



Training Diffusion Language Models at Scale using Autoregressive Models

Radical Numerics Inc.

Abstract

Diffusion Language Models (DLMs) enable parallel, any-order generation, offering new opportunities for inference-time optimization compared to autoregressive models. Despite these advantages, recipes for training DLMs at scale remain underexplored. We introduce RND1-Base, a general-purpose 30B-parameter sparse mixture-of-experts DLM trained with a simple and scalable continual pretraining recipe. Following a simple *autoregressive-to-diffusion* (A2D) conversion recipe, we continually pretrain an autoregressive base model on 500B tokens to obtain a high-capacity DLM. RND1-Base achieves state-of-the-art performance among general-purpose DLMs on common sense / reasoning (e.g., MMLU: 69.6%), STEM (e.g., GSM8K: 80.0%), and coding (e.g., MBPP: 65.4%) benchmarks. To our knowledge, this is the first open effort to scale DLMs beyond 8B parameters. We release RND1-Base and our recipe to catalyze research in post-training, inference, and architectural innovation in DLMs. Model weights, inference code, and samples are available.

Code: <https://github.com/RadicalNumerics/RND1>

Correspondence: research@radicalnumerics.ai

Date: October 9, 2025

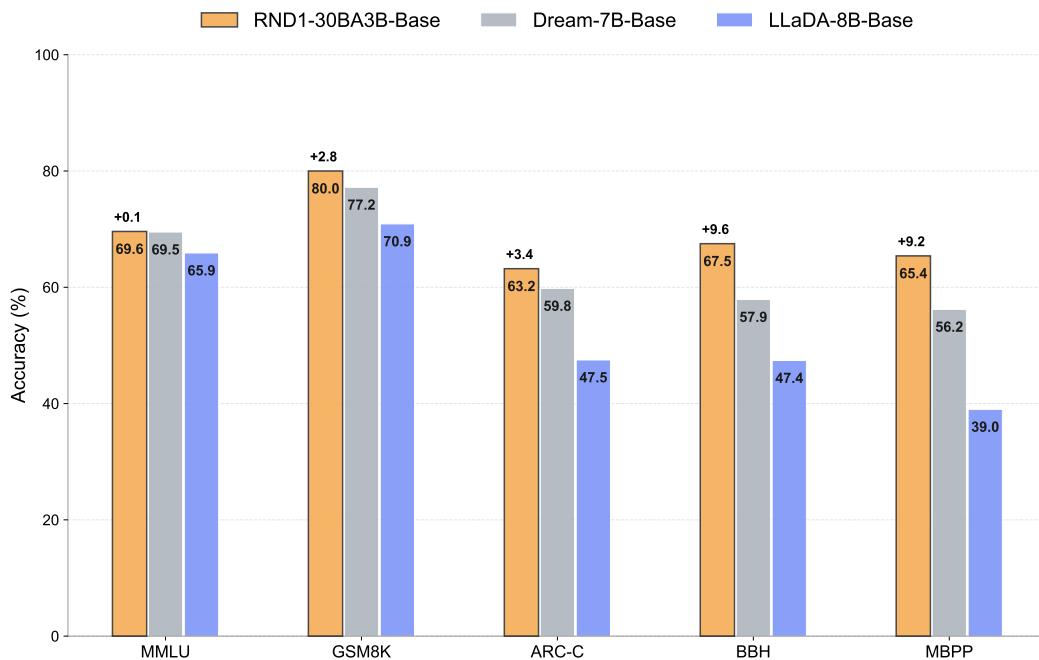


Figure 1: **Evaluation.** Benchmark performance comparison of diffusion language models (DLMs): Dream-7B Base (Ye et al., 2025), LLaDA-8B-Base (Nie et al., 2025), and RND1-Base (Ours).

1 Introduction

Autoregressive (AR) modeling has been the dominant paradigm for large-scale language modeling over the past decade. AR models generate text sequentially in a fixed left-to-right order, producing one token per function evaluation. Recently, Diffusion Language Models (DLMs) have emerged as an alternative, enabling parallel, flexible-order text generation and promising advances in the quality–efficiency frontier (Austin et al., 2021a; Lou et al., 2024; Sahoo et al., 2024; He et al., 2023). However, recipes for training DLMs at scale remain largely unexplored.

A key challenge in training high-quality DLMs is their relative scaling inefficiency (Prabhudesai et al., 2025) e.g., direct DLM training requires more passes over a finite dataset to outperform direct AR training. Moreover, AR models benefit from more mature infrastructure and well-understood training recipes.

To leverage the advantages of AR training for DLM scaling, we propose a simple continual pretraining recipe to obtain large-scale DLMs from pretrained AR models, a procedure referred to as AR-to-Diffusion (A2D) conversion. The connection is straightforward: AR models learn to predict next-token in a fixed left-to-right order, while DLMs generalize this to predicting multiple tokens in flexible orders. This makes AR models a strong initialization for DLM training. A2D can be viewed as customizing the training objective—specifically, by learning multiple token orderings by leveraging the structure of autoregressive pretraining to trace an efficient path from AR pretraining to a DLM. A2D conversion presents two key challenges: **(1) How to endow DLMs with bidirectional context capabilities during A2D conversion?** AR models use causal attention that restricts context to preceding tokens, whereas DLMs can leverage bidirectional context for sequence modeling. **(2) How to retain AR pretraining knowledge during A2D conversion?** AR models are trained on trillions of text tokens—encoding broad world knowledge—which must be preserved during the conversion. Prior small-scale works address the first challenge using attention-mask annealing (Gong et al., 2025; Ye et al., 2025), which requires design decisions such as mask transition policies and annealing schedules. Further, AR knowledge retention during A2D conversion has received little attention. In this work, we propose a *simple single-stage continual pretraining recipe* that directly addresses both challenges: (1) we replace the causal mask with a bidirectional mask at initialization, avoiding design choices associated with attention mask annealing techniques, and (2) we restrict updates to dense layers—specifically, Mixture-of-Experts (MoE) layers—during A2D conversion to preserve AR model pretraining knowledge.

We introduce RND1-Base, a general-purpose 30B-parameter DLM trained on 500B tokens using our proposed A2D recipe. RND1-Base achieves state-of-the-art performance among general-purpose DLMs on benchmarks spanning common sense/reasoning (e.g., MMLU (Hendrycks et al., 2021): 69.6% , BBH (Suzgun et al., 2023): 67.5%), STEM (e.g., GSM8K (Cobbe et al., 2021): 80.0%), and coding (MBPP (Austin et al., 2021b): 65.4%). We release RND1-Base and our A2D methods to provide a foundation for future research on training, inference, and architectural design for scalable diffusion language modeling.

2 Preliminaries

Let $x = (x_1, \dots, x_L)$ denote a sequence of L tokens drawn from a vocabulary \mathcal{V} . We write $x_{<j} = (x_1, \dots, x_{j-1})$ for the prefix up to position $j - 1$. The vocabulary \mathcal{V} includes a designated [MASK] token used in masked diffusion training.

Autoregressive models. Autoregressive (AR) language models predict each token given its left context. The training objective is the standard next-token prediction loss $\mathcal{L}(\theta) = -\mathbb{E}_x \left[\sum_{j=1}^L \log p_\theta(x_j \mid x_{<j}) \right]$, corresponding to a left-to-right factorization of the sequence probability.

Masked diffusion language models. We focus on masked diffusion language models, a form of discrete diffusion that uses an absorbing transition kernel. Recently, masked diffusion has proven to be an effective formulation among diffusion-based language modeling approaches (Amin et al., 2025). Masked diffusion language models (MDLMs) treat generation as iterative denoising. At each step, a random subset of tokens is masked, and

the model is trained to predict them given the unmasked tokens. The training objective can be written as:

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

Here, x_0 is the clean sequence, x_t the corrupted sequence at step t , and the indicator $\mathbf{1}[x_t^i = \text{[MASK]}]$ selects the masked positions to be predicted.

Connection. AR models learn from a single fixed ordering (left-to-right), while MDLMs are trained over a distribution of random orderings. This motivates adapting pretrained AR models into MDLMs through continual pretraining.

Continual pretraining. We now define the continual pretraining objective that adapts an AR model into a masked diffusion language model. The continual pretraining loss applies a right-shifted next-token prediction rule: token x_0^i is predicted only if the next token x_t^{i+1} was masked in the corrupted sequence. Formally,

$$\mathcal{L}_{\text{A2D}}(\theta) = -\mathbb{E}_{t, x_0, x_t} \left[\frac{1}{t} \sum_{i=1}^{L-1} \mathbf{1}[x_t^{i+1} = \text{[MASK]}] \log p_{\theta}(x_0^i | x_t) \right]. \quad (2)$$

This objective is identical in form to the MDLM denoising loss.

3 Simple Continual Pretraining (SCP) for A2D conversion

We obtain diffusion language models (DLMs) beyond 8B parameters with a simple, scalable, single-stage continual pretraining recipe. Our approach begins from an AR model and transitions directly to DLM training, avoiding additional stages such as attention annealing. Central to our approach is the adaptation of training science techniques (e.g., *critical batch size analysis*, *layer-specific learning rates*) to bring DLM training recipes closer in sophistication to those developed for AR models.

To address the first challenge—endowing DLMs with bidirectional context during A2D conversion—we compare A2D recipes and examine their training dynamics.

3.1 Experiment Setup

Datasets, evaluation, and AR base model. We train on FineWeb-Edu (Lozhkov et al., 2024) and report accuracy on ARC-Easy (ARC-E), ARC-Challenge (ARC-C) (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), and PiQA (Bisk et al., 2020). We conduct experiments at the 4B-parameter scale, using Qwen3-4B (Bai et al., 2023) as the AR base model for A2D conversion.

Training settings. We use a global batch size of 2M tokens and train models using AdamW (Loshchilov & Hutter, 2017). The learning rate follows linear warmup for 2% of training and then remains constant at a peak value of 3e-4. We apply weight decay of 1e-4 and train models for 20B tokens.

3.2 Bidirectional Context for A2D conversion

We compare three strategies to equip an AR model with bidirectional context for A2D conversion:

1. **Random initialization:** Train a DLM from scratch.
2. **Grafting:** Starting from an AR model, train a *causal* DLM (causal mask) using limited data, graft bidirectional MHA operators following Chandrasegaran et al. (2025), then continue pretraining.
3. **Simple Continual Pretraining (SCP):** Starting from an AR model, set the attention mask to bidirectional at initialization, then continue pretraining with learning rate warmup.

Results. At initialization (0B tokens), the *random* model has the highest training loss; *grafting* starts with a lower training loss than *SCP* because it has already undergone DLM training; *SCP* begins between the two. After 20B tokens, the *random* model exhibits higher training loss, whereas *grafting* and *SCP* exhibit comparable training losses and similar accuracy on five benchmarks. Since *grafting* and *SCP* yield similar performance at 20B tokens, and *grafting* requires causal-DLM pretraining and operator-replacement steps, we adopt **SCP** as our default A2D recipe.

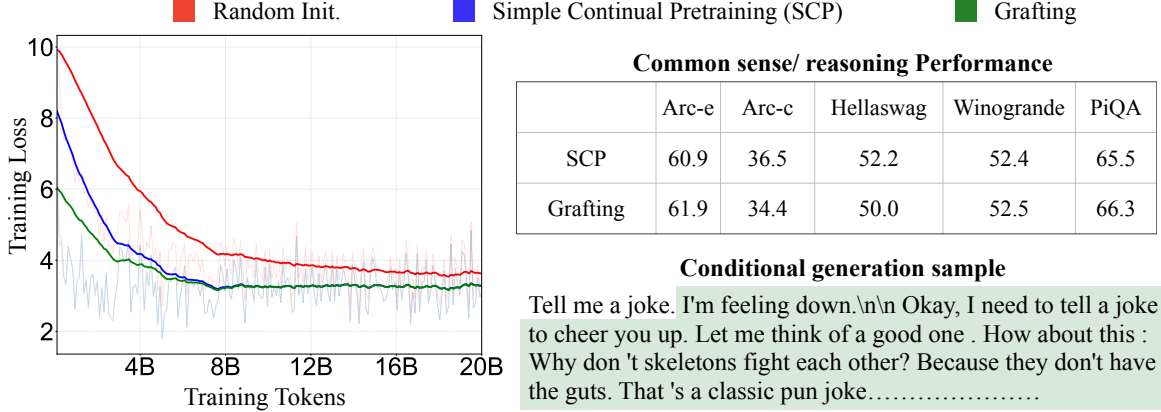


Figure 3.1: **Probing A2D recipes at 4B.** *Setup:* Qwen3-4B AR base; 20B FineWeb-Edu tokens; global batch size: 1024; peak LR: 3×10^{-4} with 2% warmup (then constant); weight decay: 10^{-4} . ① Training loss vs. tokens for *Random* (from scratch), *Grafting* (train causal DLM on limited data, graft bidirectional MHA per Chandrasegaran et al. (2025), then continual training), and *SCP* (continual training of AR model with a bidirectional mask from initialization). ② Zero-shot accuracy on ARC-E/C, HellaSwag, WinoGrande and PIQA. ③ Conditional generation from the SCP-trained DLM. **Key takeaway:** At 20B tokens, *SCP* \approx *Grafting*; both outperform from-scratch. Best viewed in color.

3.3 Critical Batch Size (CBS) Estimation for A2D Conversion

Masked diffusion training provides less supervision per batch than autoregressive training, since only masked positions contribute to the DLM loss (Nie et al., 2024; Prabhudesai et al., 2025). Under the standard DLM objective, the expected fraction of supervised tokens per sequence is $\approx 50\%$. As a result, batch-size and learning-rate heuristics tuned for AR models do not necessarily transfer to DLMs.

We estimate the *critical batch size (CBS)*—the batch size threshold beyond which greater data parallelism leads to diminishing returns characterized by training and validation loss—using *branched training* (McCandlish et al., 2018; Merrill et al., 2025).

Experimental settings. We start from a checkpoint that was trained for 60B tokens with the SCP recipe described above, using Qwen3-4B as the AR base model, and branch training into four runs that differ only in the *effective global batch size* $B \in \{1M, 2M, 4M, 8M\}$ tokens. We accordingly scale learning rate as $\eta(B) = \eta_0 \sqrt{\frac{B}{B_0}}$, keep optimizer hyperparameters and weight decay fixed, and align warmup (1%) and decay *in token space* across branches. Each branch trains for a constant budget of **5B tokens**.

Result. Final DLM loss decreases monotonically with increasing effective batch size, indicating that the CBS lies *beyond 8M tokens* and DLMs tolerate larger batches during A2D conversion.

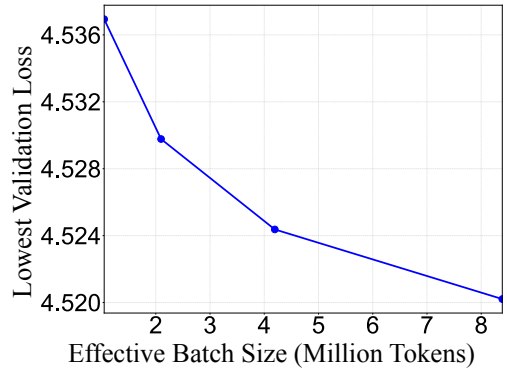


Figure 3.2: **Critical batch size via branched training.** Validation loss vs. tokens for branches with 1M, 2M, 4M, and 8M token batches, with learning-rate scaling $\eta(B) = \eta_0 \sqrt{B/B_0}$. Lower is better.

4 Scaling Diffusion Language Models using SCP with Sparse Mixture-of-Experts

Given that the goal of this work is to scale Diffusion Language Models using A2D recipes, we extend our SCP recipe to a 30B-parameter sparse Mixture-of-Experts DLM.

4.1 Experimental setup

Pretraining data and AR base model. We scale the SPC recipe to a sparse Mixture-of-Experts model with a total capacity of 30B parameters. We use Qwen3-30B-A3B (Yang et al., 2025) as the AR base model. Pretraining uses a 500B token mixture comprising Dolmino Text (general web/text), FLAN (instruction-style sources, including a focused FLAN subset), Dolmino Math, StackExchange, and Wikipedia.

Training settings. We use a global batch size of 33.5M tokens and train models using AdamW (Loshchilov & Hutter, 2017).

Benchmark evaluation. We assess model performance on a range of benchmarks spanning general reasoning, math and STEM, and coding domains. For general reasoning, we evaluate on MMLU (Hendrycks et al., 2021) (5-shot), MMLU-redux (Gema et al., 2025) (5-shot), BBH (Suzgun et al., 2023) (3-shot, CoT), Arc-c and RACE. For math and STEM, we include GSM8K (Cobbe et al., 2021) (4-shot, CoT). For coding, we report results on MBPP (0-shot) (Austin et al., 2021b).

4.2 AR Knowledge Retention during A2D Conversion

AR models are trained on trillions of tokens and encode broad world knowledge; a key question is how to preserve this knowledge during A2D conversion? In particular, we note that recent works have shown that knowledge (esp. factual associations) is encoded primarily in the MLP/FFN layers in transformer models (Meng et al., 2022; Dai et al., 2022). We therefore evaluate SCP’s ability to retain pretrained knowledge by tracking **GSM8K** accuracy at **30B**, **60B**, and **120B** tokens during conversion under four learning settings. All settings *update all parameters* and use 2% warmup; they differ only in weight decay and in how peak learning rates are assigned to parameter groups (attention vs. non-attention).

- **Setting 1: single peak LR, low weight decay.**
Peak lr 3×10^{-4} for all parameters; weight decay 1×10^{-4} .
Observation: GSM8K decreases monotonically from 60B→120B→180B, indicating forgetting under unconstrained adaptation.
- **Setting 2: single peak LR, high weight decay.**
Peak lr 3×10^{-4} for all parameters; weight decay 0.1.
Observation: The decline is less pronounced than in Setting 1, suggesting stronger regularization reduces forgetting, though degradation remains.
- **Setting 3: separate peak LRs by parameter group, high weight decay.**
Peak lr 1×10^{-4} for *attention* parameters; peak LR 1×10^{-6} for *non-attention* parameters (FFNs,

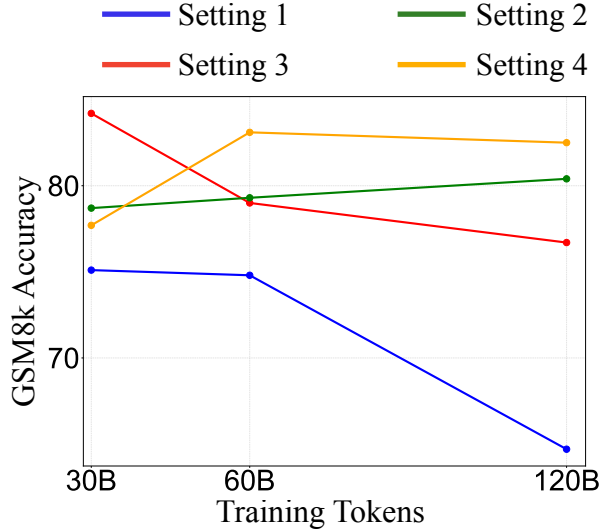


Figure 4.1: **AR Knowledge Retention during A2D conversion.** We report GSM-8k accuracy for four settings under 30B, 60B, and 120B token budgets. **Setting 1:** peak lr = 3×10^{-4} , wd = 1×10^{-4} ; **Setting 2:** peak lr = 3×10^{-4} , wd = 1×10^{-1} ; **Setting 3:** peak lr for attention = 3×10^{-4} , non-attention = 1×10^{-6} , wd = 1×10^{-1} ; **Setting 4:** peak lr for attention = 3×10^{-4} , non-attention = 1×10^{-8} , wd = 1×10^{-1} .

Table 4.1: Benchmark results. Best DLM results per row are in bold. * indicates reproduced results.

	RND1-Base	Dream 7B	LLaDA 8B	Qwen3-30B-A3B
Architecture	MoE	Dense	Dense	MoE
Type	DLM	DLM	DLM	AR
Training Tokens	0.5T	0.5T	2.3T	>30T
Total Parameters	30B	7B	8B	30B
Activated Params	3B	7B	8B	3B
<i>General Tasks</i>				
MMLU	69.6	69.5	65.9	79.5*
MMLU-Redux	72.6	-	-	81.2
BBH	67.5	57.9	47.4	81.5
ARC-C	63.2	59.8	47.5	55.7*
RACE	57.6	44.7	38.7	40.2
<i>Math & STEM</i>				
GSM8K	80.0	77.2	70.9	85.2*
<i>Coding</i>				
MBPP	65.4	56.2	39.0	74.2*

embeddings, norms, routers); weight decay 0.1.

Observation: Forgetting is small, but overall learning is slow.

- **Setting 4: separate peak LRs by parameter group, high weight decay.**

Peak lr 3×10^{-4} for *attention* parameters; peak LR 1×10^{-8} for *non-attention* parameters; weight decay 0.1.

Observation: Retention is strong (no systematic degradation across 60B/120B/180B).

Update to the SCP recipe. We adopt **Setting 4** as the default for RND1-Base: assign *separate peak learning rates* to parameter groups—higher for attention, near-zero for non-attention—with *weight decay* 0.1 across the model. This preserves autoregressive knowledge during A2D conversion at scale.

4.3 Results

We report comprehensive results for RND1-Base in Tab. 4.1. Unless stated otherwise, decoding and prompting follow the standard protocols for each benchmark (few-shot counts and CoT usage as listed in the evaluation suite), and scores are reported as accuracy (%). We include three comparisons: (i) the *AR baseline*, (ii) the RND1-Base (Ours), (iii) Dream7B-Base and (iv) LLaDA-8B Base. As one can observe, RND1-Base outperforms all prior DLMS in all 10 benchmarks.

4.4 Infrastructure

RND1-Base was trained on a cluster comprising 64 NVIDIA HGX B200 GPUs (8 GPUs per node, 8 nodes total). Within each node, GPUs are interconnected via NVLink and NVSwitch; cross-node communication is provided by InfiniBand.

Profiling We performed targeted profiling of a baseline autoregressive Qwen3-30B-A3B (Yang et al., 2025) model to optimize throughput for our experimental runs.

Since model size permitted sharding intranode only, we considered only tensor parallel (TP) sharding in non-MoE layers and expert parallel (EP) in MoE layers. This also enabled us to extrapolate our profiling

measurements on a single node to a multi-node setting, since increasing the number of nodes translated one-to-one to higher data parallelism, which with a concomitant increase in global batch size, kept computation per GPU relatively constant. Given the relatively small aspect ratios of model weights, tensor parallel resulted in consistently lower throughput when combined with expert parallel, with an EP-only configuration the highest in throughput.

For MoE communication, we benchmarked Megatron’s native **all-to-all** against DeepEP. By tuning the latter specifically for the intranode (NVLink) domain, we achieved a boost of +50 TFLOPs.

Too small or large a micro batch size (MBS) resulted in low throughput, due to insufficient computation / communication overlap at one end and memory pressure at the other end. We found an MBS of 8 to 16, with memory-intensive ops checkpointed (layernorms and MoE activations), to be a sweet spot.

WS	TP	PP	EP	DP	MBS	GBS	Seq	GAS	Dispatcher	Recompute	TFLOPs	Mem (GB)
8	1	1	8	8	8	2048	2048	32	DeepEP	Selective	454.0	173.3
8	1	1	8	8	8	2048	2048	32	A2A	Selective	391.6	170.2
8	1	1	8	8	16	2048	2048	16	DeepEP	None	378.8	109.7
8	1	1	8	8	16	2048	2048	16	A2A	None	331.0	104.6

Table 4.2: Profiling configurations for Qwen3-30B-A3B. WS = world size, TP = tensor parallel, PP = pipeline parallel, EP = expert parallel, DP = data parallel, MBS = micro batch size, GBS = global batch size, Seq = sequence length, GAS = gradient accumulation steps.

5 Conclusion

We release RND1-Base, A2D recipes, and inference code to catalyze progress in post-training, inference, and architectural innovation for scalable diffusion language modeling.

References

- Alan N Amin, Nate Gruver, and Andrew Gordon Wilson. Why masking diffusion works: Condition on the jump schedule for improved discrete diffusion. *arXiv preprint arXiv:2506.08316*, 2025.
- 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, 2021a.
- 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*, 2021b.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- 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*, volume 34, pp. 7432–7439, 2020.
- Keshigeyan Chandrasegaran, Michael Poli, Daniel Y. Fu, Dongjun Kim, Lea M. Hadzic, Manling Li, Agrim Gupta, Stefano Massaroli, Azalia Mirhoseini, Juan Carlos Nieves, Stefano Ermon, and Fei-Fei Li. Exploring diffusion transformer designs via grafting. 2025. URL <https://arxiv.org/abs/2506.05340>.
- 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.
- 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.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.
- 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, et al. Are we done with mmlu? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, 2025.
- Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4521–4534, 2023.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024.
- Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu: the finest collection of educational content, 2024.
- Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

- William Merrill, Shane Arora, Dirk Groeneveld, and Hannaneh Hajishirzi. Critical batch size revisited: A simple empirical approach to large-batch language model training. *arXiv preprint arXiv:2505.23971*, 2025.
- 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.
- 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. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025.
- Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. *arXiv preprint arXiv:2507.15857*, 2025.
- Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.
- 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. In *ACL (Findings)*, 2023.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- 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.
- 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.