Theoretical Benefit and Limitation of Diffusion Language Model

Guhao Feng*, Yihan Geng*, Jian Guan, Wei Wu, Liwei Wang, Di He

Yihan Geng @ Peking University

Paper: https://arxiv.org/abs/2502.09622

ASAP Seminar | 2025.04.23

Overview

- Introduction to Masked Diffusion Models (MDMs)
- Efficiency-Accuracy Trade-off of MDMs under Different Metrics
- Experiments and Analysis
- Future Directions

Introduction to Masked Diffusion Models

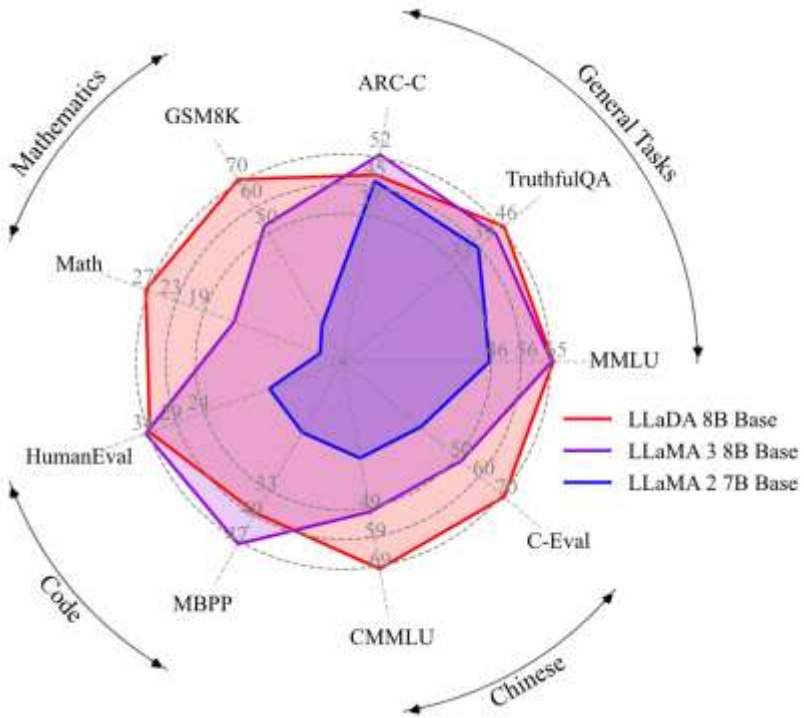
Current LLMs

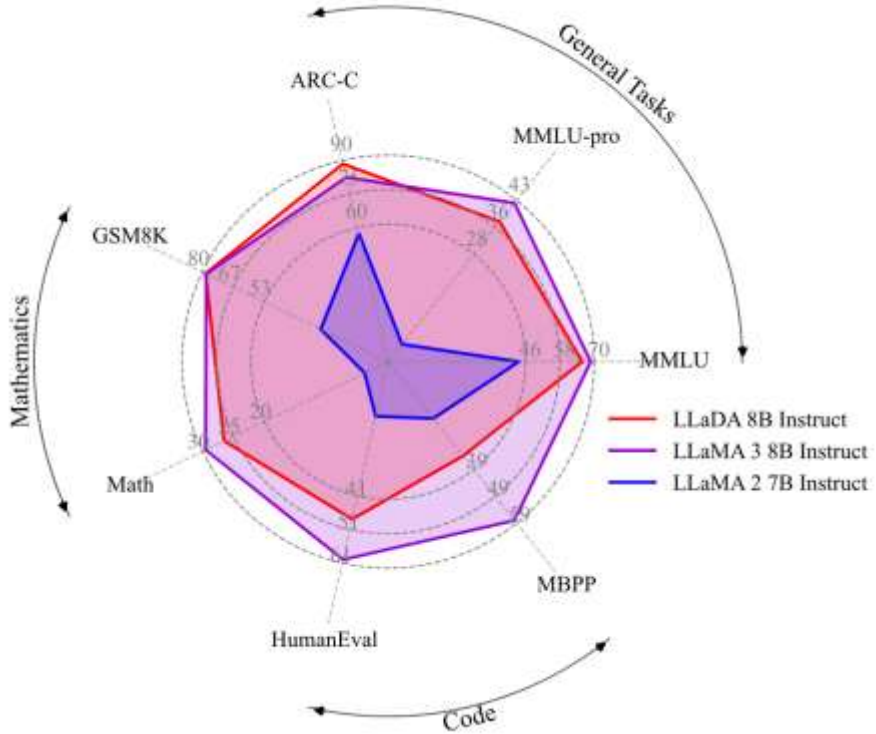
- Current Large Language Models (LLMs) have shown impressive abilities across various domains.
- Most LLMs follow the **auto-regressive** design paradigm.
  - LLMs perform Next Token Prediction, generating sequences token-by-token
- Challenges: relatively low efficiency, hard to control, etc.

How to Speed Up?

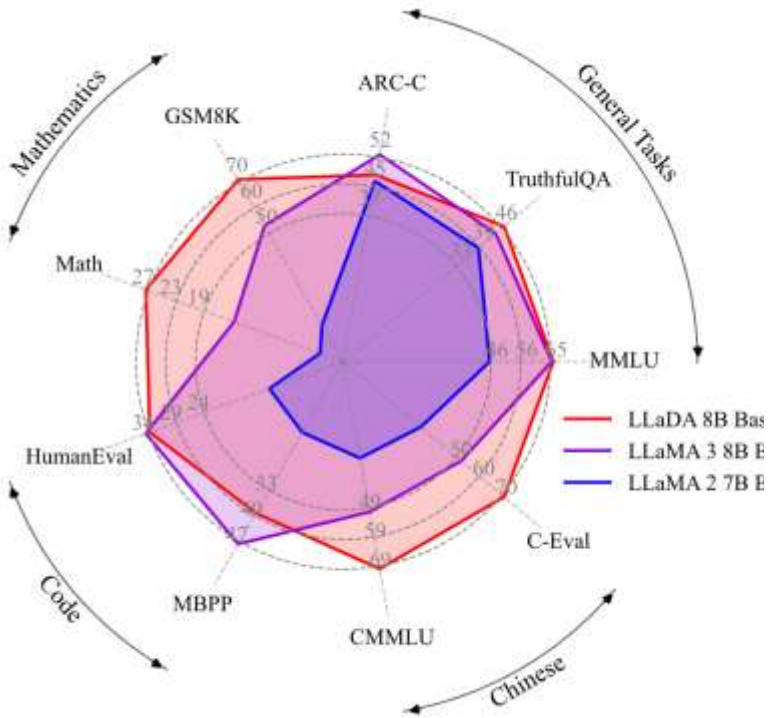
- Potential Solutions:
  - Reduce per-token cost: Efficient Transformer
  - Reduce execution rounds: Discrete Diffusion Model
  - And so on…
- Intuition: generate **multiple tokens simultaneously** during each step

Discrete Diffusion Models



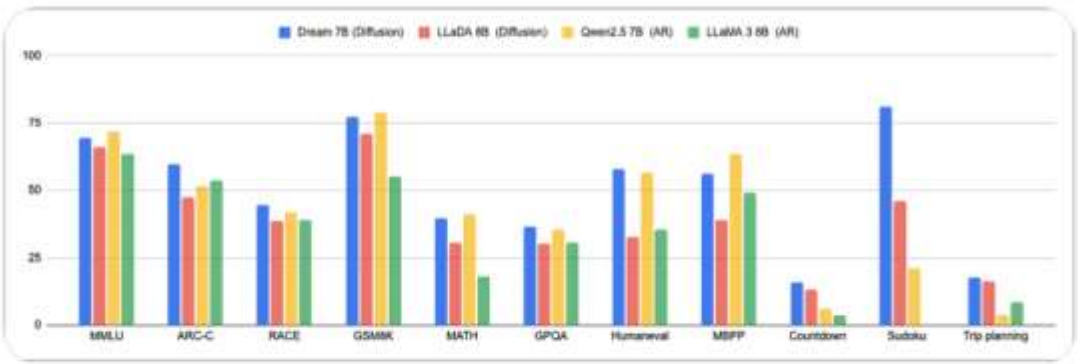


Discrete Diffusion Models



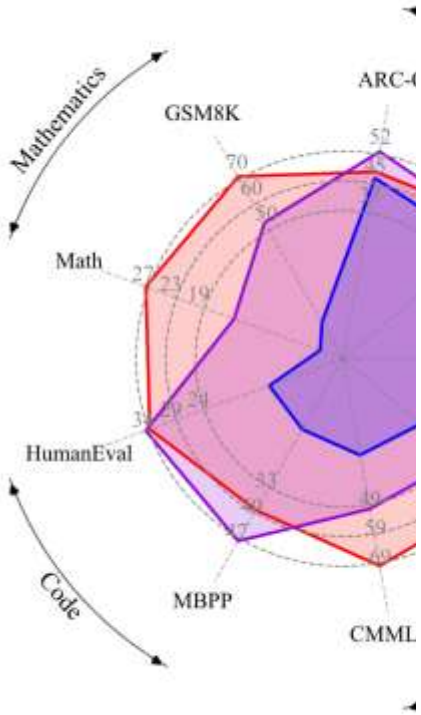
Jiacheng Ye @JiachengYe15 · 6天

🚀Excited to announce **Dream 7B** (Diffusion reasoning model): the most powerful open diffusion large language model to date.



45 255 1239 24万

Discrete



Jiacheng Ye @JiachengYe15 · 6天

In short, Dream 7B:

- outperforms existing diffusion language models by a large margin;
- matches or exceeds top-tier AR language models of similar size on the general, math, and coding abilities;
- demonstrates strong planning ability and inference flexibility.

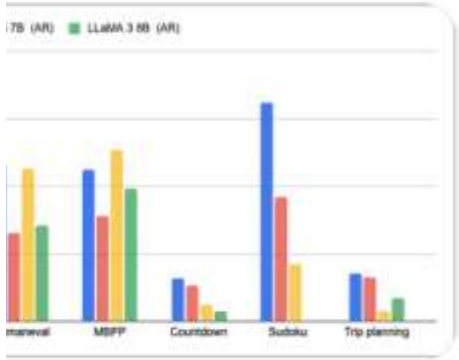
Model	Dream 7B* Diffusion	LLaDA 8B* Diffusion	Qwen2.5 7B* AR	LLaMA3 8B* AR	Mistral 7B AR	DeepSeek 7B AR
General Tasks						
MMLU	69.5 (5)	65.9 (5)	**71.9 (5)**	63.5 (5)	60.1 (5)	48.2 (5)
BBH	57.9 (3)	47.4 (3)	**63.9 (3)**	62.7 (3)	-	39.5 (3)
ARC-E	**83.9 (0)**	71.8 (0)	77.4 (0)	81.1 (0)	80.0 (0)	67.9 (0)
ARC-C	**59.8 (0)**	47.5 (0)	51.5 (0)	53.6 (0)	55.5 (0)	48.1 (0)
Hellaswag	73.3 (0)	72.7 (0)	**79.0 (0)**	78.9 (0)	81.3 (0)	75.4 (0)
WinoGrande	74.5 (5)	73.5 (5)	76.4 (5)	**76.9 (5)**	75.3 (0)	70.5 (0)
PIQA	75.8 (0)	74.8 (0)	79.8 (0)	**81.3 (0)**	83.0 (0)	79.2 (0)
RACE	**44.7 (0)**	38.7 (0)	41.9 (0)	39.2 (0)	-	46.5 (5)
Mathematics & Science						
GSM8K	77.2 (8)	70.9 (8)	**78.9 (8)**	55.3 (8)	52.1 (8)	17.4 (8)
MATH	39.6 (4)	30.7 (4)	**41.1 (4)**	18.0 (4)	13.1 (4)	6.0 (4)
GPQA	**36.6 (5)**	30.4 (5)	35.5 (5)	30.6 (5)	-	-
Code						
Humaneval	**57.9 (0)**	32.9 (0)	56.7 (0)	35.4 (0)	30.5 (0)	26.2 (0)
MBPP	56.2 (4)	39.0 (4)	**63.6 (4)**	49.2 (4)	47.5 (3)	39.0 (3)
Planning Tasks						
Countdown	**16.0 (8)**	13.2 (8)	6.2 (8)	3.7 (8)	-	-
Sudoku	**81.0 (8)**	46.0 (8)	21.0 (8)	0.0 (8)	-	-
Trip planning	**17.8 (2)**	16.4 (2)	3.6 (2)	8.7 (2)	-	-

1 9 97 1万

General Ta...

6天

m 7B (Diffusion
owerful open
el to date.



24万

Discrete Diffusion Models

- Generation process: Begins with an initial sequence, then iteratively **modifies tokens** in the sequence.
- Categories:
  - The initial sequence is composed of **mask tokens**: Masked Diffusion Models (MDMs), including SEDD Absorb, RADD, etc.
  - The initial sequence is composed of **randomly sampled tokens,** including SEDD Uniform.

sewing new shorts for your child

Masked Diffusion Model (MDM): Overview

- How does it work?
  - It iteratively refines a sequence by **replacing masked tokens** with predicted ones.
- Key Concepts:
  - Uses a **forward diffusion process** to mask tokens.
  - Uses a **reverse denoising process** to reconstruct sequences.
  - Enables **parallel token generation** instead of step-by-step decoding, potentially **speeding up inference**.

Hypothesis and Questions

- Parallel sampling improves efficiency.
- Parallel sampling may hurt generation quality.

Sam bought two dozen eggs for $5 each, so he spent $10 total.

Sam bought one dozen eggs for $5 each, so he spent $5 total.

Sam bought [mask] dozen eggs for $5 each, so he spent [mask] total.

Hypothesis and Questions

- Parallel sampling improves efficiency.
- Parallel sampling may hurt generation quality.

- We will answer:
  - Do MDMs achieve superior efficiency when the generated content meets an acceptable quality standard?

MDM: Forward Diffusion Process

- Given a sequence, the forward process gradually replaces its non-masked tokens with a **special mask token** $[m]$. The process is controlled by a **masking schedule** α_t.

- Initial sequence of length L: $\boldsymbol{x}_0$
- Sequence at time $t \in [0,1]$: $\boldsymbol{x}_\mathrm{t} = (x_t^1, x_t^2, \dots, x_t^L)$
- Fully masked sequence: $\boldsymbol{x}_1 = ([m], [m], , \dots, [m])$

- Distribution of $\boldsymbol{x}_\mathrm{t}$: $q_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \prod_{i=1}^{L} q_{t|0}(x_t^i|x_0^i)$
- Token being unchanged/masked:

$$q_{t|0}(x_t^i|x_0^i) = \begin{cases} \alpha_t, & x_t^i = x_0^i, \\ 1-\alpha_t, & x_t^i = [m]. \end{cases}$$

MDM: Reverse Denoising Process

- Given the sequence at time t, the reverse process **reconstructs the sequence** from a (partially) masked version by **reversing the forward dynamics**

- The true reverse process for $s < t$: $q_{s|t}(\boldsymbol{x}_s|\boldsymbol{x}_t) = \prod_{i=1}^{L} q_{s|t}(x_s^i|\boldsymbol{x}_t)$

- $$q_{s|t}(x_s^i|\boldsymbol{x}_t) = \begin{cases} 1, & x_t^i \neq [m], x_s^i = x_t^i, \\ \frac{1-\alpha_s}{1-\alpha_t}, & x_t^i = [m], x_s^i = [m], \\ \frac{\alpha_s-\alpha_t}{1-\alpha_t} q_{0|t}(x_s^i|\boldsymbol{x}_t), & x_t^i = [m], x_s^i \neq [m], \\ 0, & otherwise. \end{cases}$$

MDM: Approximation by Neural Network

- Neural network predicts the **probability distribution over possible tokens** for each masked position. This prediction is used for **parallel sampling**.

- $p_\theta(x_0^i \,|\boldsymbol{x}_t)$ is used to approximate $q_{0|t}(x_0^i|\boldsymbol{x}_t)$
- Prediction of the sequence: $p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_t) = \prod_{i=1}^{L} p_\theta(x_0^i \,|\boldsymbol{x}_t)$

- Each token is **predicted independently**, allowing for **efficient** parallel sampling, but it also **disregards interdependencies** between tokens within the sequence.

Efficiency-Accuracy Trade-off of MDMs under Different Metrics

How Do We Evaluate?

- Perspectives on evaluating language models:
  - **Fluency**: How natural and readable is the generated text?
  - **Correctness**: Does the generated text match certain rules?
  - **Efficiency**: How fast can the model generate text?
- Popular Metrics:
  - **Perplexity**: Evaluates the next-token-prediction ability, with lower perplexity indicating better fluency.
  - **Accuracy**: Widely used to measure math/coding ability, calculates the correctness of generated answer.

Our Evaluation Metrics

- Basic ideas:
  - For NLP tasks
  - Measures some kind of “Fluency” and “Correctness”
  - **Different metrics may lead to different conclusions**
  - Measures the performance **from different perspectives** – token level & sequence level

Our Evaluation Metrics – TER & SER

- **Token Error Rate (TER)**: Defined by perplexity.
  - Ground-truth is q, and the evaluated model is p
  - $\mathrm{TER}(p) = 2^{\mathbb{E}_{x\sim q}\left[\frac{-\log(p(x))}{|x|}\right]}$, where $\log(x)$ represents $\log_2(x)$
- **Sequence Error Rate (SER)**: Evaluates the correctness of the entire sequence by calculating the probability of generating an unlikely sequence.
  - The target language q is defined on vocabulary $\mathcal{V}$
  - $\mathrm{SER}(p) = 1 - \sum_{\boldsymbol{x}\in\mathcal{L}_q} p(\boldsymbol{x})$, where $\mathcal{L}_q = \{\boldsymbol{x} \in \mathcal{V}^* \mid q(\boldsymbol{x}) > 0\}$ is the support set of q

Comments on TER and SER

- Lower **TER** means better token-level accuracy, in other words more **fluent and coherent** text.
- **SER** evaluates the **correctness of an entire sequence** rather than individual tokens, and it is much stricter than TER.
- **SER** is particularly well-suited for tasks that demand logical consistency or reasoning.
- **Accuracy** is a special case of SER, because it only requires the correctness of the answer.

Assumption

- Intuition: we want to focus on the performance and inference cost of a **well-trained MDM**.
- Assumption: **Learning with Small Error**

- Target language with vocabulary $\mathcal{V}$: q
- Reverse model under the masking schedule α_t: p_θ
- There exists $\epsilon_{learning}$ such that for any i, t and sequence $\boldsymbol{x}_t$, $D_{KL}\left(q_{0|t}\left(x_0^i \mid \boldsymbol{x}_t\right) \| p_\theta\left(x_0^i \mid \boldsymbol{x}_t\right)\right) < \epsilon_{learning}$
- $\epsilon_{learning} \to 0 \Leftrightarrow loss \to 0$

Theoretical Guarantee for TER

Theorem (TER Bounds for Language Generation):

- For any n-gram language q, let p_θ denote the reverse model and L denote the sequence length. The distribution over sequences generated by p_θ is denoted as p.
- Under the Assumption, for any ϵ and $L \geq O\left(\frac{n-1}{\epsilon^{n+0.5}}\right)$, there exists α_t such that, with $N = O\left(\frac{n-1}{\epsilon^{n+0.5}}\right)$ sampling steps, the TER of the MDM is upper-bounded by:

$$\log \mathrm{TER}(p) \leq \log \mathrm{TER}(q) + \epsilon_{learning} + 4\epsilon \log |\mathcal{V}|$$

- Intuition: MDMs can generate long sequences efficiently with high fluency.

Comments on TER

- For any reverse model p, the optimal TER is TER(q).
- To obtain TER close to the optimal, MDMs only require at most $O\left(\frac{n-1}{\epsilon^{n+0.5}}\right)$ sampling steps.
- For sufficiently long sequences, the number of **sampling steps** needed is **independent of sequence length**, which offers advantage over auto-regressive models.
- Thus, **MDMs are efficient regarding TER**.

High SER with Sufficient Steps

Theorem (Accurate Generation of HMM with Sufficient Steps):

- For any HMM q, let p_θ denote the reverse model and L denote the sequence length. The distribution over sequences generated by p_θ is denoted as p.
- For any $\delta > 0$, under the Assumption with $\epsilon_{learning} < O\left(\frac{\delta}{L}\right)$, and **given a sufficient number of steps** (i.e. when N is large enough), the SER of generated text satisfies:

$$\mathrm{SER}(p) \leq \delta$$

- Intuition: MDMs have the ability to generate correct sequences with sufficient steps.

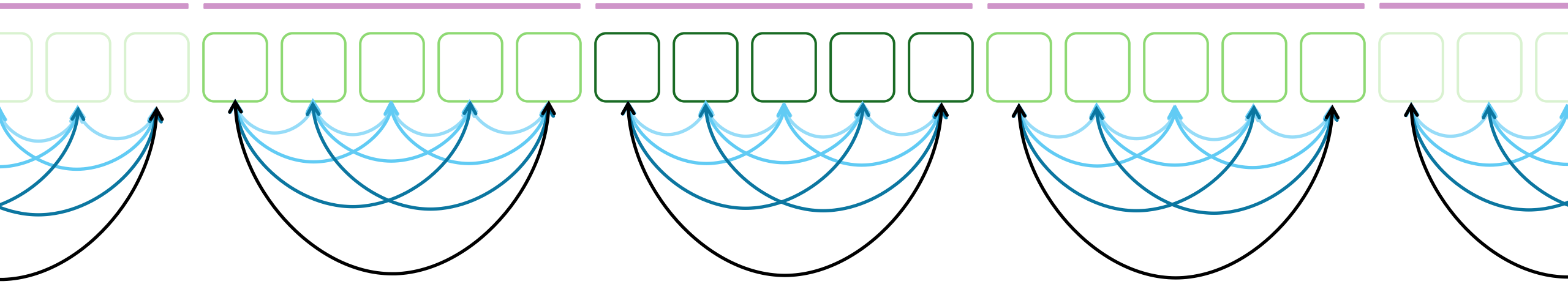
Inefficiency Regarding SER

Theorem (SER Bound for HMM Generation):

- **There exists an HMM** q over a vocabulary of size 16 that satisfies the following conditions: for any reverse model p_θ under the Assumption with $\epsilon_{learning} < \frac{1}{128}$, and any masking schedule α_t, lets p denote sequences generated by p_θ.
- There exists a constant C such that if the number of sampling steps satisfies $N = CL$ for sequence length L, the SER of the generated text is lower-bounded by:

$$\mathrm{SER}(p) > \frac{1}{2}$$

The Example



Comments on SER

- To generate sequences with low SER, the number of sampling steps in MDMs may have to **scale at least linearly** with the sequence length L.
- For sufficiently long sequences, the requirement for SER can be much **stricter than TER**.
- Given that the average computational cost of one inference step in auto-regressive models is lower than that of MDMs, MDMs are **less efficient** compared to AR regarding SER.

Implications of the Theory

- MDMs can efficiently generate low-TER sentences, but may incur higher costs when evaluating the generation under SER.
- Do TER and SER Conflict?
  - **Evaluation metrics heavily influence conclusions.** For example, some have questioned that "Emergence" in AI is caused by non-smooth metrics.
  - There is **no "gold" metric for all tasks**, and it is natural to arrive at different conclusions depending on the metric used.
  - For example, perplexity is criticized for not accurately gauging model performance in complex tasks, as shown by prior studies.

Takeaways

- The efficiency of MDMs heavily depends on the evaluation metric employed.
- Specifically, MDMs can produce low-TER outputs efficiently, but struggle with SER.
- Compared to auto-regressive models, MDMs *prioritize fluency* and are *effective* for **language generation**, but are *less efficient* in **tasks requiring precise reasoning or accuracy**.

Experiments and Analysis

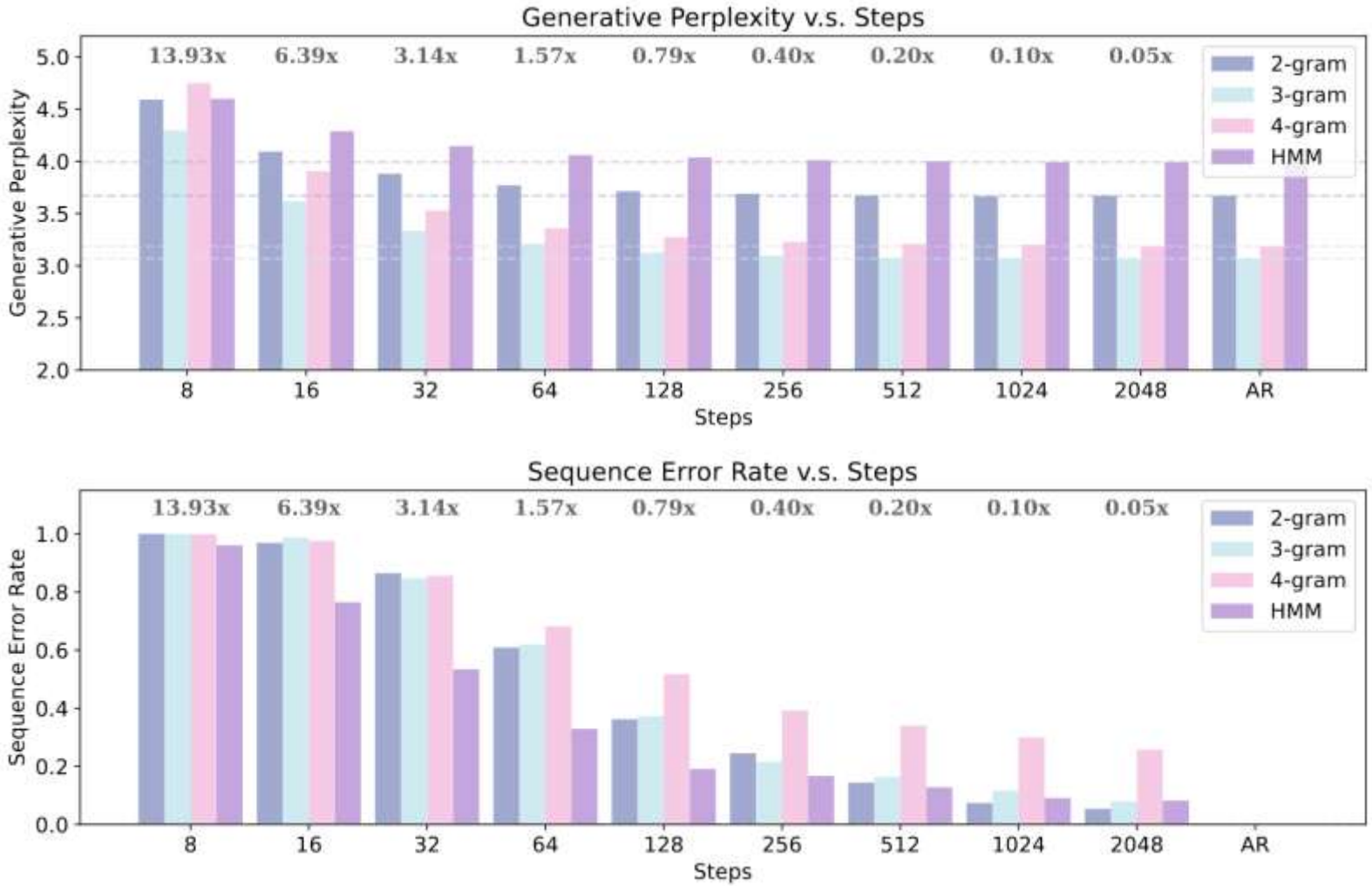
Experiments

- Experiments on Synthetic Languages
  - n-Gram Languages (n=2,3,4)
  - HMM
- Experiments on Large Models
  - Text Generation
  - Mathematical Reasoning

Settings for Synthetic Experiments

- We generate n-Gram and HMM languages with randomly chosen parameters (transition matrices, observation matrices, initial distribution etc.)
- In order to evaluate SER, we set a threshold to prune the tail probabilities, ensuring that there exists “incorrect” sequences.
- We trained our masked diffusion models and auto-regressive models for 20 epochs on 1,000,000 generated samples (with 10,000 used as validation set).

TER and SER Results



* Length = 512

TER Regarding Different Sequence Lengths

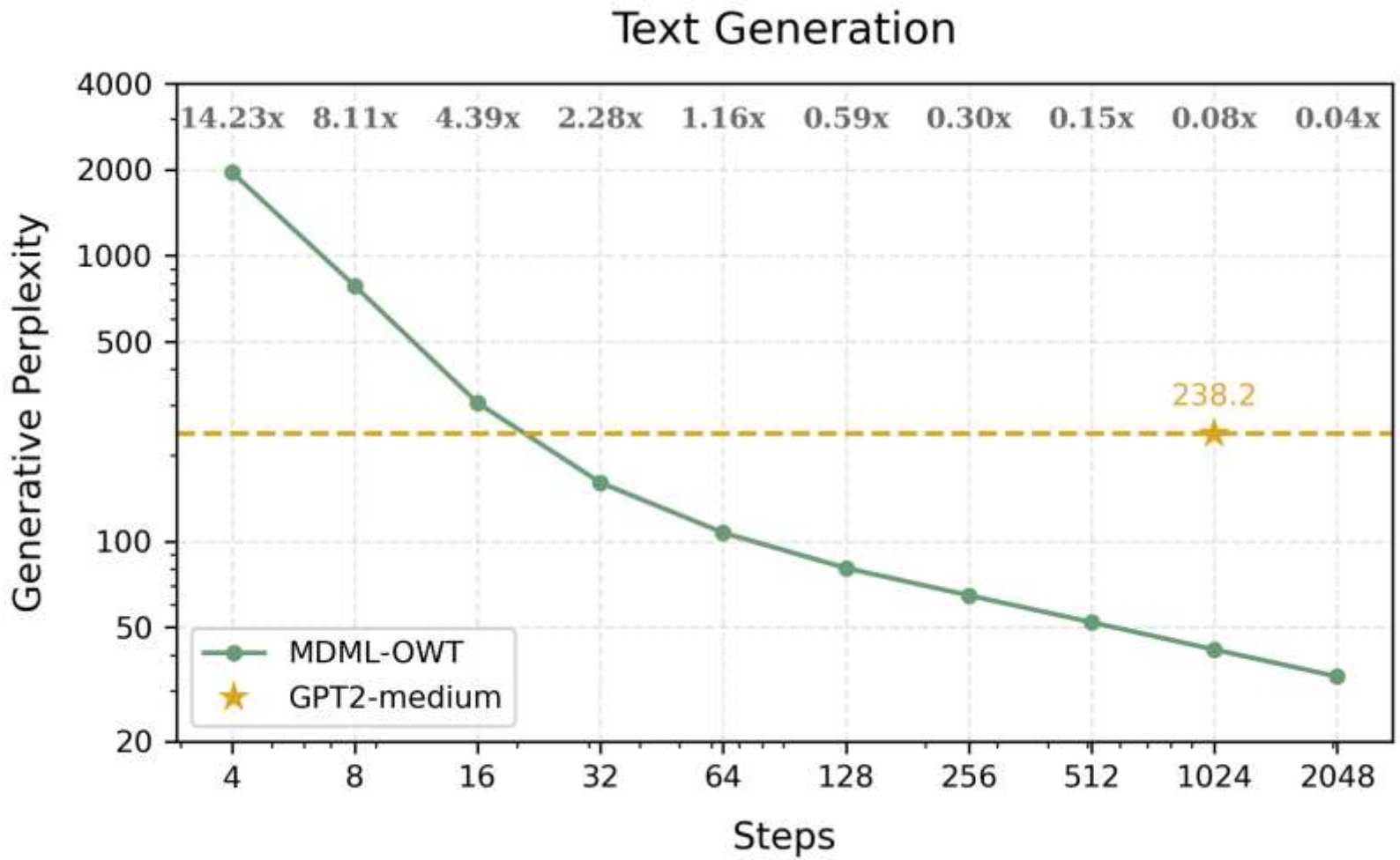
Sampling Steps	64	128	256	512	1024	2048
2-gram Models						
L = 256	2.72%	1.09%	0.27%	0.11%	0.10%	0.05%
L = 512	2.72%	1.09%	0.54%	0.09%	0.05%	0.02%
L = 1024	2.46%	1.09%	0.55%	0.13%	0.07%	0.02%
3-gram Models						
L = 256	4.58%	1.63%	0.98%	0.33%	0.15%	0.08%
L = 512	4.56%	1.95%	0.98%	0.42%	0.17%	0.09%
L = 1024	4.56%	2.28%	0.98%	0.65%	0.33%	0.13%
4-gram Models						
L = 256	5.26%	3.10%	1.86%	1.24%	0.93%	0.06%
L = 512	5.33%	2.51%	1.25%	0.63%	0.31%	0.07%
L = 1024	5.37%	2.91%	1.45%	0.84%	0.35%	0.05%

- Evaluates the relative increase of TER:

$$\epsilon_{total} = \frac{\mathrm{TER}(p_\theta)}{\mathrm{TER}(q)} - 1$$

Sequence Length	256	512	1024	2048	4096	8192
MDMs (512 steps)	2.0s	3.1s	4.2s	4.7s	4.7s	7.4s
AR	0.9s	1.7s	3.3s	7.0s	15.4s	45.8s

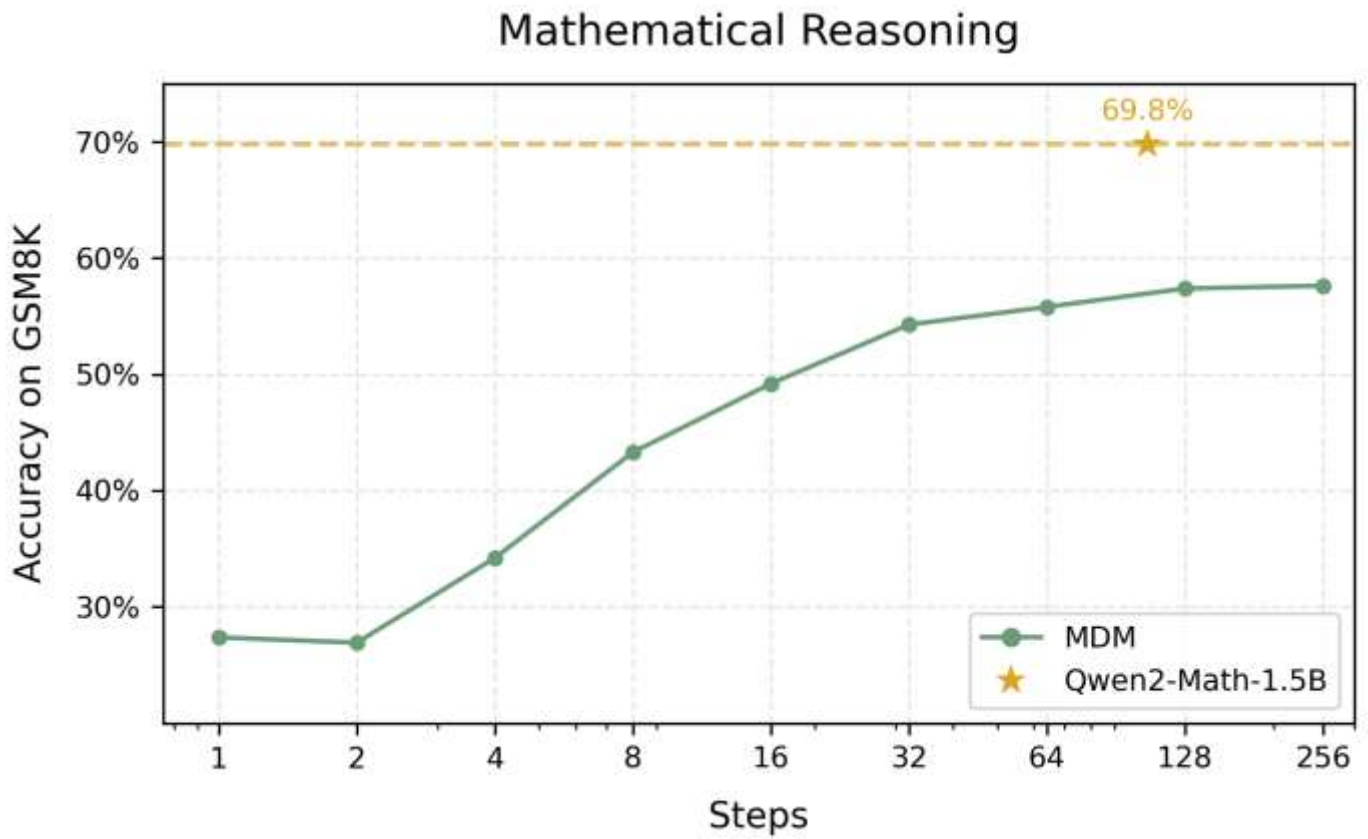
Text Generation



* Length = 1024

Mathematical Reasoning

- Average length of generated answer: 30



Future Directions

Future Directions

- Potential Ways to Mitigate SER Limitation:
  - Advanced sampling methods
  - Error correction mechanisms
    - e.g. remasking strategies
  - Discrete diffusion models beyond the masked setting
    - e.g. models based on SEDD-Uniform, DFM, etc.
- Extending the theory into a broader family of diffusion-based language model and real-world settings
- The approximation error of discrete diffusion models

Thanks!