Text Diffusion Model

2025.10.30 (목)

홍성태

ghdchlwls123@korea.ac.kr

Papers

Large Language Diffusion Models

Shen Nie[1,2,3*†] **Fengqi Zhu**[1,2,3*†] **Zebin You**[1,2,3†] **Xiaolu Zhang**[4‡] **Jingyang Ou**[1,2,3]
Jun Hu[4‡] **Jun Zhou**[4] **Yankai Lin**[1,2,3‡] **Ji-Rong Wen**[1,2,3] **Chongxuan Li**[1,2,3‡§]
[1] Gaoling School of Artificial Intelligence, Renmin University of China
[2] Beijing Key Laboratory of Research on Large Models and Intelligent Governance
[3] Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE
[4] Ant Group
{nieshen,fengqizhu,chongxuanli}@ruc.edu.cn

ANT GROUP | INCLUSION AI

中国人民大学高瓴人工智能学院 Gaoling School of Artificial Intelligence, Renmin University of China

GSAI-ML

LLaDA-MoE: A Sparse MoE Diffusion Language Model

Fengqi Zhu[1,2,*], **Zebin You**[1,2,*], **Yipeng Xing**[2,*], **Zenan Huang**[2,*], **Lin Liu**[2,*], **Yihong Zhuang**[2,*], **Guoshan Lu**[2,*], **Kangyu Wang**[2,3], **Xudong Wang**[2], **Lanning Wei**[2], **Hongrui Guo**[2], **Jiaqi Hu**[2,4], **Wentao Ye**[2,4], **Tieyuan Chen**[2,3], **Chenchen Li**[2], **Chengfu Tang**[2], **Haibo Feng**[2], **Jun Hu**[2], **Jun Zhou**[2], **Xiaolu Zhang**[2,‡], **Zhenzhong Lan**[2,‡], **Junbo Zhao**[2,4,‡], **Da Zheng**[2,‡], **Chongxuan Li**[1,†], **Jianguo Li**[2,†], **Ji-Rong Wen**[1,†]

[1]Renmin University of China, [2]Ant Group, [3]Shanghai Jiao Tong University, [4]Zhejiang University

Large Language Diffusion Models

Shen Nie[1,2,3*†] **Fengqi Zhu**[1,2,3*†] **Zebin You**[1,2,3†] **Xiaolu Zhang**[4‡] **Jingyang Ou**[1,2,3]
Jun Hu[4‡] **Jun Zhou**[4] **Yankai Lin**[1,2,3‡] **Ji-Rong Wen**[1,2,3] **Chongxuan Li**[1,2,3‡§]
[1] Gaoling School of Artificial Intelligence, Renmin University of China
[2] Beijing Key Laboratory of Research on Large Models and Intelligent Governance
[3] Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE
[4] Ant Group
{nieshen,fengqizhu,chongxuanli}@ruc.edu.cn

Prompt: Explain what artificial intelligence is.

Introduction

LLM의 현주소: Autoregressive Model (ARM)

현존하는 대부분의 LLM(GPT, LLaMA 등)은 자기회귀 모델(ARM), 즉 ′ Causal Language Modeling' 기반으로 수행

Question: "LLM의 핵심 역량은 오직 'Auto-Regressive Modeling' 패러다임을 통해서만 달성할 수 있는가?"

Answer: "아니다."

LLM의 핵심 속성은 ARM 고유의 것이 아니라, 더 상위 개념인 Generative Modeling Principles 비롯

: 트랜스포머 아키텍처, 데이터/모델 크기, 피셔 일관성(Fisher consistency) 등의 상호작용 결과임

: ICL, 지시 사항 준수 등은 ARM만의 전유물이 아님

ARM 패러다임의 내재적 한계 – Reversal Curse (Reversal Reasoning 작업에는 구조적으로 취약)

Introduction

LLaDA – Diffusion 모델을 통한 새로운 접근

목표: ARM이 아닌 생성 모델링 원칙(확산 모델)을 통해 LLM의 핵심 역량이 발현될 수 있음을 증명하고자 함

LLaDA의 동작 원리

- Forward Process (Data Masking): 원본 데이터(문장)에 점진적으로 마스크(noise)를 추가
- Reverse Process (Generation): 트랜스포머 기반의 Mask Predictor가 마스킹된 토큰을 예측(복원)하도록 학습

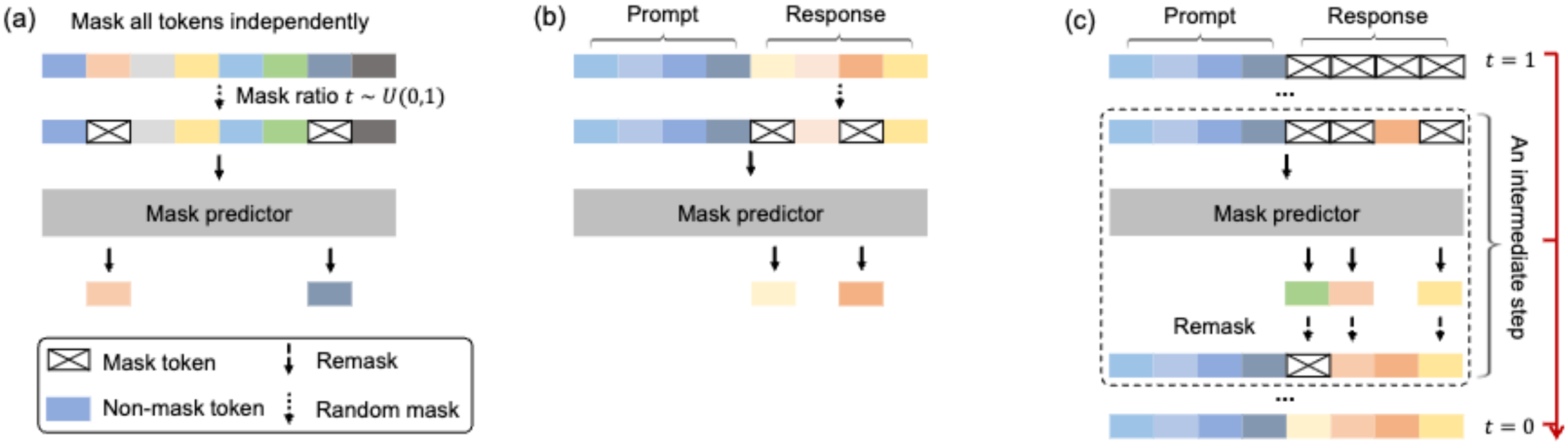
주요 특징:

1. Bidirectional Dependencies를 자연스럽게 모델링
2. Variational Lower Bound, VLB을 최적화하는 원칙적인(principled) 생성 접근 방식을 따름

Training

LLaDA

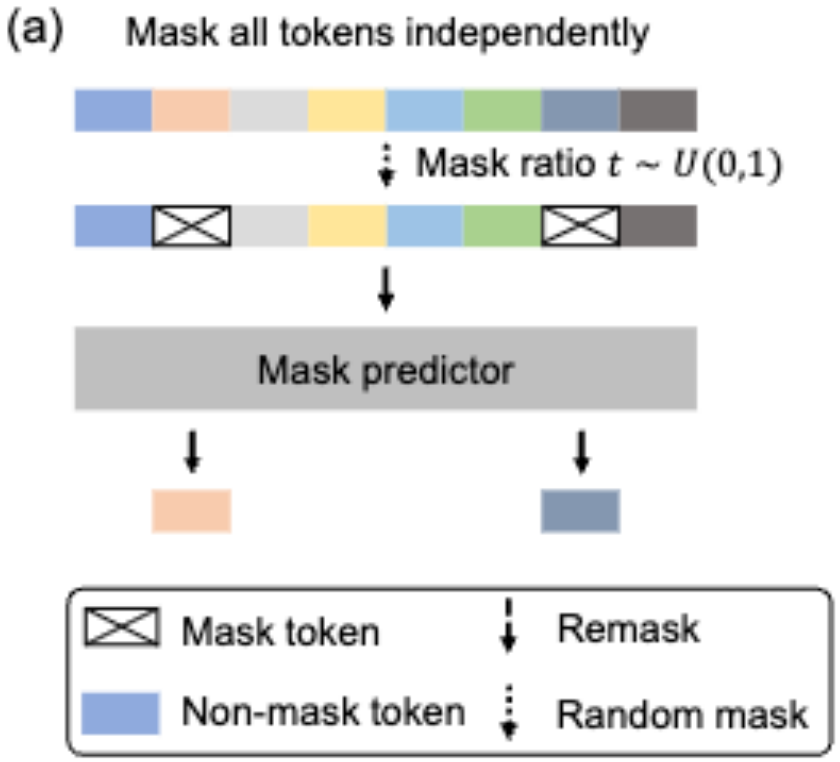
Overview



LLaDA

Training

Pre-training (1)



Transformer를 Mask Predictor로 쓰자

: Causal Mask ➔ Non-Causal Mask

- 기존 ARM 모델과 달리, 예측 시 전체 입력을 참조

vs LLaMA3 8B

- LLaMA3: GQA / LLaDA: MHA (LLaDA는 KV Caching 호환 x)
- MHA사용으로 늘어난 파라미터는 FFN 차원 축소를 통해 사이즈 맞춤

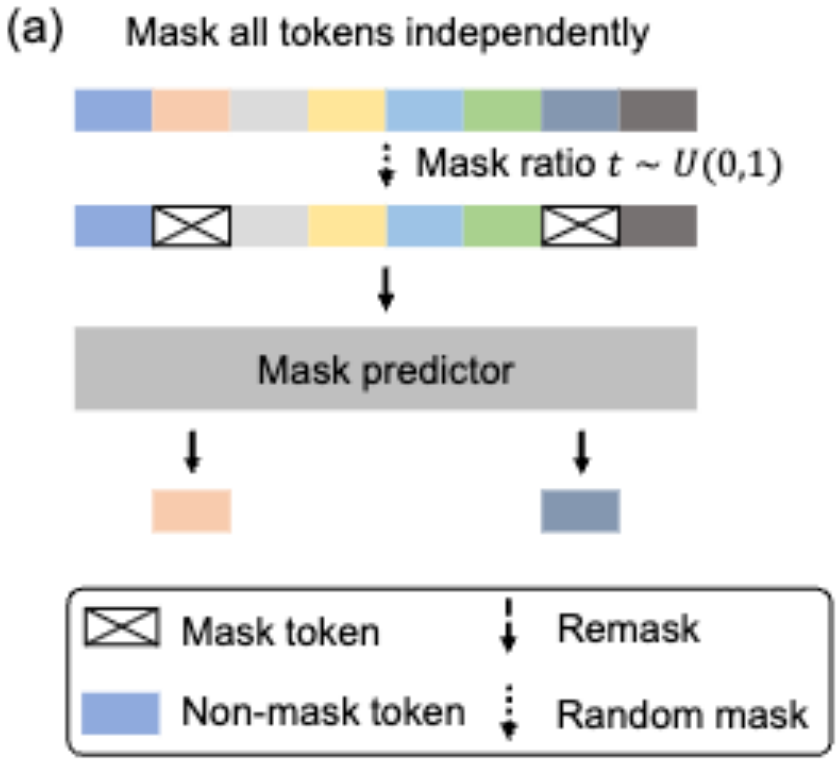
Data

- Online Corpora: 고품질 코드, 수학, 다국어 데이터 2.3T

LLaDA

Training

Pre-training (2)



Masking & Objective

- [0, 1] 범위에서 마스킹 확률 t를 무작위 샘플링
- 원본 시퀀스 x_0의 각 토큰을 독립적으로 t 확률로 마스킹하여 x_t 생성
- 고정 길이 (99%): 4096 토큰
- 가변 길이 (1%): [1, 4096] 범위에서 랜덤 샘플링

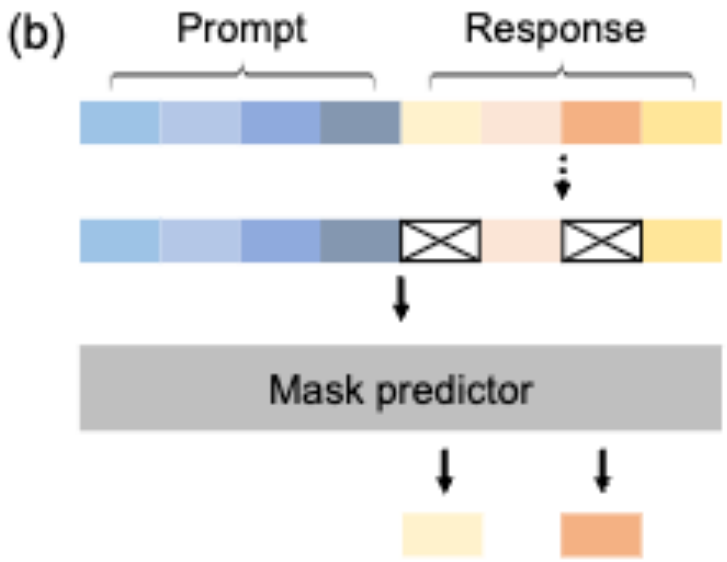
Hyperparams

- AdamW (Weight Decay: 0.1)
- 배치 크기: 1280 (Global) / GPU당 4 (Local)
- Warmup-Stable-Decay
  - Warmup (2k iter): 0 ➔ 4e-4
  - Stable 1 (1.2T tok): 4e-4
  - Decay 1 (0.8T tok): 1e-4
  - Final Decay (0.3T tok): 1e-5

LLaDA

Training

SFT



Instruction Following

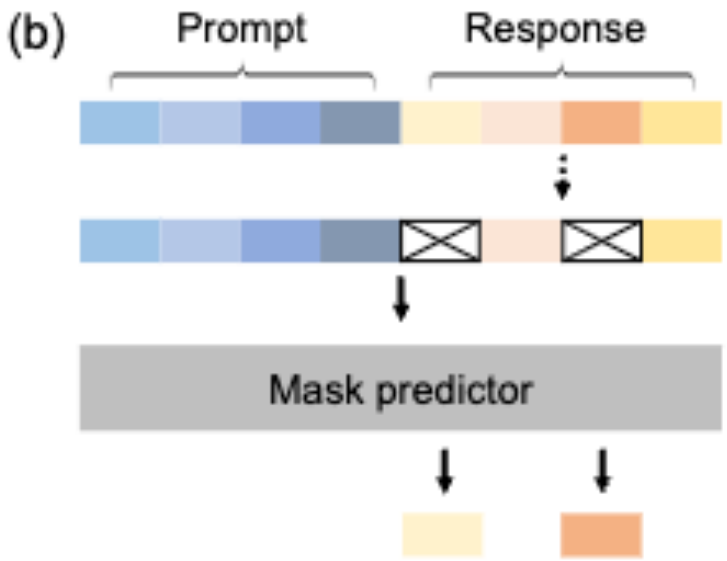
: SFT를 사전 학습(PT)과 완벽하게 호환되는 방식으로 구현

- 프롬프트 (p_0): 변경 없음 (마스킹 X)
- 응답 (r_0): PT와 동일하게 t 확률로 독립적 토큰 마스킹
- 모델 입력: p_0와 r_t를 연결 ([p_0, r_t])
- Loss 계산: r_t 부분의 마스킹된 토큰에 대해서만 계산

LLaDA

Training

SFT



Data

- 450만 개의 쌍[Code, Mathematics, Instruction-following 등 다중 도메인]

Batch & EOS

: prompt+response의 끝에 |EOS| 토큰을 추가하여 길이 통일

- 학습 시: **일반 토큰**으로 취급하여 학습 (그냥 PAD 대신에 쓴다는말ㅇㅇ)
- 추론 시: |EOS|를 샘플링하면 응답 생성 중단
- 모델이 응답 길이를 자동으로 제어하는 능력을 학습

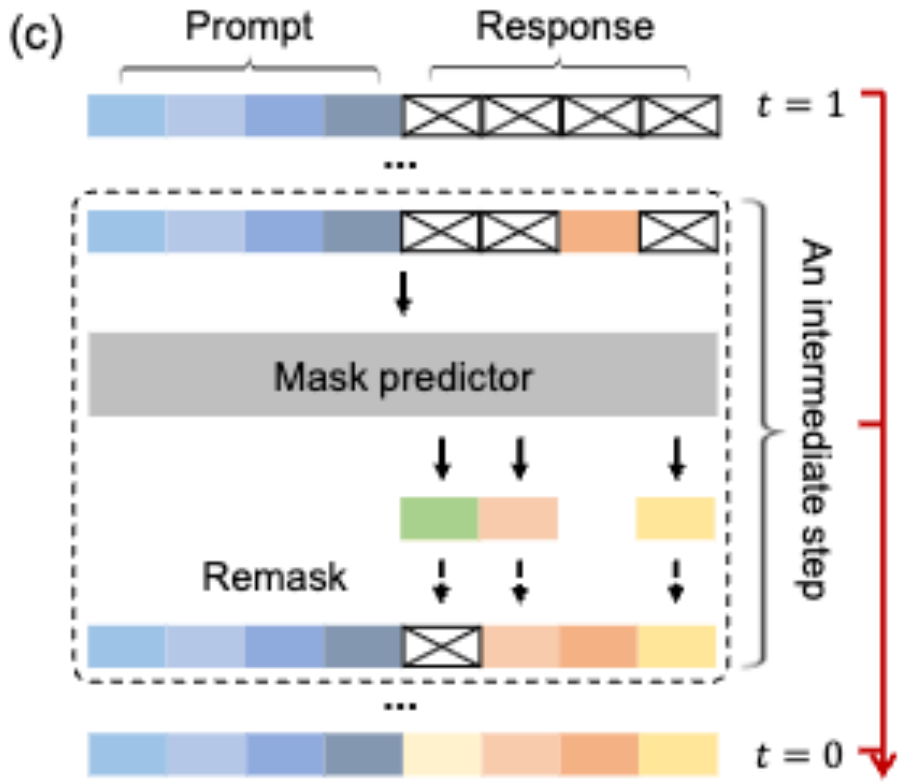
Hyperparams

- Epochs: 3
- LR: 2.5e-5 (Max) [Warm-up: 50 iter ➔ 유지 ➔ 마지막 10% 선형 감소]
- Weight Decay: 0.1
- Batch : 256 (Global) / GPU당 2 (Local)

LLaDA

Inference

Predicting (1)



LLaDA의 추론 방식: Non-Autoregressive

- AR: left-to-right) 한 토큰씩 순차적 생성
- LLaDA: 확산(Diffusion) 방식 채택 [동시 예측이 핵심]

추론 과정: Reverse Generation

시작 (T=1): 프롬프트(p_0) + 완전히 마스킹된 응답 [M]...[M]

중간 (t → s): [p_0, r_t] (일부 마스킹된 응답) ➔ 마스크 예측기로 모든 [M] 토큰 동시 예측

종료 (T=0): [p_0, r_0] (완전한 응답)

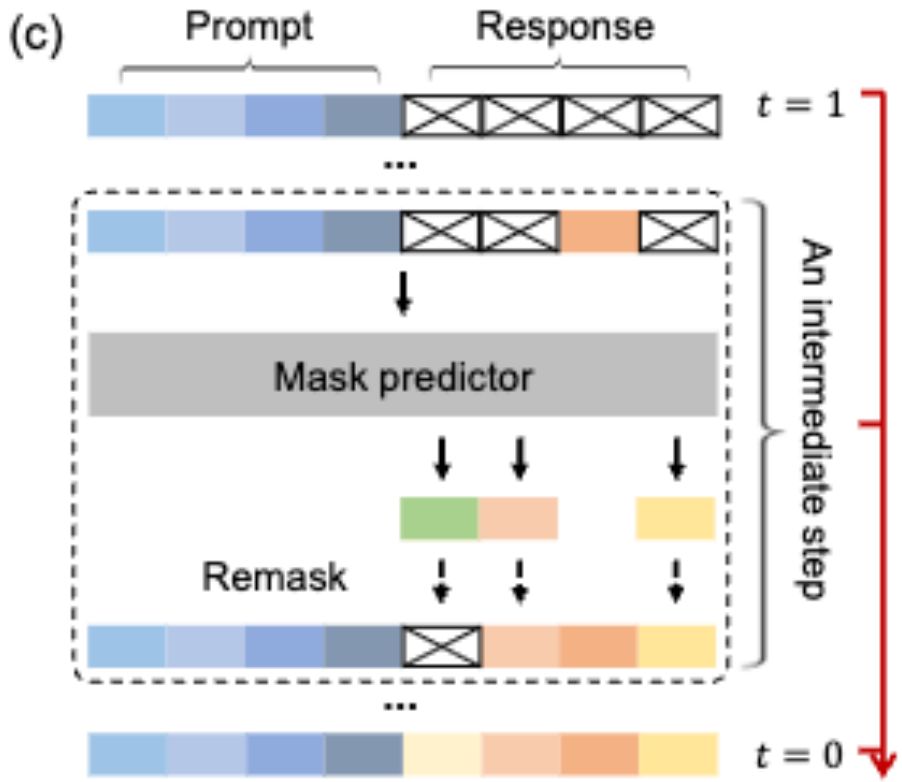
Core Parameter

- 총 샘플링 단계 (Steps): 추론 속도(효율성) vs 샘플 품질 간의 트레이드오프 발생
- 생성 길이 (Length): 샘플링 시작 시 생성할 [M]의 개수 (예: 256, 512...)

LLaDA

Inference

Predicting (2)



핵심 전략: Remasking for Efficiency

- t시점에서 예측한 토큰을 s시점에 모두 믿어버리면(채워 넣으면), diffusion의 transition이 깨지고 샘플링이 불안정
- t ➔ s 단계로 이동 시
  1. t시점의 모든 [M]을 예측
  2. 예측된 토큰 중 일부 비율을 다시 마스킹하여 r_s를 생성

Low-Confidence

원칙적으로 순수하게 무작위로 다시 마스킹해야 함

But, annealing 기법에서 영감을 받아 Low-Confidence Remasking 수행

모델이 예측한 토큰들 중 confidence가 낮은 토큰을 골라 특정 비율만큼 우선적으로 리마스킹

LLaDA

Inference

Remasking Algorithm

Algorithm 4 Random Remasking Strategy of LLaDA

Require: mask predictor p_θ, prompt p_0, answer length L, sampling steps N

Set r_1 is a fully masked sequence of length L.
for $t \leftarrow 1$ **down to** $\frac{1}{N}$ **step** $\frac{1}{N}$ **do**
  $s = t - \frac{1}{N}$
  $r_0 = \arg\max_{r_0} p_\theta(r_0 | p_0, r_t)$ # we employ greedy sampling when predicting masked tokens
  for $i \leftarrow 1$ to L **do**
    if $r_t^i \neq$ M **then**
      $r_0^i = r_t^i$
    else
      with probability $\frac{s}{t}$, r_0^i is set to M
    end if
  end for
  $r_s = r_0$
end for
Return r_0

t (Time) = 현재 상태의 마스킹 비율

s (Step) = 다음 단계의 목표 마스킹 비율

N=4로 정하면

- t는 1.0 ➔ 0.75 ➔ 0.5 ➔ 0.25 ➔ 0.0
- s는 0.75 ➔ 0.5 ➔ 0.25 ➔ 0.0 ➔ 완료

역과정은 t=1 (전부 마스크)에서 출발해 t=0 (최종응답)으로 가는 과정

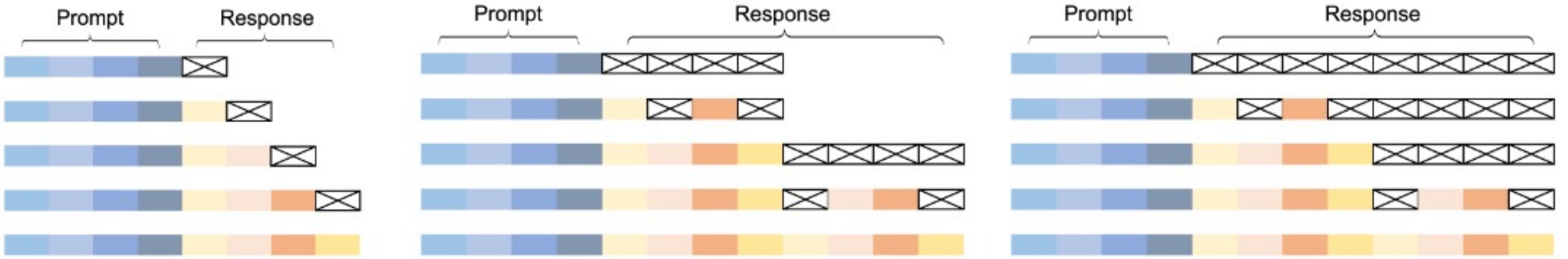
실제로는 연속 시간을 여러 단계로 나눠서(discretize) 진행

: 사용자가 N을 정하면, 1.0에서 0.0까지를 N개의 균일한 간격으로 나눔

LLaDA

Inference

Sampling Strategy for Flexibility



(a) Autoregressive. (b) Block Diffusion. (c) Block Diffusion LLaDA.

Default: 완전히 마스킹된 상태에서 시작 (parallel decoding)

(a) AR – LLaDA도 기존 LLM처럼 동작 가능

(b) Block Diffusion – 블록 간 Autoregressive

(c) Block Diffusion – Semi-autoregressive

Experiments Results (1)

Benchmark Results of Pre-trained LLMs.

	LLaDA 8B*	LLaMA3 8B*	LLaMA2 7B*	Qwen2 7B†	Qwen2.5 7B†	Mistral 7B†	Deepseek 7B¶
Model	Diffusion	AR	AR	AR	AR	AR	AR
Training tokens	2.3T	15T	2T	7T	18T	-	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	49.7 (3)	**62.1** (3)	39.4 (3)	62.3 (3)	70.4 (3)	56.1 (3)	39.5 (3)
ARC-C	45.9 (0)	**53.1** (0)	46.3 (0)	60.6 (25)	63.7 (25)	60.0 (25)	48.1 (0)
Hellaswag	70.5 (0)	**79.1** (0)	76.0 (0)	80.7 (10)	80.2 (10)	83.3 (10)	75.4 (0)
TruthfulQA	**46.1** (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	73.6 (0)	**80.6** (0)	79.1 (0)	-	-	-	79.2 (0)
Mathematics & Science							
GSM8K	**70.3** (4)	48.7 (4)	13.1 (4)	80.2 (4)	85.4 (4)	36.2 (4)	17.4 (8)
Math	**31.4** (4)	16.0 (4)	4.3 (4)	43.5 (4)	49.8 (4)	10.2 (4)	6.0 (4)
GPQA	25.2 (5)	**25.9** (5)	25.7 (5)	30.8 (5)	36.4 (5)	24.7 (5)	-
Code							
HumanEval	**35.4** (0)	34.8 (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	40.0 (4)	**48.8** (4)	23.2 (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 Results (2)

Benchmark Results 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	69.4 (4)	**78.3** (4)	29.0 (4)	85.7 (0)	91.6 (0)	76.7 (0)	63.0 (0)
Math	**31.9** (0)	29.6 (0)	3.8 (0)	52.9 (0)	75.5 (0)	44.3 (0)	15.8 (0)
GPQA	**33.3** (5)	31.9 (5)	28.4 (5)	34.3 (0)	36.4 (0)	32.8 (0)	-
Code							
HumanEval	49.4 (0)	**59.8** (0)	16.5 (0)	79.9 (0)	84.8 (0)	68.9 (0)	48.2 (-)
MBPP	41.0 (4)	**57.6** (4)	20.6 (4)	67.2 (0)	79.2 (0)	74.9 (0)	35.2 (-)

Reversal Reasoning and Analyses

Reversal Curse

: ARMs가 "A는 B이다"와 같은 사실을 학습할 때 "B는 A이다"라는 역방향 사실을 추론하는 데 어려움을 겪음

: ARM은 L2R 방식으로 처리하고 생성하는 본질적인 특성으로, 입력 시퀀스의 방향성에 강한 inductive bias 가짐

평가 프로토콜 (Poem Completion Task)

: LLaDA의 Reversal Reasoning 능력을 정량적으로 평가하기 위해 Allen-Zhu and Li [35]의 프로토콜을 차용

: 496쌍의 유명한 중국 시 문장으로 구성된 데이터셋을 구축

Forward Task: 다음 줄을 생성. Reversal Task: 이전 줄을 생성

Reversal Reasoning and Analyses

Table 4: **Comparison on 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	51.8	**45.6**

LLaDA는 ....

- Forward 성능은 ARM들보다 낮지만 Reversal 성능에서 큰 우위 → 전체적으로 더 균형 잡힌 처리 가능
- masked diffusion의 훈련으로 토큰을 균등하게 취급하여 방향성 편향이 적음

LLaDA

Conclusion

8B 규모의 Diffusion 언어 모델을 최초로 제안

기존 ARM의 한계를 극복하고 언어 모델링의 새로운 패러다임 제시

Bidirectional Modeling & Enhanced Robustness (e.g., Reversal Curse)

Limitations

- 강화 학습을 통한 정렬 과정을 거치지 않음
- 확산 샘플링 알고리즘이 아직 예비 단계
- 계산 제약으로 ARM과 동일한 데이터/규모로 직접 비교 X

Future Work

- 모델 및 데이터 확장: SOTA ARM 모델 규모로 확장 필요
- 다중 모드 (Multi-modal) 데이터 처리 능력 탐구
- RL 기반 정렬 적용을 통한 성능 및 인간 의도 일치성 개선

LLaDA-MoE: A Sparse MoE Diffusion Language Model

Fengqi Zhu[1,2,*], Zebin You[1,2,*], Yipeng Xing[2,*], Zenan Huang[2,*], Lin Liu[2,*], Yihong Zhuang[2,*], Guoshan Lu[2,*], Kangyu Wang[2,3], Xudong Wang[2], Lanning Wei[2], Hongrui Guo[2], Jiaqi Hu[2,4], Wentao Ye[2,4], Tieyuan Chen[2,3], Chenchen Li[2], Chengfu Tang[2], Haibo Feng[2], Jun Hu[2], Jun Zhou[2], Xiaolu Zhang[2,‡], Zhenzhong Lan[2,‡], Junbo Zhao[2,4,‡], Da Zheng[2,‡], Chongxuan Li[1,†], Jianguo Li[2,†], Ji-Rong Wen[1,†]

[1]Renmin University of China, [2]Ant Group, [3]Shanghai Jiao Tong University, [4]Zhejiang University

LLaDA-MOE

Introduction

기존 MDM의 아키텍처 한계

- 현재까지의 MDM 연구는 대부분 Dense한 트랜스포머 백본에 의존
- 대조적으로, AR 모델 분야에서는 Sparse Mixture-of-Experts 아키텍처가 널리 검증됨

MDM을 Sparse MoE 아키텍처로 scratch부터 사전 학습시킨 선행 연구가 부재 ➔ 효율성과 성능을 모두 잡자

어떻게? 약 20T(조) 토큰 데이터로 학습된 MDM과 Sparse MoE 아키텍처를 가진 모델 학습

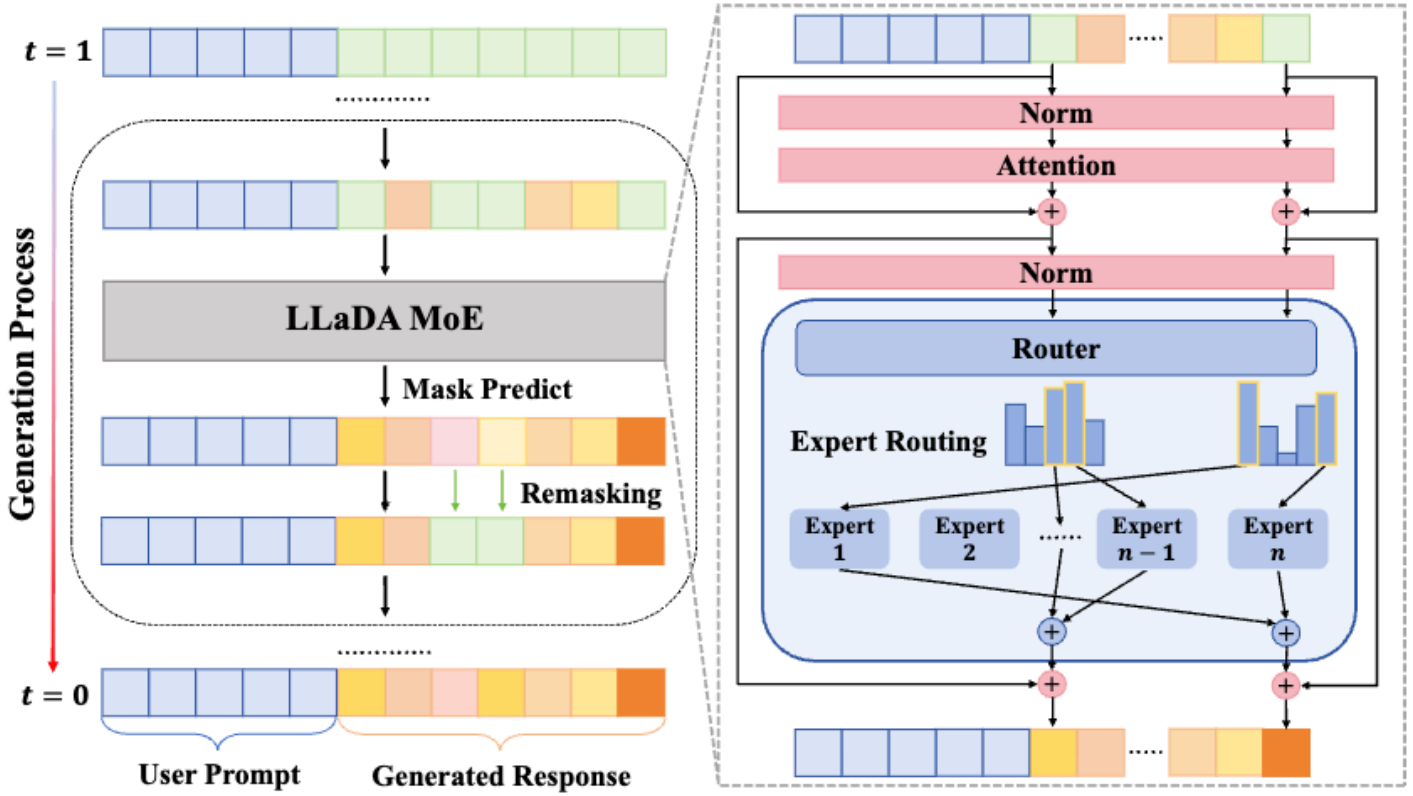
1. **확산 언어 모델(MDM) 중 SOTA 달성**
   : 총 7B 파라미터 중 단 1.4B 개의 활성 파라미터만 사용, 적은 추론 비용으로도 이전의 8B Dense MDM 의 성능을 능가

2. **타 모델과의 경쟁력 입증**

LLaDA-MOE

Architecture

Overview



LLaDA-MoE는 정규화를 위해 아래 항목 사용

1. RMSNorm
2. Activation Func: SwiGLU
3. ROPE
4. MHA 내에 QK-layernorm

Layers	16
Hidden Dimension	2048
Attention Heads	16
Total Experts	64
Activated Experts	8
Expert Dimension	1024
RoPE Base	50,000
Active Parameters	1.4B
Non-embedding Parameters	7B

LLaDA-MOE

Architecture

MoE Routing.

- 현재까지의 MDM 연구는 대부분 Dense한 트랜스포머 백본에 의존
- 대조적으로, AR 모델 분야에서는 Sparse Mixture-of-Experts 아키텍처가 널리 검증됨

Auxiliary Losses.

1) Load-Balancing Loss: Expertes가 공평하게 나누어 받도록 보장

$$\mathcal{L}_{\text{LB}} = N \sum_{i=1}^{N} f_i P_i$$

N. 전문가의 수
f_i. i번째 전문가가 모든 토큰에 걸쳐 선택된 빈도
P_i i번째 전문가에게 할당된 평균 라우팅 확률

2) Z-Loss: 라우터가 출력하는 로짓(z_t)값 자체가 너무 커지는 것을 방지하여 학습을 안정화

$$\mathcal{L}_{\text{Z}} = \frac{1}{T} \sum_{t=1}^{T} \left(\log \sum_{j=1}^{N} e^{z_{t,j}} \right)^2$$

T: number of tokens
z_{t,j}: Router(h_t) 가 t번째 토큰에 대해 j번째 E에게 부여한 logit

LLaDA-MOE

Training Pipeline

Figure 3: **Training pipeline.** LLaDA-MoE is trained through Pretrain stage 1 (10T tokens), pretrain stage 2 (10T tokens), annealing stage 1 (500B tokens), annealing stage 2 (500B tokens with 8k context length), followed by SFT on curated prompt–answer pairs.

Pretrain Stage

PT Stage 1

- 대규모 혼합 텍스트 코퍼스 (10T Tokens)

PT Stage 2

- sampling reweighted to increase the fraction of mathematics and code (+10T)

LLaDA-MOE

Pipeline

Annealing Stage

Annealing Stage 1

- 500B tokens of high-quality text
- 사전 학습 2단계의 최고 성능 체크포인트 사용

Annealing Stage 2

- 500B토큰
- RoPE Base 10,000에서 50,000으로 증가
- 컨텍스트 길이 4k에서 8k로 확장

SFT Stage

- high-quality question-answer pairs
- 다중 턴 대화(multi-turn dialogs)의 경우, 특정 턴의 응답에만 마스킹 커널을 적용
- SFT 중 샘플은 대부분 4k보다 짧아 최대 4k 토큰으로 제한하여 학습 (불필요한 |EOS| 토큰 생성 방지)

LLaDA-MOE

Training Pipeline

Forward Process

: 마스크가 추가된 입력 y_t를 보고 원본 토큰 y^i를 예측하도록 모델을 학습 (LLaDA와 동일)

Train-Test Discrepancy

- 실제 테스트 시에는 4k보다 훨씬 짧거나 다양한 길이의 입력이 들어옴

Variable-Length Training

- 99%: 기존과 동일하게 고정된 4k 컨텍스트 사용
- 1%의 : 8에서 4096 사이의 무작위 길이를 샘플링하여, 입력을 해당 길이로 truncate

LLaDA-MOE

Training Pipeline

Supervised Fine-tuning

: 답변 부분만 확률 t에 따라 마스킹하여, 원본 토큰 y^i를 예측하도록 모델을 학습 (LLaDA와 동일)

|EOS| 토큰 학습

: 문장 끝을 알리는 |EOS| 토큰도 답변의 일부로 취급하여 함께 마스킹하고 손실 계산에 포함

다중 턴 대화

: [q1, a1, q2]까지를 입력으로 보고, a2를 답변으로 취급하여 동일하게 학습

4k 컨텍스트 제한

: 어닐링 2단계(Annealing Stage 2) 동안 4k에서 8k로 확장되지만, SFT는 4k로 제한

LLaDA-MOE

Inference

추론은 모델이 완전히 마스크된 상태([M][M]...[M])에서 시작하여 점차 [M]을 실제 토큰으로 채워 나감

Train-Test Discrepancy

- 실제 테스트 시에는 4k보다 훨씬 짧거나 다양한 길이의 입력이 들어옴

Generation Strategy

: LLaDA와 동일한 전략 사용 가능

LLaDA-MOE

Experiments Results (base)

Knowledge, Reasoning, Coding, Math, Agent 등에 대한 벤치마크 평가 수행

: Semi-Autoregressive로 평가

: Generation length 1024

: block size 64

	LLaDA-MoE-7B-A1B-Base	LLaDA-8B-Base	Dream-v0-Base-7B	Qwen2.5-3B-Base
Architecture	MoE	Dense	Dense	Dense
Model	Diffusion	Diffusion	Diffusion	AR
Method	Pretrain	Pretrain	Continue Pretrain	Pretrain
# Total Params	7B	8B	7B	3B
# Activated Params	1B	8B	7B	3B
		General Tasks		
MMLU	64.59	65.90	**69.50**	67.98
MMLU-Pro	39.16	41.80	**48.15**	35.50
CEval	65.56	70.50	59.18	**75.00**
CMMLU	65.65	69.90	60.87	**73.65**
RACE	84.96	**88.37**	44.70	87.88
		Reasoning Tasks		
BBH	52.71	49.80	**57.90**	56.50
Drop	65.86	72.93	**75.16**	51.61
KorBench	31.20	33.68	**37.44**	27.44
		Math Tasks		
GSM8K	66.41	70.70	77.79	**78.17**
MATH	36.10	27.30	39.60	**40.94**
OlympiadBench	10.07	6.85	**10.22**	9.33
		Coding Tasks		
CRUX-O	**39.00**	31.00	37.75	35.62
MBPP	52.40	38.20	56.20	**69.56**
MultiPL-E	**41.13**	23.61	27.60	40.80
HumanEval	45.73	33.50	57.90	**57.93**
LiveCodeBench v6	16.18	2.53	14.87	**16.99**
BigCodeBench-Full	21.23	13.42	18.33	**30.88**
Avg	46.94	43.53	46.66	**50.34**

LLaDA-MOE

Experiments Results (Instruct)

Knowledge, Reasoning, Coding, Math, Agent 등에 대한 벤치마크 평가 수행

: Semi-Autoregressive로 평가

: Generation length 1024

: block size 64

	LLaDA-MoE-7B-A1B-Instruct	LLaDA-8B-Instruct	LLaDA-1.5	Dream-v0-Instruct-7B	Qwen2.5-3B-Instruct
Architecture	MoE	Dense	Dense	Dense	Dense
Model	Diffusion	Diffusion	Diffusion	Diffusion	AR
Method	Pretrain + SFT	Pretrain + SFT	Pretrain + SFT + DPO	Continue Pretrain + SFT	Pretrain + SFT + RL
# Total Params	7B	8B	8B	7B	3B
# Activated Params	1B	8B	8B	7B	3B
General Tasks					
MMLU	67.18	65.50	66.00	67.00	**69.11**
MMLU-Pro	**44.64**	37.00	35.70	43.30	44.13
CMMLU	64.30	55.21	58.72	58.82	**65.62**
CEval	63.93	54.48	58.41	57.98	**68.20**
Reasoning Tasks					
Drop	79.77	83.09	**84.89**	76.25	68.56
KorBench	**38.40**	33.68	37.20	32.56	36.88
Math Tasks					
GSM8K	82.41	78.60	83.30	81.00	**86.28**
MATH	58.68	42.20	42.60	39.20	**67.02**
OlympiadBench	21.04	10.52	10.96	10.44	**30.41**
Coding Tasks					
CRUX-O	42.38	28.50	29.12	40.12	**46.75**
MBPP	**70.02**	41.00	42.80	58.80	65.81
MultiPL-E	52.53	29.08	29.04	29.86	**54.92**
HumanEval	**61.59**	49.40	52.40	55.50	60.37
LiveCodeBench v6	**13.27**	6.66	6.94	5.23	9.20
BigCodeBench-Full	20.44	11.32	11.93	19.04	**27.81**
Agent & Alignment Tasks					
IFEval Strict Prompt	59.33	51.39	58.23	**62.50**	58.20
BFCL-Live	63.09	47.47	**66.20**	53.03	50.40
Avg	53.12	42.65	45.56	46.51	**53.51**

Q & A