arXiv:2606.25331v1 [cs.CL] 24 Jun 2026

ByteDance | Seed

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

Improved Large Language Diffusion Models

Shen Nie[1,2,3,†], **Qiyang Min**[4], **Shaoxuan Xu**[1,2,3], **Zihao Huang**[4], **Yuxuan Song**[4], **Yong Shan**[4], **Yankai Lin**[1,2,3], **Wayne Xin Zhao**[1,2,3], **Chongxuan Li**[1,2,3,*], **Ji-Rong Wen**[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]ByteDance Seed

[†]Work done during an internship at ByteDance Seed, [*]Corresponding authors

Abstract

Modern large language models are predominantly trained with autoregressive factorization and causal attention. We present iLLaDA, an 8B masked diffusion language model trained from scratch with fully bidirectional attention. iLLaDA keeps the masked diffusion objective throughout pre-training and supervised fine-tuning (SFT), scaling pre-training to 12T tokens and fine-tuning on a 25B-token instruction corpus for 12 epochs. We further use variable-length generation for efficiency and introduce confidence-based scoring for multiple-choice evaluation. Compared with LLaDA, iLLaDA improves broadly across general, mathematical, and code benchmarks; for example, iLLaDA-Base improves by 21.6 points on BBH and 14.9 points on ARC-Challenge, while iLLaDA-Instruct improves by 14.5 points on MATH and 16.5 points on HumanEval. Despite its non-autoregressive training, iLLaDA also remains competitive with Qwen2.5 7B on several benchmarks. These results show that fully bidirectional diffusion training from scratch is a competitive path toward strong language models. Model weights and codes: https://github.com/ML-GSAI/LLaDA.

Contact: {nieshen,chongxuanli}@ruc.edu.cn
Correspondence: Chongxuan Li and Ji-Rong Wen

1 Introduction

Large language models (LLMs) are currently dominated by the autoregressive paradigm [1–4]. Recently, diffusion language models have attracted increasing attention as a different approach to language generation. Following the masked diffusion formulation [5–9], LLaDA trains a language model from scratch with fully bidirectional attention [10]. It shows that a non-autoregressive model can acquire core LLM capabilities such as in-context learning and instruction-following, challenging the common belief that language intelligence must rely on autoregressive modeling.

Beyond this conceptual implication, bidirectional diffusion language models have shown advantages in reversal and bidirectional reasoning [10, 11], long-horizon planning [12], and multimodal or omni-modeling [13–16]. Recent studies further show that bidirectional diffusion pre-training can better exploit limited data under repeated training, enabling diffusion language models to outperform autoregressive models in data-constrained settings [17, 18]. However, LLaDA was still an initial large-scale attempt, and its performance remained behind strong autoregressive models such as Qwen2 [19] and Qwen2.5 [20], leaving substantial room to improve.

We introduce iLLaDA (improved LLaDA), an 8B fully bidirectional masked diffusion language model trained from scratch. For pre-training, iLLaDA scales the corpus to 12T tokens, uses grouped-query attention [21] to reduce cache-style inference memory and tied input/output embeddings to reduce parameter count, and modifies the learning-rate schedule for large-scale training. For post-training, iLLaDA modifies the SFT strategy for variable-length generation and trains on a 25B-token instruction corpus for 12 epochs. For inference and evaluation, iLLaDA uses variable-length generation for efficiency and confidence-based scoring for multiple-choice benchmarks.

Experiments show that these changes substantially improve LLaDA. Compared with previous bidirectional diffusion language models, including LLaDA trained from scratch and Dream fine-tuned from Qwen2.5 [22], iLLaDA obtains the best average performance in both base and instruction-tuned evaluations. Against Qwen2.5 7B [20], iLLaDA-Base is slightly stronger on average, while iLLaDA-Instruct still lags behind Qwen2.5 7B Instruct. Ablations further show that confidence-based scoring improves multiple-choice evaluation and that iLLaDA continues to benefit from SFT over multiple epochs.

2 Approach

This section describes the training and inference procedures of iLLaDA. We keep the masked diffusion formulation of LLaDA [5–10], while making several practical changes that are important for scaling, post-training, and evaluation.

2.1 Pre-training

iLLaDA follows the same pre-training objective as LLaDA. Given a clean sequence x_0 of length L, we sample a masking ratio $t \sim U[0, 1]$, independently replace each token by the mask token M with probability t, and obtain a corrupted sequence x_t. The model is trained to predict all masked tokens:

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

This objective is a likelihood-based masked diffusion objective for discrete data [7–9], where the indicator function $\mathbf{1}[\cdot]$ ensures that the loss is computed only for masked tokens. It differs from masked language modeling with a fixed masking ratio [23].

The backbone of iLLaDA is a dense Transformer, which uses RMSNorm [24], SwiGLU [25], RoPE [26], and no attention or MLP bias. In contrast to LLaDA, which uses multi-head attention, iLLaDA uses grouped-query attention (GQA) [21]. Recent work has shown that KV-cache-like mechanisms can be adapted to diffusion language models [27–31]; under such cache-style implementations, GQA reduces the memory footprint of cached key/value states. To further control the parameter count, iLLaDA ties the input embedding and LM-head parameters. The architectural differences between iLLaDA and LLaDA are summarized in Tab. 1.

We pre-train iLLaDA with maximum sequence length 8192. We randomly split an 8192-token sequence into two shorter segments with probability 30%, inspired by random-length training for masked diffusion language models [11]. We pack variable-length examples in each batch and compute attention with a FlashAttention-based variable-length attention kernel[1], which uses cumulative sequence offsets to separate examples without padding them to a common length. The learning rate is linearly warmed up to 2×10^{-4} and then kept constant. During training, when we observed that the pretraining loss stopped decreasing, we switched to a cosine decay schedule with minimum learning rate 5×10^{-6}, after which the pretraining loss continued to improve. We use the AdamW optimizer [32] with weight decay 0.1.

2.2 Supervised Fine-Tuning

Prior works [10, 13, 33] typically construct each SFT instance by concatenating a prompt with its full reference response. During training, the prompt tokens are kept visible, while masks are applied only within the

[1] https://docs.pytorch.org/docs/2.12/nn.attention.varlen.html

Table 1 Architecture comparison between iLLaDA and LLaDA.

	iLLaDA 8B	LLaDA 8B
Layers	32	32
Model dimension	4096	4096
Attention heads	32	32
Key/Value heads	8	32
FFN dimension	14,336	12,288
Vocabulary size	155,136	126,464
Maximum sequence length	8192	4096
Embedding and LM-head	Tied	Untied
Total parameters	7.62B	8.02B
Non-embedding parameters	6.98B	6.98B

response region. Within each mini-batch, shorter responses are padded with |EOS| tokens to match the length of the longest response.

iLLaDA instead uses the same data processing and masking scheme as pre-training. We format each instruction example as a prompt-response sequence followed by a single terminal |EOS|, concatenate all formatted examples into a continuous instruction corpus, and sample 8192-token training sequences from this corpus. We then apply random masks to the entire sequence and optimize Eq. (1), so prompt tokens, response tokens, and |EOS| tokens may all be masked. We also use the same random-length training as in pre-training. This SFT format naturally supports the variable-length block generation described in Sec. 2.3.

Our SFT corpus contains approximately 25 billion tokens, and we fine-tune for 12 epochs. As shown in Sec. 3.2, our ablation study shows that iLLaDA continues to improve as the number of SFT epochs increases. During SFT, the learning rate is first linearly warmed up to 5×10^{-6}, then kept constant, and finally linearly decayed to 5×10^{-7} over the last 10% of training. We use the AdamW optimizer [32] with weight decay 0.1.

2.3 Inference

iLLaDA uses the same probabilistic formulation as LLaDA [10]. In particular, both models are trained with the masked diffusion objective in Eq. (1), which corresponds to an upper bound on the negative log-likelihood of the model distribution.

Many language-model benchmarks are formulated as multiple-choice tasks, such as HellaSwag [34], PIQA [35], and ARC-Challenge [36]. Given a prefix p and a finite set of candidate continuations, evaluation requires assigning a score to each candidate and selecting the highest-scoring one. For iLLaDA, we use a deterministic confidence-based scoring rule, which performs better empirically than the upper-bound of log-likelihood on multiple-choice tasks.

Given a candidate continuation y of length L, we start from an all-masked candidate and repeatedly reveal one ground-truth candidate token. At step k, among the remaining masked positions $\mathcal{M}_{k-1}$, we choose the token that the model assigns the highest confidence to:

$$i_k = \arg\max_{i \in \mathcal{M}_{k-1}} p_\theta(y^i \mid p, \tilde{y}_{k-1}), \quad S_{\text{conf}}(y \mid p) = \sum_{k=1}^{L} \log p_\theta(y^{i_k} \mid p, \tilde{y}_{k-1}), \tag{2}$$

where $\tilde{y}_{k-1}$ contains the revealed ground-truth tokens and masks elsewhere. This confidence score is not a likelihood estimate; rather, it is a task-specific scoring surrogate for comparing a finite set of candidate answers.

For open-ended generation, iLLaDA uses variable-length generation. Given a prompt, we append a block of mask tokens and run the diffusion sampler within this block. At each sampling step, the model predicts all masked positions, and we transfer the most confident predictions to visible tokens while keeping low-confidence

Table 2 Benchmark Results of Base Models. Results marked by † and ‡ are from Nie et al. [10] and Ye et al. [22], respectively. For Dream, 18T denotes Qwen2.5 pre-training tokens and 0.6T denotes diffusion fine-tuning tokens.

	iLLaDA 8B	LLaDA 8B†	Dream 7B‡	Qwen2.5 7B‡
Model	Diffusion	Diffusion	Diffusion	AR
Training tokens	12T	2.3T	18T + 0.6T	18T
General Tasks				
MMLU	**74.8**	65.9	69.5	71.9
BBH	**71.3**	49.7	57.9	63.9
ARC-C	**60.8**	45.9	59.8	51.5
Hellaswag	76.6	70.5	73.3	**79.0**
Mathematics & Science				
GSM8K	**81.9**	70.3	77.2	78.9
Math	38.4	31.4	39.6	**41.1**
Code				
HumanEval	50.0	35.4	**57.9**	56.7
MBPP	57.8	40.0	56.2	**63.6**
Average	**63.9**	51.1	61.4	63.3

Table 3 Benchmark Results of Instruct Models. Results marked by † and ‡ are from Nie et al. [10] and Ye et al. [22], respectively.

	iLLaDA 8B	LLaDA 8B†	Dream 7B‡	Qwen2.5 7B‡
Model	Diffusion	Diffusion	Diffusion	AR
General Tasks				
MMLU	71.6	65.5	67.0	**76.6**
MMLU-Pro	52.3	37.0	43.3	**56.3**
MMLU-Redux	**76.4**	68.9	76.3	75.7
Mathematics & Science				
GSM8K	89.0	77.5	81.0	**91.6**
Math	56.7	42.2	39.2	**75.5**
Code				
HumanEval	65.9	49.4	55.5	**84.8**
MBPP	58.0	41.0	58.8	**79.2**
Avg.	67.1	54.5	60.2	**77.1**

positions masked, following the low-confidence remasking strategy of LLaDA and MaskGIT [37]. Once a block is decoded, generation terminates if an |EOS| or other stop token appears; otherwise, a new block of masks is appended and the process continues until a maximum generation budget is reached.

3 Experiments

In this section, we evaluate the base and instruction-following capabilities of iLLaDA on standard benchmarks, followed by ablation studies on multiple-choice scoring and SFT duration. The results show that iLLaDA substantially improves over prior diffusion language models and remains competitive with strong autoregressive baselines on several reasoning benchmarks.

Table 4 Ablation Results of Multiple-Choice Scoring Rules.

Scoring rule	PIQA	ARC-C	Hellaswag
Likelihood	77.2	60.2	74.3
Confidence	**78.5**	**60.8**	**76.6**

3.1 Benchmark Results

We evaluate iLLaDA in both base and instruction-tuned settings. The benchmark suite covers general language understanding and reasoning, including MMLU [38], BBH [39], ARC-Challenge [36], and HellaSwag [34]; mathematical reasoning, including GSM8K [40] and MATH [41]; and code generation, including HumanEval [42] and MBPP [43]. For instruction-tuned models, we additionally report MMLU-Pro [44], a more challenging multi-task understanding benchmark, and MMLU-Redux [45], an error-corrected re-annotation of MMLU. We compare with representative diffusion language models, LLaDA 8B [10] and Dream 7B [22], as well as the autoregressive Qwen2.5 7B [20].

Tab. 2 compares base models. iLLaDA substantially improves over LLaDA across all tasks, with particularly large gains on BBH, ARC-Challenge, GSM8K, HumanEval, and MBPP. Compared with Dream 7B, iLLaDA achieves stronger results on most general and mathematical benchmarks, while Dream remains stronger on HumanEval. Against Qwen2.5 7B, iLLaDA is competitive despite using a diffusion formulation, and obtains the best results on MMLU, BBH, ARC-Challenge, and GSM8K among the models reported in the table.

Tab. 3 reports instruction-tuned results. iLLaDA continues to outperform LLaDA and Dream on most benchmarks after SFT, and the improvements are especially pronounced on GSM8K, MATH, and HumanEval. Compared with Qwen2.5 7B, iLLaDA remains behind on several math and code benchmarks, but achieves competitive results on MMLU-Redux and substantially narrows the gap between diffusion language models and strong autoregressive baselines. Since iLLaDA Base is already competitive with Qwen2.5 Base in Tab. 2, we believe the remaining gap in the instruct setting is largely due to the additional reinforcement-learning alignment used by Qwen2.5 after SFT. We leave reinforcement-learning alignment for iLLaDA to future work. Please refer to Appendix A for evaluation details.

3.2 Ablation Studies

We first ablate the scoring rule for multiple-choice evaluation. As shown in Tab. 4, confidence-based scoring consistently improves over the likelihood-style multiple-choice baseline, with gains of 1.3 on PIQA, 0.6 on ARC-Challenge, and 2.3 on HellaSwag. This result motivates the use of confidence-based scoring for the multiple-choice evaluations in Sec. 2.3.

We further study the effect of SFT duration. Fig. 1 shows that performance generally improves as the number of SFT epochs increases, supporting the use of long SFT for iLLaDA, especially on reasoning-heavy benchmarks. This observation is consistent with recent studies of diffusion language models in data-constrained regimes [17, 18]; for example, Ni et al. [18] show that diffusion language models can continue to improve under extreme repeated pre-training settings, such as training on 1B unique tokens for 96 epochs. Our results suggest that a similar data-reuse effect also appears in SFT, where the instruction corpus is much smaller than the pre-training corpus and thus more relevant to practical instruction tuning. Due to compute constraints, we did not train beyond 12 SFT epochs.

4 Conclusion and Discussion

We present iLLaDA, an 8B fully bidirectional diffusion language model trained from scratch. iLLaDA scales pre-training to 12T tokens and updates several parts of the practical recipe, including the model design, learning-rate schedule, SFT format, confidence-based multiple-choice scoring, and variable-length generation. We also find that iLLaDA continues to benefit from SFT over multiple epochs. Across base and instruction-tuned evaluations, these changes lead to substantial improvements over LLaDA on general, mathematical,

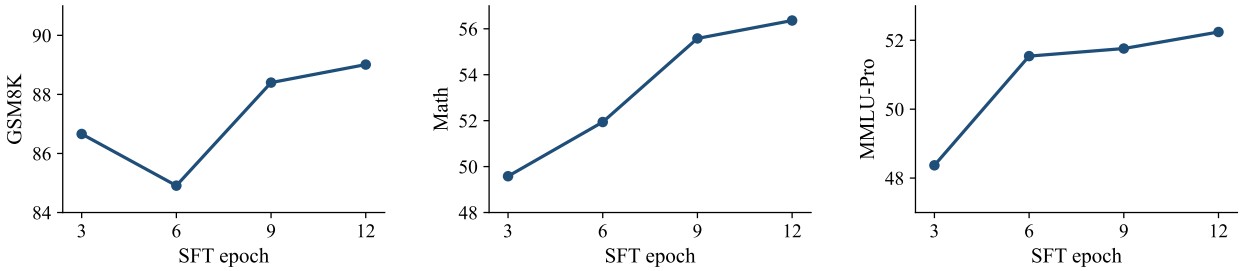

Figure 1 SFT epoch ablation. We evaluate iLLaDA at different SFT epochs on GSM8K, MATH, and MMLU-Pro.

and code benchmarks, suggesting that fully bidirectional diffusion training from scratch can achieve strong language modeling performance.

This report also leaves several limitations. First, iLLaDA has not been further aligned with reinforcement learning, which may partly explain the remaining gap between iLLaDA-Instruct and strong autoregressive instruct models. Recent RL methods developed for masked diffusion LLMs, such as VRPO, diffu-GRPO, MDPO, and ESPO [33, 46–48], can be directly applied to iLLaDA and are likely to further improve its instruction-following and reasoning abilities. Second, due to limited compute, our study is limited to the 8B scale and does not provide a fully matched comparison with autoregressive models; instead, we allocate our compute to 12T-token pre-training. We leave reinforcement-learning alignment and larger-scale studies for future work.

References

[1] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. arXiv preprint arXiv:2303.18223, 2023.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

[3] OpenAI. ChatGPT: Optimizing Language Models for Dialogue. OpenAI blog, November 2022. URL https://openai.com/blog/chatgpt/.

[4] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

[5] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. Advances in Neural Information Processing Systems, 34:17981–17993, 2021.

[6] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. arXiv preprint arXiv:2310.16834, 2023.

[7] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. arXiv preprint arXiv:2406.04329, 2024.

[8] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. arXiv preprint arXiv:2406.07524, 2024.

[9] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. arXiv preprint arXiv:2406.03736, 2024.

[10] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen,

and Chongxuan Li. Large language diffusion models. Advances in Neural Information Processing Systems, 38: 50608–50646, 2026.

[11] Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. arXiv preprint arXiv:2410.18514, 2024.

[12] Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. In International Conference on Learning Representations, 2025.

[13] Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. arXiv preprint arXiv:2505.16933, 2025.

[14] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. arXiv preprint arXiv:2505.15809, 2025.

[15] Zebin You, Xiaolu Zhang, Jun Zhou, Chongxuan Li, and Ji-Rong Wen. Llada-o: An effective and length-adaptive omni diffusion model. arXiv preprint arXiv:2603.01068, 2026.

[16] Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, et al. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding. arXiv preprint arXiv:2510.06308, 2025.

[17] Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. arXiv preprint arXiv:2507.15857, 2025.

[18] Jinjie Ni, Qian Liu, Longxu Dou, Chao Du, Zili Wang, Hang Yan, Tianyu Pang, and Michael Qizhe Shieh. Diffusion language models are super data learners. arXiv preprint arXiv:2511.03276, 2025.

[19] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

[20] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. arXiv preprint arXiv:2412.15115, 2024.

[21] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 4895–4901, 2023.

[22] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. arXiv preprint arXiv:2508.15487, 2025.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[24] Biao Zhang and Rico Sennrich. Root mean square layer normalization. Advances in Neural Information Processing Systems, 32, 2019.

[25] Noam Shazeer. Glu variants improve transformer. arXiv preprint arXiv:2002.05202, 2020.

[26] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 568:127063, 2024.

[27] Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dkv-cache: The cache for diffusion language models. arXiv preprint arXiv:2505.15781, 2025.

[28] Quan Nguyen-Tri, Mukul Ranjan, and Zhiqiang Shen. Attention is all you need for kv cache in diffusion llms. arXiv preprint arXiv:2510.14973, 2025.

[29] Minsoo Cheong, Donghyun Son, Woosang Lim, and Sungjoo Yoo. Entropycache: Decoded token entropy guided kv caching for diffusion language models. arXiv preprint arXiv:2603.18489, 2026.

[30] Yicun Yang, Cong Wang, Shaobo Wang, Zichen Wen, Biqing Qi, Hanlin Xu, and Linfeng Zhang. Diffusion llm with native variable generation lengths: Let [eos] lead the way. arXiv preprint arXiv:2510.24605, 2025.

[31] Yu-Yang Qian, Junda Su, Lanxiang Hu, Peiyuan Zhang, Zhijie Deng, Peng Zhao, and Hao Zhang. d3llm: Ultra-fast diffusion llm using pseudo-trajectory distillation. arXiv preprint arXiv:2601.07568, 2026.

[32] I Loshchilov. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

[33] Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. arXiv preprint arXiv:2505.19223, 2025.

[34] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830, 2019.

[35] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In Proceedings of the AAAI conference on artificial intelligence, 2020.

[36] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457, 2018.

[37] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11315–11325, 2022.

[38] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300, 2020.

[39] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. arXiv preprint arXiv:2210.09261, 2022.

[40] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

[41] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874, 2021.

[42] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021.

[43] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. arXiv preprint arXiv:2108.07732, 2021.

[44] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024.

[45] Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. Are we done with mmlu? arXiv preprint arXiv:2406.04127, 2024.

[46] Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. arXiv preprint arXiv:2504.12216, 2025.

[47] Haoyu He, Katrin Renz, Yong Cao, and Andreas Geiger. MDPO: Overcoming the training-inference divide of masked diffusion language models. arXiv preprint arXiv:2508.13148, 2025.

[48] Jingyang Ou, Jiaqi Han, Minkai Xu, Shaoxuan Xu, Jianwen Xie, Stefano Ermon, Yi Wu, and Chongxuan Li. Principled rl for diffusion llms emerges from a sequence-level perspective. arXiv preprint arXiv:2512.03759, 2025.

A Evaluation Details

This appendix provides additional details for the evaluations in Sec. 3.

For iLLaDA-8B-Base, we use open-ended generation for BBH, GSM8K, MATH, HumanEval, and MBPP. For BBH, GSM8K, MATH, and MBPP, we set the maximum generation length to 1024 and the block length to 32. For HumanEval, we set both the maximum generation length and the block length to 512, since we observed that semi-autoregressive block sampling hurts performance on this benchmark.

For iLLaDA-8B-Instruct, we use benchmark-specific inference settings. For MMLU and MMLU-Redux, where the model only needs to generate a single answer letter, we set the maximum generation length and block length to 4/4 and 3/3, respectively. For GSM8K and HumanEval, we set the maximum generation length to 2048 and the block length to 32. For MMLU-Pro and MATH, we set the maximum generation length to 4096 and the block length to 32. For MBPP, we set the maximum generation length to 2048 and the block length to 16.

For iLLaDA-8B-Instruct, we observe repetitive reasoning loops on some difficult problems, where the model may repeatedly produce phrases such as "Wait, let me check again" and fail to produce a final answer. We attribute this behavior to a subset of the SFT corpus that contains structured chain-of-thought traces generated by reasoning models. To mitigate such loops, as generation becomes longer, we gradually increase the probability of emitting the stop-thinking token `</think>`, encouraging the model to terminate the reasoning trace and produce the final answer.