Scaling Diffusion Language Models via Adaptation from Autoregressive Models

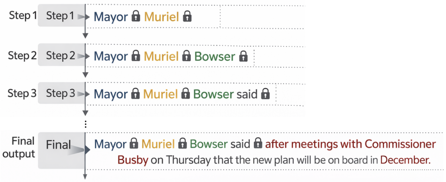
Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, Lingpeng Kong

Pengxi Liu

January 9, 2026

Motivation: Comparison of AR and DLM

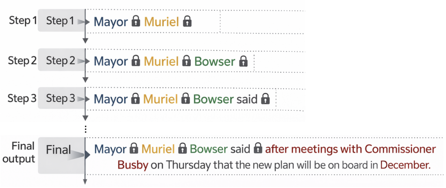
Autoregressive LLM (AR)



+ **Dominant performance** in many language tasks such as generating high-quality text and in-context learning.
+ **Abundant pretrained models** and mature training infrastructure.

Motivation: Comparison of AR and DLM

Autoregressive LLM (AR)

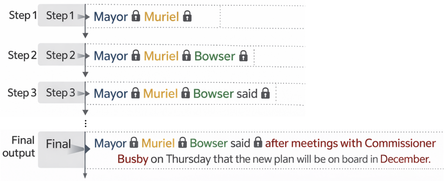


+ **Dominant performance** in many language tasks such as generating high-quality text and in-context learning.
+ **Abundant pretrained models** and mature training infrastructure.
− **Strict left-to-right generation** limits performance in tasks like global planning and self-correction.

$$p(x) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$$

Motivation: Comparison of AR and DLM

Autoregressive LLM (AR)



+ **Dominant performance** in many language tasks such as generating high-quality text and in-context learning.
+ **Abundant pretrained models** and mature training infrastructure.
− **Strict left-to-right generation** limits performance in tasks like global planning and self-correction.

$$p(x) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$$

Diffusion Language Model (DLM)

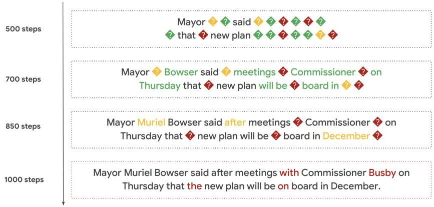
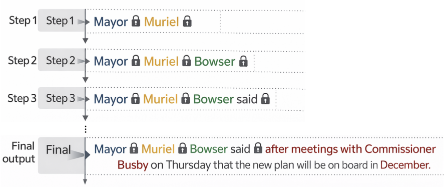


Figure adapted from Shi et al., "Simplified and Generalized Masked Diffusion for Discrete Data".

+ **Controllable and parallel** generation.

Motivation: Comparison of AR and DLM

Autoregressive LLM (AR)



- + **Dominant performance** in many language tasks such as generating high-quality text and in-context learning.
- + **Abundant pretrained models** and mature training infrastructure.
- − **Strict left-to-right generation** limits performance in tasks like global planning and self-correction.

$$p(x) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$$

Diffusion Language Model (DLM)

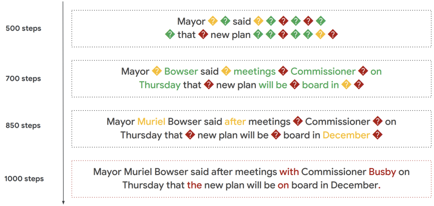
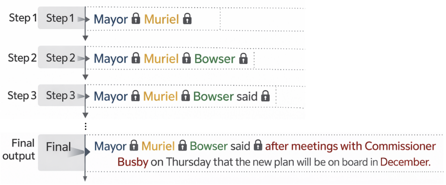


Figure adapted from Shi et al., "Simplified and Generalized Masked Diffusion for Discrete Data".

- + **Controllable and parallel** generation.
- − **Relatively small model size** limits the competitiveness of DLMs compared to AR models.

Motivation: Comparison of AR and DLM

Autoregressive LLM (AR)



- **Dominant performance** in many language tasks such as generating high-quality text and in-context learning.
- **Abundant pretrained models** and mature training infrastructure.
- **Strict left-to-right generation** limits performance in tasks like global planning and self-correction.

$$p(x) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$$

Diffusion Language Model (DLM)

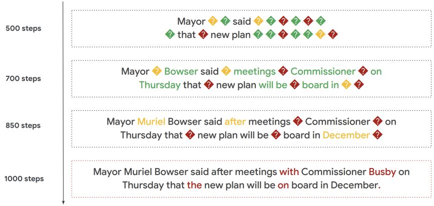


Figure adapted from Shi et al., "Simplified and Generalized Masked Diffusion for Discrete Data".

- **Controllable and parallel** generation.
- **Relatively small model size** limits the competitiveness of DLMs compared to AR models.

Solution: Adapt large pretrained AR LLMs to diffusion, combining the **scalability** of AR models with the **flexibility** of diffusion.

Motivation: Challenges of Adaptation

- **AR:** Trained with causal masking on clean token sequences, predicting the next token at each step.
- **DLM:** Trained with bidirectional context on noisy sequences, predicting denoised tokens at arbitrary positions.

Motivation: Challenges of Adaptation

- **AR:** Trained with causal masking on clean token sequences, predicting the next token at each step.
- **DLM:** Trained with bidirectional context on noisy sequences, predicting denoised tokens at arbitrary positions.

Key challenge: Mismatched training objectives and input distributions.

Preliminaries: Continuous Diffusion Models

- **Clean data:** $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0)$
- **Terminal noise:** $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
- **Noisy state at time** t**:** $\mathbf{x}_t \sim q(\mathbf{x}_t)$

Forward Process:

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \quad q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$$

which gradually corrupts clean data $\mathbf{x}_0$ into increasingly noisy variables.

Reverse Process:

$$p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t),$$

which learns to iteratively denoise $\mathbf{x}_t$ to reconstruct $\mathbf{x}_0$.

Training Objective (ELBO).

$$-\log p_\theta(\mathbf{x}_0) \ \leq \ \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)] + D_{\mathrm{KL}}(q(\mathbf{x}_T \mid \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_T)) + \mathcal{L}_T,$$

where

$$\mathcal{L}_T = \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[D_{\mathrm{KL}}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t))],$$

which enforces consistency between the true and learned reverse transitions at intermediate timesteps.

Preliminaries: Discrete Diffusion Models

In discrete denoising models, each token is a one-hot vector:

$$\mathbf{x}_t \in \{0,1\}^K$$

Forward Process:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathrm{Cat}(\mathbf{x}_t; \mathbf{Q}_t^\top \mathbf{x}_{t-1})$$

Absorbing diffusion transition:

$$\mathbf{Q}_t = (1-\beta_t)\mathbf{I} + \beta_t \mathbf{1}\mathbf{m}^\top$$

- With probability $1-\beta_t$: token unchanged
- With probability β_t: token $\rightarrow$ `[MASK]`

The marginal distribution at time t has a closed form:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \alpha_t \mathbf{x}_0 + (1-\alpha_t)\mathbf{m}, \quad \alpha_t = \prod_{i=1}^{t}(1-\beta_i).$$

Model: Continuous-time Discrete Diffusion Process

Consider dividing $[0, 1]$ into T intervals ($T \to \infty$). For $0 \leq s < t \leq 1$:

Forward Process

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_s) = \frac{\alpha_t}{\alpha_s}\boldsymbol{x}_s + \left(1 - \frac{\alpha_t}{\alpha_s}\right)\boldsymbol{m}.$$

Backward Process

$$q(\boldsymbol{x}_s \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \begin{cases} \dfrac{\alpha_s - \alpha_t}{1 - \alpha_t}\boldsymbol{x}_0 + \dfrac{1 - \alpha_s}{1 - \alpha_t}\boldsymbol{m}, & \text{if } \boldsymbol{x}_t = \boldsymbol{m}, \\ \boldsymbol{x}_0, & \text{if } \boldsymbol{x}_t \neq \boldsymbol{m}. \end{cases}$$

Approximate the true backward transition using a denoising model:

$$p_\theta(\boldsymbol{x}_s \mid \boldsymbol{x}_t, f_\theta(\boldsymbol{x}_t)),$$

where $f_\theta(\boldsymbol{x}_t)$ approximates $\boldsymbol{x}_0$.

Define a similar form of backward transition:

$$p_\theta(\boldsymbol{x}_s \mid \boldsymbol{x}_t) = \frac{\alpha_s - \alpha_t}{1 - \alpha_t} f_\theta(\boldsymbol{x}_t) + \frac{1 - \alpha_s}{1 - \alpha_t}\boldsymbol{m}.$$

$f_\theta(\cdot)$ is implemented by a neural network (e.g., a Transformer).

Model: Continuous-time Discrete Diffusion Process

At each timestep, the KL term simplifies to:

$$D_{\mathrm{KL}}(q(\boldsymbol{x}_s \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_s \mid \boldsymbol{x}_t)) = -\frac{\alpha_s - \alpha_t}{1 - \alpha_t} \delta_{\boldsymbol{x}_t, \boldsymbol{m}} \boldsymbol{x}_0^\top \log f_\theta(\boldsymbol{x}_t).$$

As $T \to \infty$, the continuous-time loss becomes:

$$\lim_{T \to \infty} \mathcal{L}_T = \int_0^1 \frac{\alpha_t'}{1 - \alpha_t} \mathbb{E}_{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)} \left[\delta_{\boldsymbol{x}_t, \boldsymbol{m}} \boldsymbol{x}_0^\top \log f_\theta(\boldsymbol{x}_t) \right] dt.$$

Following prior work, choose $\alpha_t = 1 - t$. This gives a simple weight $\frac{-\alpha_t'}{1-\alpha_t} = \frac{1}{t}$. The formulation extends independently to a sequence of N tokens:

$$\mathbf{x}_t = [\boldsymbol{x}_t^1, \ldots, \boldsymbol{x}_t^N].$$

During training, sample $t \sim \mathcal{U}(0, 1)$ and optimize:

$$\mathcal{L}_t^{1:N} = \frac{1}{t} \mathbb{E}_{q(\mathbf{x}_t \mid \mathbf{x}_0)} \left[-\sum_{n=1}^{N} \delta_{\boldsymbol{x}_t^n, \boldsymbol{m}} (\boldsymbol{x}_0^n)^\top \log f_\theta(\mathbf{x}_t^{1:N})_n \right].$$

Model: Unifying Language Modeling Objectives

AR LLM:

$$\mathcal{L}_{\text{AR}}^{1:N} = -\sum_{n=1}^{N} (\mathbf{x}_0^n)^\top \log f_\theta\left(\mathbf{x}_0^{1:n-1}\right)_{n-1}$$

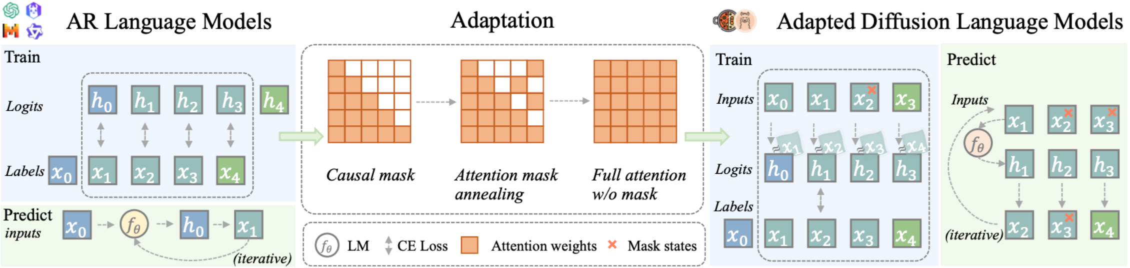
DLM:

$$\mathcal{L}_t^{1:N} = \frac{1}{t}\,\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[-\sum_{n=1}^{N} \delta_{\mathbf{x}_t^n,\boldsymbol{m}}\,(\mathbf{x}_0^n)^\top \log f_\theta\left(\mathbf{x}_t^{1:N}\right)_n\right]$$

Key differences:

- Reweighting $\frac{1}{t}$: importance over noise levels.
- Mask indicator: predicts only corrupted tokens.
- Noisy, bidirectional context vs. clean, causal attention.

Model: Adaptation



- **Attention Mask Annealing:** causal → bidirectional attention.
- **Shift Operation:** preserve AR target shifting to align diffusion denoising.
- **Time-Embedding-Free Architecture:** reuse AR architecture; noise level encoded implicitly.

Experiment: Adaptation Setup

Models

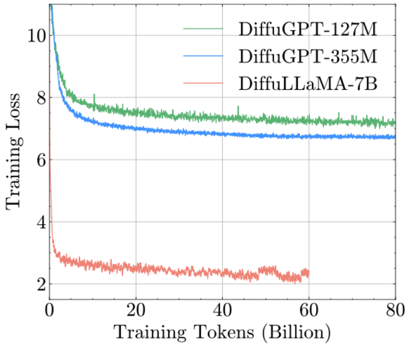
- **DiffuGPT:** GPT-2 base $\rightarrow$ diffusion
- **DiffuLLaMA:** LLaMA-2-7B $\rightarrow$ diffusion

Data

- DiffuGPT: 30B tokens from FineWeb
- DiffuLLaMA: 60B tokens from SlimPajama + Starcoder

Training

- Full-parameter finetuning (fp16)
- Sequence packing of length 2048 + logits shifting
- Attention mask annealing for GPT-2 and bi-directional for LLaMA



Training loss vs. total tokens for different model sizes.

Experiment: Evaluation Setup

Tasks:

- Reading comprehension and long-range dependency
- Commonsense and math reasoning
- Text and code infilling

Category	**Datasets**	**Metric**
QA / Completion	TriviaQA, LAMBADA	Exact Match (Accuracy)
Common Sense	HellaSwag, Wino, SIQA, PIQA	Accuracy
Math	GSM8K	Accuracy
Infilling	ROCStories	ROUGE-1/2/L
Code Infilling	HumanEval	pass@1

Experiment: Evaluation Setup (Metrics I: Exact Match / Accuracy)

Datasets: TriviaQA (reading comprehension), LAMBADA (last word prediction), GSM8K (math reasoning)

Definition:

$$\mathrm{EM} = \mathbf{1}\left[\mathrm{normalize}(\hat{y}) = \mathrm{normalize}(y^{\star})\right]$$

- String-level exact match after normalization (case, punctuation, articles).
- Dataset score = average EM over all samples.

Key details:

- QA and math tasks have well-defined final answers.
- Avoids ambiguity from paraphrases.

For GSM8K, only the final numeric answer is evaluated (not intermediate reasoning).

Experiment: Evaluation Setup (Metrics II: Multiple-Choice Reasoning)

Datasets: HellaSwag, WinoGrande, SIQA, PIQA

Evaluation Protocol:

- Each question has a prompt and several candidate choices.
- For each choice c, compute the **token-averaged negative log-likelihood**:

$$\text{Score}(c) = \frac{1}{|c|} \sum_{i \in c} - \log p(x_i \mid \text{prompt})$$

- Select the choice with the lowest score.
- Metric = accuracy (fraction of correct choices).

Key details:

- Length normalization avoids bias toward shorter options.
- Same scoring applies to AR and diffusion models.

Experiment: Evaluation Setup (Metrics III: Infilling and Code Evaluation)

Text Infilling: ROCStories

- Task: fill missing spans in short stories.
- Metric: ROUGE-1 / ROUGE-2 / ROUGE-L.

- ROUGE-1: unigram overlap
- ROUGE-2: bigram overlap
- ROUGE-L: longest common subsequence

Code Infilling: HumanEval

- Task: fill missing code so that unit tests pass.
- Metric: pass@1.

$$\text{pass@1} = \frac{\#\{\text{programs passing tests}\}}{\#\{\text{problems}\}}$$

Experiment: Results (Benchmark performance)

Model	Size	Type	QA	Word	CommonSense Reasoning				Math	Infilling	
			TriQA	Lamb.	HSwag	Wino.	SIQA	PIQA	GSM8K*	ROCStories	Code
GPT2-S	127M	AR	4.0	25.9	29.9	48.5	35.7	62.1	44.8	(7.8/0.8/7.4)	(1.6)
SEDD-S	170M	DD	1.5	12.4	30.2	50.1	34.4	55.6	45.3	11.9/0.7/10.9	0.7
DiffuGPT-S	127M	DD	2.0	21.6	<u>33.4</u>	<u>50.8</u>	<u>37.0</u>	57.7	<u>50.2</u>	<u>13.7/1.4/12.6</u>	0.3
GPT2-M	355M	AR	6.7	37.7	38.3	50.7	37.7	67.4	45.6	(8.6/0.9/8.2)	(2.6)
SEDD-M	424M	DD	1.8	23.1	31.5	49.0	35.4	56.1	53.5	13.1/1.4/12.2	0.5
DiffuGPT-M	355M	DD	3.8	30.3	37.2	<u>52.6</u>	<u>39.0</u>	59.6	<u>61.8</u>	<u>18.7/2.7/17.0</u>	<u>2.9</u>
Plaid1B	1.3B	CD	1.2	8.6	39.3	51.3	32.3	54.5	32.6	12.1/1.1/11.2	0.1
LLaMA2	7B	AR	45.4	68.8	74.9	67.1	44.8	78.3	58.6	(11.6/2.1/10.5)	(1.7)
DiffuLLaMA	7B	DD	**18.5**	**53.9**	**58.7**	**56.4**	**43.2**	**63.3**	<u>**63.1**</u>	<u>**23.3/5.5/21.2**</u>	<u>**15.5**</u>

Note: SEDD and Plaid are prior state-of-the-art DLMs. Bold denotes the best DLM result; underlined values outperform the corresponding AR base model.

+ DiffuGPT and DiffuLLaMA consistently outperform **prior SOTA DLMs**.
+ DiffuGPT and DiffuLLaMA excel at **global reasoning** like math, code, and infilling tasks.
− DiffuLLaMA still lags behind LLaMA2 on some benchmarks. The paper believes this gap is primarily data-limited

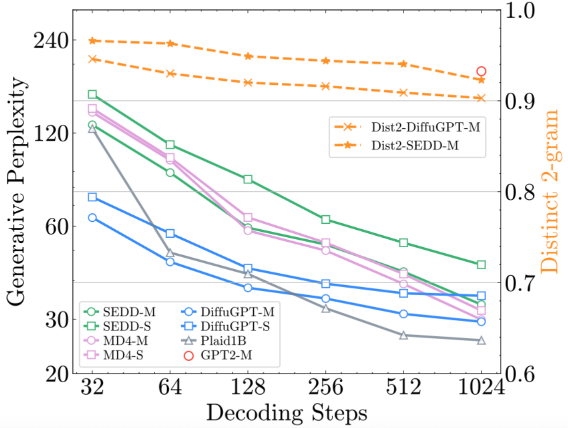
Experiment: Results (Unconditional Generation)

- **Quality: Perplexity (PPL).** Given a generated sequence $x_{1:N}$, an external evaluator (GPT-2 Large) assigns token probabilities $p(x_{1:N}) = \prod_{i=1}^{N} p(x_i \mid x_{<i})$. Then

$$\mathrm{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log p(x_i \mid x_{<i})\right)$$

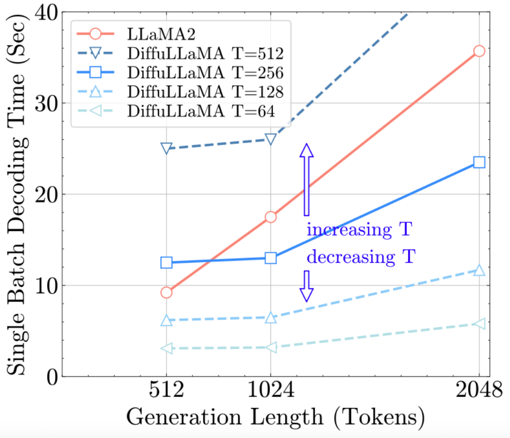
- **Diversity: Distinct-2** measures the fraction of **unique 2-grams** (bigrams) in the generated text.

$$\text{Distinct-2} = \frac{\#\{\text{unique 2-grams}\}}{\#\{\text{total 2-grams}\}}$$



Experiment: Results (Inference Speed)

- Decoding time mainly scales with the number of **diffusion steps** T.
- Diffusion inference offers an explicit knob (T) to control latency and quality.



Accommodation

Is it worth reading? Yes!

- Provides a clear comparison between DLMs and AR LLM objectives.
- Serves as a useful reference for model formulations and evaluation metrics in diffusion-based text generation.

Is it worth implementing? Yes!

- Proposes a simple and practical way to align AR and diffusion objectives via engineering tricks.
- Demonstrates promising empirical results, with room for further scaling and optimization.