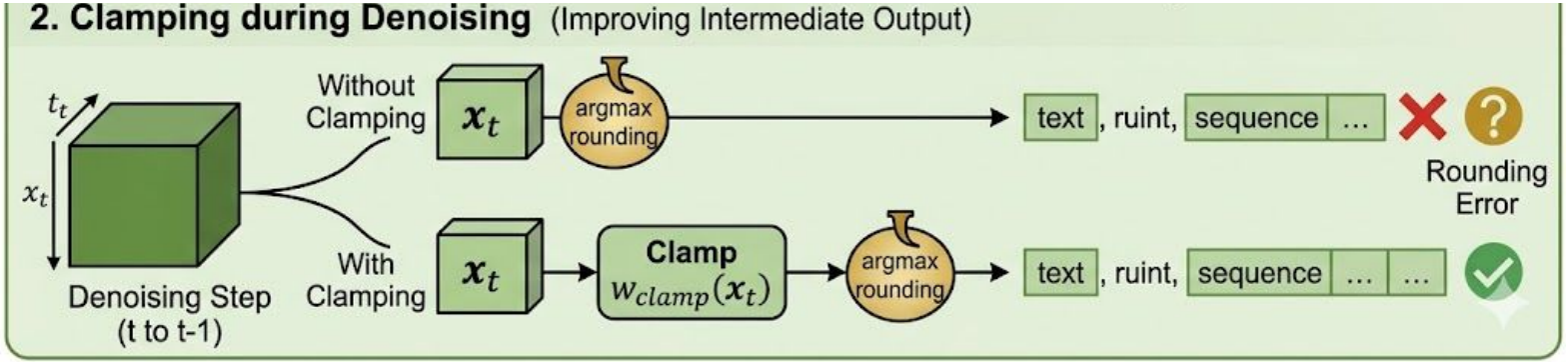
Rounding Errors

- Loss guidance to make x_t stay close to embedding only happens when t is close to 0

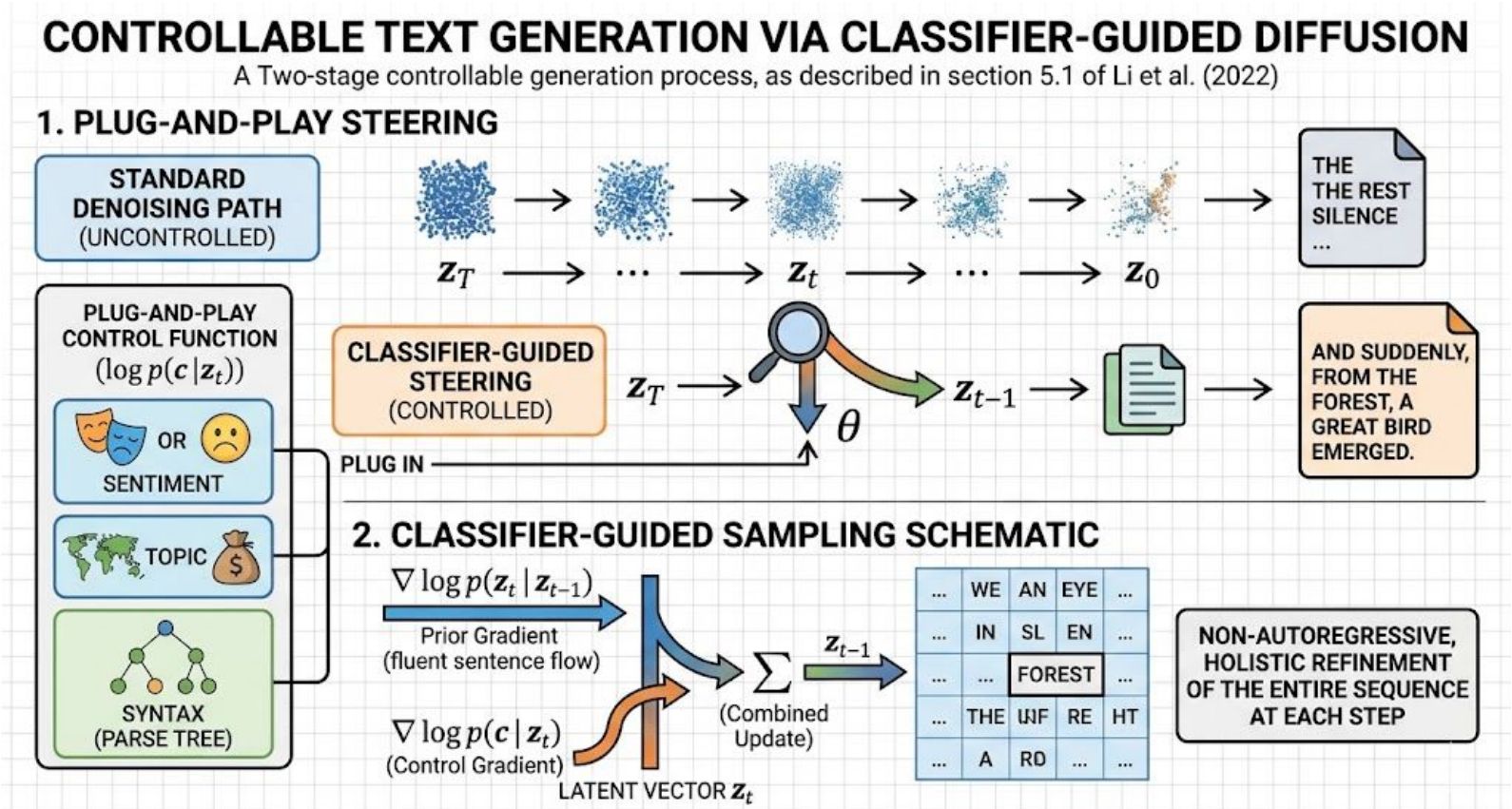


Rounding Errors

- Reparameterize the loss to make “every” step predict x_0 directly during training
- Apply clamping during decoding



Gradient-Based Guidance



6 Control Tasks Evaluated

Three types of control — spanning simple to highly structured:

Classifier-Guided	
Part-of-Speech (POS) *Example: "The [ADJ] [NOUN] quickly [VERB]..."*	Control exact POS tags at each position
Syntactic Parse Tree *Example: (S (NP...) (VP...))*	Control full constituency parse structure
Semantic Content *Example: Must contain: 'economics', 'trade'*	Keywords must appear in output
Sentiment *Example: Output must be positive / negative*	Simpler — baseline can handle too
No Classifier Needed	
Length Control *Example: Generate a sentence of exactly 15 tokens*	Directly encoded; no classifier required
Infilling *Example: The [BLANK] quickly jumped [BLANK]*	Fill in blanks given surrounding context

Classifier-Tasks

	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

Diffusion-LM significantly outperforms prior plug-and-play autoregressive models on complex tasks like syntactic tree matching because it can plan globally across the sequence.

Non-Classifier-Tasks

	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**	$\mathbf{0.39}^{+0.02}_{-0.03}$

For sentence infilling, Diffusion-LM significantly outperforms prior work COLD and Delorean (numbers taken from paper), and matches the performance of an autoregressive LM trained from scratch to do infilling.

Takeaways

1

Continuous Diffusion Unlocks Gradient Control

By working in a continuous embedding space, Diffusion-LM enables gradient-based guidance from arbitrary classifiers — something discrete token generation fundamentally cannot do.

2

Complex, Fine-Grained Control is Now Feasible

For the first time, a plug-and-play method achieves strong results on syntactic parse tree control — a task that was essentially unsolved before this paper.

3

Trade-Off: Control vs. Fluency vs. Speed

Diffusion-LM is not a drop-in replacement for GPT-style LMs. It excels at constrained generation but sacrifices some fluency and is much slower due to iterative denoising.

4

A New Research Direction: Diffusion for NLP

This paper catalysed a wave of follow-up work (DiffuSeq, SSD-LM, GENIE...). The core idea — continuous diffusion + gradient guidance — remains highly active and is being scaled up.