Large Language Diffusion Models

Pu Yang
School of Mathematical Sciences, Peking University

2025.4.2

Table of contents

1. Introduction
2. Related Work: Diffusion Models in NLP
   - Diffusion-LM
   - Discrete Diffusion Model
3. Large Language Diffusion Models
4. Discussion

Introduction: Autoregressive Models (ARMs)

- Autoregressive models (ARMs) are widely regarded as the cornerstone of large language models (LLMs).
- ARMs commonly referred to as the *next-token prediction* paradigm:

$$\underbrace{p_\theta(x) = p_\theta\left(x^1\right) \prod_{i=2}^{L} p_\theta\left(x^i \mid x^1, \ldots, x^{i-1}\right)}_{\text{Autoregressive formulation}},$$

where x is a sequence of length L, and x^i is the i-th token

Limitations of ARMs

- **Computational efficiency**
  - ARMs cannot be generated in parallel for inference.
  - ARMs exhibits a quadratic computational complexity of order $O(n^2)$ due to its sequential dependence structure.

Limitations of ARMs

- **Unidirectional nature**
  - ARMs are not free to choose the location of outputs.
  - ARMs cannot modify what has already been output.

  v.s. Turing machine
  - Write or erase a symbol on a piece of paper;
  - Move the attention from one place on the paper to another.

Diffusion Models in LLMs

- Diffusion models in LLMs aim to capture the true data distribution through MLE:

$$\underbrace{\max_{\theta} \mathbb{E}_{p_{\text{data}}(x)} \log p_{\theta}(x) \Leftrightarrow \min_{\theta} \mathrm{KL}\left(p_{\text{data}}(x) \| p_{\theta}(x)\right)}_{\text{Generative modeling principles}}.$$

- Trivial idea: from noise/[MASK] to token.
- What's the benefits?
  - generate in parallel for inference
  - modify any token
  - better computational complexity (?)
  - ...

1 Introduction

2 Related Work: Diffusion Models in NLP

- Diffusion-LM
- Discrete Diffusion Model

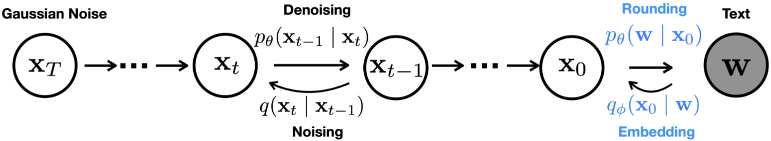
3 Large Language Diffusion Models

4 Discussion

Diffusion-LM: Continuous Diffusion Language Modeling

Diffusion-LM[1] defines the diffusion process in a **continuous** word vector space, specifically:

- Discrete to continuous: Map each word to a continuous vector
- Diffusion: Add continuous noise to these vectors and iteratively denoise them
- Continuous to discrete: Map continuous vectors to words


Gaussian Noise
$\mathbf{x}_T$
$\mathbf{x}_t$
Denoising
$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$
$q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$
Noising
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
Rounding
$p_\theta(\mathbf{w} \mid \mathbf{x}_0)$
$q_\phi(\mathbf{x}_0 \mid \mathbf{w})$
Embedding
Text
$\mathbf{w}$

[1] Xiang Li et al. "Diffusion-LM Improves Controllable Text Generation". In: Advances in Neural Information Processing Systems. Vol. 35. Curran Associates, Inc., 2022, pp. 4328–4343.

Controllable Text Generation

Controlling $x_{0:T}$ is equivalent to decoding from the posterior:

$$\begin{aligned} p(x_{0:T} \mid c) &= \prod_{t=1}^{T} p(x_{t-1} \mid x_t, c) \\ &\propto \prod_{t=1}^{T} p(x_{t-1} \mid x_t) \cdot p(c \mid x_{t-1}, x_t) \\ &= \prod_{t=1}^{T} p(x_{t-1} \mid x_t) \cdot p(c \mid x_{t-1}) \end{aligned}$$

Similar to conditional generation in diffusion models.[2]

[2] Yang Song et al. “Score-based generative modeling through stochastic differential equations”. In: arXiv preprint arXiv:2011.13456 (2020).

D3PM: Discrete Denoising Diffusion Probabilistic Model[5]

- For scalar discrete random variables with K categories, the forward transition probabilities can be represented by matrices: $[\mathbf{Q}_t]_{ij} = q(x_t = j \mid x_{t-1} = i)$.
  - Then we get the forward process: $q(x_t \mid x_{t-1})$.
  - Then we get the reverse process: $q(x_{t-1} \mid x_t, x_0)$.
  - Then we get the loss function ...
- Choice of Markov transition matrices for the forward process, e.g.,
  - Uniform: $\mathbf{Q}_t = (1 - \beta_t)\mathbf{I} + \frac{\beta_t}{K}\mathbb{1}\mathbb{1}^T$
  - **Absorbing state: fully observed text to a sequence of [MASK]**[3][4]
  - ...
- Connection to existing probabilistic models for text
  - BERT: one-step discrete diffusion
  - ARMs: autoregressive discrete diffusion

[3] Aaron Lou and Stefano Ermon. “Reflected diffusion models”. In: International Conference on Machine Learning. PMLR. 2023, pp. 22675–22701.

[4] Jingyang Ou et al. “Your absorbing discrete diffusion secretly models the conditional distributions of clean data”. In: arXiv preprint arXiv:2406.03736 (2024).

[5] Jacob Austin et al. “Structured denoising diffusion models in discrete state-spaces”. In: Advances in neural information processing systems 34 (2021), pp. 17981–17993.

Masked Diffusion Models (MDMs) and Its Forward Process

- Define the model distribution as $p_\theta(x_0)$
- Introduce a **forward process** $\{x_t\}$ indexed by a time $t \in [0, 1]$.
  - The data point x_0 is fully observed with no masks.
  - The conditional distribution of x_t given x_0 is

    $$q_{t|0}\left(x_t \mid x_0\right) = \prod_{i=1}^{L} q_{t|0}\left(x_t^i \mid x_0^i\right)$$

    where the conditional distribution for each token is given by:

    $$q_{t|0}\left(x_t^i \mid x_0^i\right) = \begin{cases} 1-t, & x_t^i = x_0^i \\ t, & x_t^i = \text{[MASK]} \end{cases}$$

    Intuitively, each token either remains unchanged or is masked, with the probability of being masked increasing linearly as t.
  - At $t = 1$, all tokens are masked, meaning that x_1 is a sequence of fully masked tokens

Reverse Process

- The **reverse process**, from time $t = 1$ to 0, generates new data from sequences of fully masked tokens.
  - The conditional distribution for the reverse process, for $0 \leq s < t \leq 1$, is

$$q_{s|t}\left(x_s \mid x_t\right) = \prod_{i=1}^{L} q_{s|t}\left(x_s^i \mid x_t\right)$$

where the conditional distribution for each token is:

$$q_{s|t}\left(x_s^i \mid x_t\right) = \begin{cases} 1, & x_t^i \neq \mathbf{M}, x_s^i = x_t^i \\ \frac{s}{t}, & x_t^i = \mathbf{M}, x_s^i = \mathbf{M} \\ \frac{t-s}{t} q_{0|t}\left(x_s^i \mid x_t\right), & x_t^i = \mathbf{M}, x_s^i \neq \mathbf{M} \\ 0, & \text{otherwise} \end{cases}$$

Similar to the data prediction form in continous diffusion models, the key function to estimate is $q_{0|t}\left(x_s^i \mid x_t\right)$, which predicts the original token if it is masked in the input x_t.

Parameterization

- An equivalent yet **time-free** parameterization can be derived as

$$q_{0|t}\left(x_s^i \mid x_t\right) = p_{\text{data}}(x_0 \mid x_t^{\text{UM}})$$

  where x_t^{UM} denotes the collection of unmasked tokens in x_t.
- We introduce the **mask predictor**, a parametric model $p_\theta(\cdot \mid x_t)$, which takes x_t for any t as input and predict all masked tokens simultaneously.

Optimization

- The mask predictor is trained using a cross-entropy loss computed only on the masked tokens:

$$\mathcal{L}(\theta) \triangleq -\mathbb{E}_{t,x_0,x_t}\left[\frac{1}{t}\sum_{i=1}^{L}\mathbf{1}\left[x_t^i = \text{[MASK]}\right]\log p_\theta\left(x_0^i \mid x_t\right)\right]$$

  where x_0 is sampled from the training data, t is sampled uniformly from $[0,1]$, and x_t is sampled from the forward process. The indicator function $\mathbf{1}[\cdot]$ ensures that the loss is computed only for masked tokens.

- Notice that,

$$-\mathbb{E}_{p_{\text{data}}(x_0)}\left[\log p_\theta\left(x_0\right)\right] \leq \mathcal{L}(\theta)$$

1 Introduction

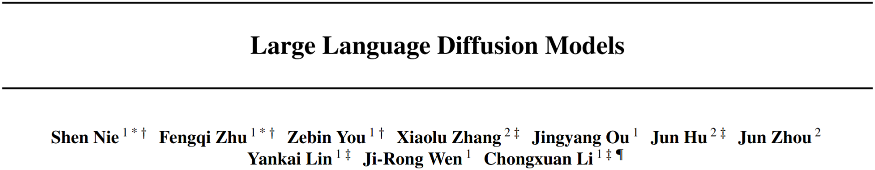
2 Related Work: Diffusion Models in NLP

- Diffusion-LM
- Discrete Diffusion Model

3 Large Language Diffusion Models

4 Discussion

LLaDA: **L**arge **L**anguage **D**iffusion with m**A**sking[6]



Large Language Diffusion Models

Shen Nie[1,*,†] **Fengqi Zhu**[1,*,†] **Zebin You**[1,†] **Xiaolu Zhang**[2,‡] **Jingyang Ou**[1] **Jun Hu**[2,‡] **Jun Zhou**[2]
Yankai Lin[1,‡] **Ji-Rong Wen**[1] **Chongxuan Li**[1,‡,¶]

1 Gaoling School
2 Ant Group

[6] Shen Nie et al. “Large Language Diffusion Models”. In: arXiv preprint arXiv:2502.09992 (2025).

Conceptual Overview

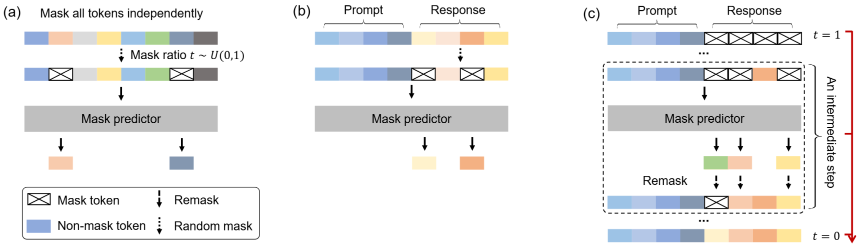


Figure 2. **A Conceptual Overview of LLaDA.** (a) Pre-training. LLaDA is trained on text with random masks applied independently to all tokens at the same ratio $t \sim U[0, 1]$. (b) SFT. Only response tokens are possibly masked. (c) Sampling. LLaDA simulates a diffusion process from $t = 1$ (fully masked) to $t = 0$ (unmasked), predicting all masks simultaneously at each step with flexible remask strategies.

Architecture and Pre-training

- LLaDA uses the Transformer without a causal mask
  - incompatible with KV caching
- Some training hyper-parameters:
  - Dataset: 2.3T tokens
  - Sequence length: 4096
  - Computational cost: 0.13M H800 GPU hours for LLaDA-8B (similar to ARMs (LLaMA3-8B) of the same scale and dataset size.)
- For a training sequence x_0, randomly sample $t \in [0, 1]$, mask each token independently with the same probability t to obtain x_t, and estimate the loss via Monte Carlo method for diffusion training.

Supervised Fine-Tuning

- Instruction tuning with paired data (p_0, r_0), where p_0 is the prompt and r_0 denotes the response with an |EOS| token at the end.
- Technically, is requires to model the conditional distribution $p_\theta(r_0 \mid p_0)$ instead of $p_\theta(x_0)$ in pre-training.

Inference Trick: Remask

At an intermediate step from time $t \in (0, 1]$ to $s \in [0, t)$, **remask** $\frac{s}{t}$ of the predicted tokens in expectation to obtain the remasked response.

- Low-confidence remasking: remask the $\frac{s}{t}$ of predicted tokens with the lowest confidence based on the predictions
- Semi-autoregressive remasking: ivide the sequence into several blocks and generate them from left to right

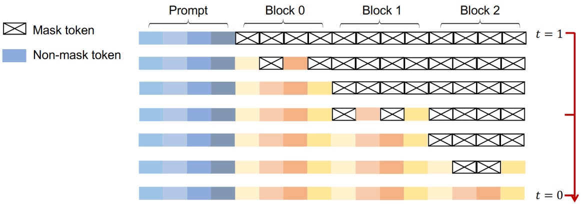


Figure 4. **A Conceptual Overview of the Semi-autoregressive Sampling.**

Experiments: Scalability

- Four models:
  - ARM baselines: 1.5B and 7B
  - MDMs: 1.5B and 8B

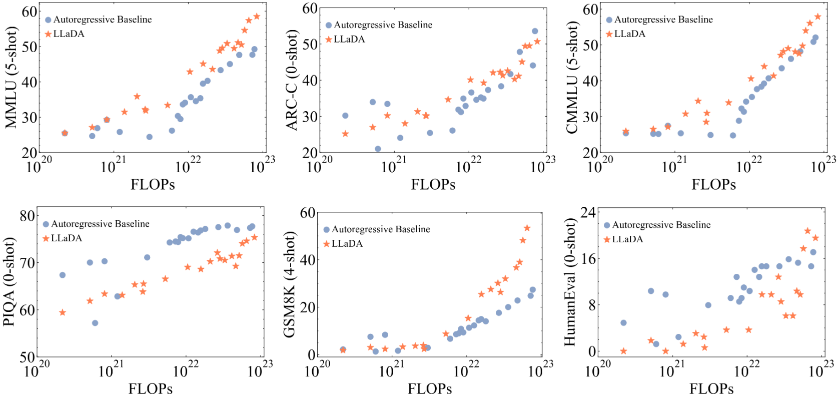
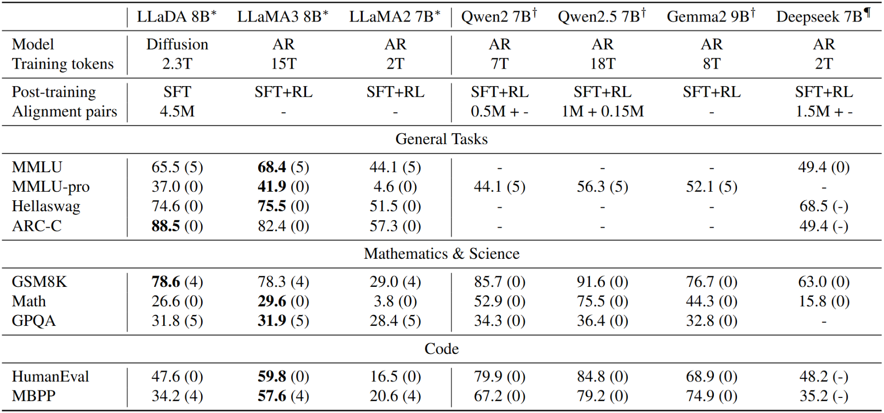


Figure 3. **Scalability of LLaDA.** We evaluate the performance of LLaDA and our ARM baselines trained on the same data across increasing computational FLOPs. LLaDA exhibits strong scalability, matching the overall performance of ARMs on six tasks.

Experiments: Benchmark of Pre-trained LLMs

	LLaDA 8B*	LLaMA3 8B*	LLaMA2 7B*	Qwen2 7B†	Qwen2.5 7B†	Mistral 7B†	Deepseek 7B¶
Model Training tokens	Diffusion 2.3T	AR 15T	AR 2T	AR 7T	AR 18T	AR -	AR 2T
General Tasks							
MMLU	**65.9** (5)	65.4 (5)	45.9 (5)	70.3 (5)	74.2 (5)	64.2 (5)	48.2 (5)
BBH	49.8 (3)	**57.6** (3)	37.3 (3)	62.3 (3)	70.4 (3)	56.1 (3)	39.5 (3)
ARC-C	47.9 (0)	**53.1** (0)	46.3 (0)	60.6 (25)	63.7 (25)	60.0 (25)	48.1 (0)
Hellaswag	72.5 (0)	**79.1** (0)	76.0 (0)	80.7 (10)	80.2 (10)	83.3 (10)	75.4 (0)
TruthfulQA	**46.4** (0)	44.0 (0)	39.0 (0)	54.2 (0)	56.4 (0)	42.2 (0)	-
WinoGrande	74.8 (5)	**77.3** (5)	72.5 (5)	77.0 (5)	75.9 (5)	78.4 (5)	70.5 (0)
PIQA	74.4 (0)	**80.6** (0)	79.1 (0)	-	-	-	79.2 (0)
Mathematics & Science							
GSM8K	**70.7** (4)	53.1 (4)	14.3 (4)	80.2 (4)	85.4 (4)	36.2 (4)	17.4 (8)
Math	**27.3** (4)	15.1 (4)	3.2 (4)	43.5 (4)	49.8 (4)	10.2 (4)	6.0 (4)
GPQA	**26.1** (5)	25.9 (5)	25.7 (5)	30.8 (5)	36.4 (5)	24.7 (5)	-
Code							
HumanEval	33.5 (0)	**34.2** (0)	12.8 (0)	51.2 (0)	57.9 (0)	29.3 (0)	26.2 (0)
HumanEval-FIM	**73.8** (2)	73.3 (2)	26.9 (2)	-	-	-	-
MBPP	38.2 (4)	**47.4** (4)	18.4 (4)	64.2 (0)	74.9 (0)	51.1 (0)	39.0 (3)
Chinese							
CMMLU	**69.9** (5)	50.7 (5)	32.5 (5)	83.9 (5)	-	-	47.2 (5)
C-Eval	**70.5** (5)	51.7 (5)	34.0 (5)	83.2 (5)	-	-	45.0 (5)

Experiments: Benchmark of Post-trained LLMs



	LLaDA 8B*	LLaMA3 8B*	LLaMA2 7B*	Qwen2 7B†	Qwen2.5 7B†	Gemma2 9B†	Deepseek 7B¶
Model	Diffusion	AR	AR	AR	AR	AR	AR
Training tokens	2.3T	15T	2T	7T	18T	8T	2T
Post-training	SFT	SFT+RL	SFT+RL	SFT+RL	SFT+RL	SFT+RL	SFT+RL
Alignment pairs	4.5M	-	-	0.5M + -	1M + 0.15M	-	1.5M + -
General Tasks							
MMLU	65.5 (5)	**68.4** (5)	44.1 (5)	-	-	-	49.4 (0)
MMLU-pro	37.0 (0)	**41.9** (0)	4.6 (0)	44.1 (5)	56.3 (5)	52.1 (5)	-
Hellaswag	74.6 (0)	**75.5** (0)	51.5 (0)	-	-	-	68.5 (-)
ARC-C	**88.5** (0)	82.4 (0)	57.3 (0)	-	-	-	49.4 (-)
Mathematics & Science							
GSM8K	**78.6** (4)	78.3 (4)	29.0 (4)	85.7 (0)	91.6 (0)	76.7 (0)	63.0 (0)
Math	26.6 (0)	**29.6** (0)	3.8 (0)	52.9 (0)	75.5 (0)	44.3 (0)	15.8 (0)
GPQA	31.8 (5)	**31.9** (5)	28.4 (5)	34.3 (0)	36.4 (0)	32.8 (0)	-
Code							
HumanEval	47.6 (0)	**59.8** (0)	16.5 (0)	79.9 (0)	84.8 (0)	68.9 (0)	48.2 (-)
MBPP	34.2 (4)	**57.6** (4)	20.6 (4)	67.2 (0)	79.2 (0)	74.9 (0)	35.2 (-)

Ablation: Reversal Reasoning

- The **Reversal Curse**: LLMs trained on ”A is B” fail to learn ”B is A”[7]
- The authors construct a dataset of 496 famous Chinese poem sentence pair. Given a sentence from a poem, models are tasked with generating the subsequent line (forward) or the preceding line (reversal) without additional fine-tuning.

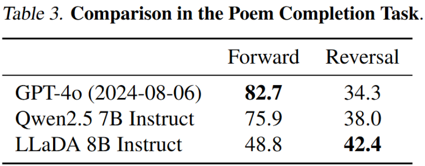


Table 3. **Comparison in the Poem Completion Task**.

	Forward	Reversal
GPT-4o (2024-08-06)	**82.7**	34.3
Qwen2.5 7B Instruct	75.9	38.0
LLaDA 8B Instruct	48.8	**42.4**

[7] Lukas Berglund et al. “The Reversal Curse: LLMs trained on” A is B” fail to learn” B is A””. In: arXiv preprint arXiv:2309.12288 (2023).

1. Introduction
2. Related Work: Diffusion Models in NLP
   - Diffusion-LM
   - Discrete Diffusion Model
3. Large Language Diffusion Models
4. Discussion

Difference between ARMs and MDMs

	ARMs	MDMs
Modeling	Unidirectional	Bidirectional
Generation flexibility	From left to right	Turning machine
Computational complexity	$O(N^3)$	$O(N^2T)$
Parallel inference	In batch	In a diffusion step
Training stability	Good	Bad

Discussion

Some weaknesses:

- Sensitive to inference hyperparameter
- No alignment with reinforcement learning
- Scale is still small

Potential strengths:

- Implicit controlling: through conditional generation $p(r \mid p, \mathbf{c})$.