d3LLM: Ultra-Fast Diffusion LLM using Pseudo-Trajectory Distillation

Yu-Yang Qian [1 2] **Junda Su** [1] **Lanxiang Hu** [1] **Peiyuan Zhang** [1] **Zhijie Deng** [4] **Peng Zhao**[† 2 3] **Hao Zhang**[† 1]

[1] University of California, San Diego [2] School of Artificial Intelligence, Nanjing University
[3] National Key Laboratory for Novel Software Technology, Nanjing University [4] Shanghai Jiao Tong University

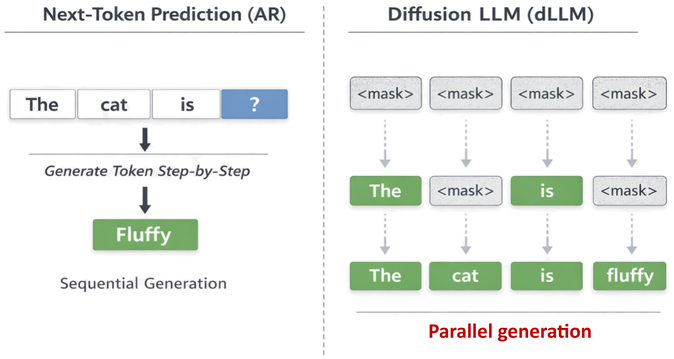
d3LLM GitHub

ICML International Conference On Machine Learning Seoul, South Korea

Background: Diffusion LLM

- ***dLLMs*** have shown promising capabilities, such as:

· Parallel generation · Error correction · Random-order generation



- *Closed-source* dLLMs showed effectiveness:
  - Inception - *Mercury*
  - Google - *Gemini Diffusion*
  - ByteDance - *Seed Diffusion*

Mercury©: 1109 tokens/s

- However, *open-source dLLMs* exhibited **lower speed and acc**:

· *LLaDA-8B* · *Dream-7B* · *SDAR*

- Previous Work: ***Improving*** dLLMs from different aspects:
  - ***Acceleration*** of dLLMs:
    - ***Fast-dLLM:*** efficient *training-free* decoding method;
    - ***dKV-Cache:*** KV-Cache for dLLMs;
    - ***dParallel:*** distilling dLLM to get a faster one;
    - ***D2F***: block-wise causal semi-AR-Diffusion;
    - ***dInfer***: systematic implementation of ***Efficient dLLM***;
    - ***ReFusion***: adapt from AR, slot-level parallel decoding.
  - Improving the ***performance*** of dLLMs:
    - ***MMaDA***: multimodal diffusion foundation models;
    - ***TraDo:*** dLLMs with *reasoning* ability;
    - ***Fast-dLLM-v2***: *directly post-training from an AR.*

However, previous works typically focus on only ***one-side of the coin***.

- Previous methods use ***single, isolated metrics***:
  - **Efficiency-only metrics:** tokens per second (TPS) or tokens per forward (TPF);
  - **Performance-only metrics:** accuracy (solve rate / pass@1).

- Key insight: a fundamental ***trade-off*** for dLLMs:
  - dLLM naturally lives on an ***accuracy–parallelism curve;***
  - Increasing parallelism almost always reduces accuracy, and vice versa;
  - Therefore, single metrics become misleading.

⇨ This motivates us to propose ***a better metric*** for dLLMs.

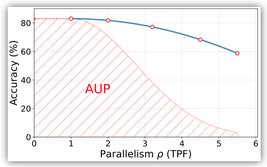
New Metric: *AUP* (Accuracy Under Parallelism)

⇨ This motivates us to propose ***a better metric*** for dLLMs.

Collect many parallelism-accuracy pairs $\{(\rho_i, y_i)\}_{i=1}^m$:

$$\text{AUP} \triangleq \rho_1 y_1 + \sum_{i=2}^{m} (\rho_i - \rho_{i-1}) \left(\frac{y_i W(y_i) + y_{i-1} W(y_{i-1})}{2} \right)$$

where $W(y) = \min(e^{-\alpha(1-y/y_{\max})}, 1)$ is weight function.



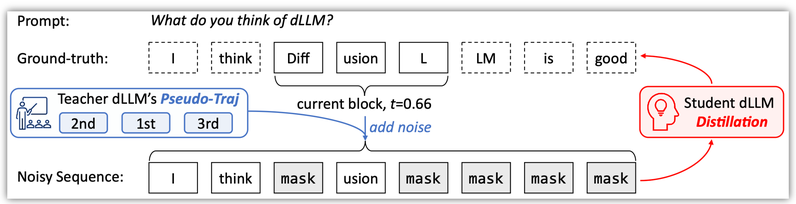
- We introduce ***AUP*** (*Accuracy Under Parallelism*) for dLLMs:
  - A weighted area under the accuracy–parallelism curve;
  - ***Jointly*** measures both accuracy and parallelism;
  - Exponentially ***penalizes*** accuracy drops.

Encourage dLLMs to ***strike a balance*** between accuracy and parallelism.

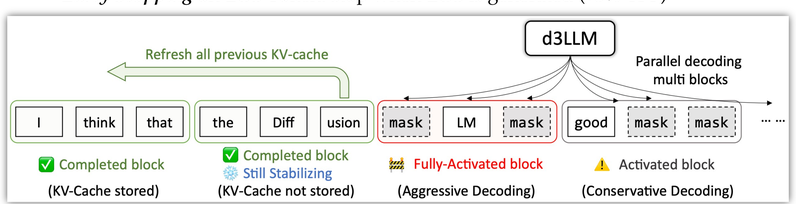
Our Approach: d3LLM framework

- Guided by AUP, we propose our method for efficient dLLMs: ***d3LLM*** (*pseuDo-Distilled Diffusion LLM*).
  - ➢ A framework that improves both ***training and inference;***
    - (i) Pseudo-trajectory distilla-tion ***(training)***
    - (ii)Multi-block decoding with KV-refresh ***(inference)***
  - ➢ ***Push*** the accuracy–parallelism ***frontier*** of dLLMs.

- Part (i): ***training recipe***
  - ➢ Regarding the *(post-)**training*** :
    - Utilizing the teacher's ***pseudo-trajectory***: which tokens can be confidently decoded earlier (+18% TPF)
    - ***Curriculum noise level***: gradually increase mask ratio (+12% TPF)
    - ***Curriculum window size***: gradually increase window size (+8% TPF)



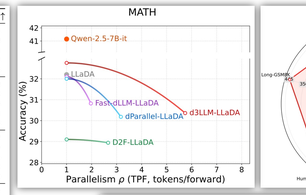
- Part (ii): ***decoding strategy***
  - ➢ Regarding the ***decoding*** :
    - Entropy-Based ***Multi-Block Decoding***: decoding multiple blocks (+30% TPF)
    - KV-Cache with ***Refresh***: refresh KV-caches before the stabilizing (+35% TPS)
    - ***Early Stopping*** on EOS Token: stop when EOS is generated (+5% TPF)

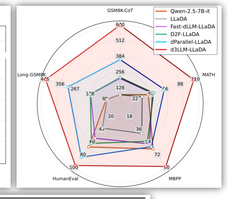


Experimental Results of d3LLM

- Experiments on *LLaDA, Dream,* and *Dream-Coder*:

Benchmark	Method	TPF ↑	Acc (%) ↑	AUP Score ↑
GSM8K-CoT (0-shot)	LLaDA	1.00 ± 0.0	72.6 ± 0.2	72.6 ± 0.2
	Fast-dLLM-LLaDA	2.77 ± 0.1	74.7 ± 0.2	205.8 ± 6.4
	D2F-LLaDA	2.88 ± 0.1	73.2 ± 0.3	209.7 ± 6.1
	dParallel-LLaDA	5.14 ± 0.1	72.6 ± 0.2	358.1 ± 6.2
	d3LLM-LLaDA	**9.11 ± 0.1**	73.1 ± 0.3	**637.7 ± 6.8**
MATH (4-shot)	LLaDA	1.00 ± 0.0	32.2 ± 0.4	32.2 ± 0.4
	Fast-dLLM-LLaDA	1.97 ± 0.1	30.8 ± 0.3	47.2 ± 2.9
	D2F-LLaDA	2.38 ± 0.1	28.7 ± 0.2	45.5 ± 2.8
	dParallel-LLaDA	3.17 ± 0.1	30.2 ± 0.2	64.5 ± 3.1
	d3LLM-LLaDA	**5.74 ± 0.1**	30.4 ± 0.3	**107.6 ± 3.2**
MBPP (3-shot)	LLaDA	1.00 ± 0.0	41.7 ± 0.3	41.7 ± 0.3
	Fast-dLLM-LLaDA	2.13 ± 0.1	38.6 ± 0.3	56.6 ± 3.7
	D2F-LLaDA	1.94 ± 0.1	38.0 ± 0.2	50.0 ± 3.6
	dParallel-LLaDA	2.35 ± 0.1	40.0 ± 0.3	60.5 ± 3.9
	d3LLM-LLaDA	**4.21 ± 0.1**	40.6 ± 0.2	**88.4 ± 4.0**
HumanEval (0-shot)	LLaDA	1.00 ± 0.0	38.3 ± 0.5	38.3 ± 0.5
	Fast-dLLM-LLaDA	2.56 ± 0.1	37.8 ± 0.4	54.0 ± 2.9
	D2F-LLaDA	2.69 ± 0.1	36.6 ± 0.5	62.0 ± 2.7
	dParallel-LLaDA	4.93 ± 0.2	39.0 ± 0.4	83.7 ± 4.8
	d3LLM-LLaDA	**5.95 ± 0.1**	39.6 ± 0.6	**96.6 ± 3.2**
Long-GSM8K (5-shot)	LLaDA	1.00 ± 0.0	78.6 ± 0.2	78.6 ± 0.2
	Fast-dLLM-LLaDA	2.45 ± 0.1	78.0 ± 0.3	175.4 ± 6.4
	D2F-LLaDA	2.70 ± 0.1	73.7 ± 0.2	168.5 ± 6.0
	dParallel-LLaDA	4.49 ± 0.1	76.7 ± 0.3	309.1 ± 6.2
	d3LLM-LLaDA	**6.95 ± 0.1**	74.2 ± 0.3	**441.1 ± 6.8**





Method	H100 TPS ↑	A100 TPS ↑	Acc (%) ↑
Qwen-2.5-7B-it	57.3 (1.0×)	50.4 (1.0×)	74.1
LLaDA	27.9 (0.5×)	19.2 (0.4×)	72.6
Fast-dLLM-LLaDA	114.3 (2.0×)	79.1 (1.6×)	74.7
D2F-LLaDA	102.1 (1.8×)	76.2 (1.5×)	73.2
dParallel-LLaDA	172.2 (3.0×)	105.9 (2.1×)	72.6
d3LLM-LLaDA	**288.9 (5.0×)**	**183.3 (3.6×)**	73.1

🚀 **3.6×–5× speedup** over AR model (Qwen-2.5-7B-it), **10× speedup** compared to vanilla LLaDA/Dream, with *negligible acc degradation.*

- ***Ablation study*** of each component in d3LLM framework:

Table 5. Ablation study on different distillation and decoding strategies of our method. We report the TPF, Accuracy, and AUP score of our *d3LLM-LLaDA* on GSM8K-CoT dataset (0-shot).

Distillation Recipe			Decoding Method		GSM8K-CoT (0-shot)		
Pseudo-trajectory	Curriculum Noise	Curriculum Window	Multi-block Decoding	Early Stop	TPF ↑	Acc (%) ↑	AUP Score ↑
					6.41 ± 0.1	72.2 ± 0.3	441.4 ± 3.2
✓			✓	✓	7.55 ± 0.1	72.1 ± 0.2	517.7 ± 3.9
✓	✓		✓	✓	8.46 ± 0.2	69.8 ± 0.4	551.3 ± 7.8
✓	✓	✓	✓	✓	**9.11 ± 0.1**	**73.1 ± 0.3**	**637.7 ± 6.8**
					7.01 ± 0.1	73.2 ± 0.1	492.9 ± 4.3
✓	✓	✓			9.07 ± 0.1	73.1 ± 0.3	635.0 ± 6.7
✓	✓	✓	✓	✓	**9.11 ± 0.1**	**73.1 ± 0.3**	**637.7 ± 6.8**

- Incorporating d3LLM models into ***SGLang engine*** (PR #20615) 🏎️:

			B200	H800/H100	A800/A100	Result		
	阈值	batch size	TPS	TPS	TPS	TPF	Acc	对齐HF
Qwen2.5-7B-Instruct	/	1	274.7	108.6	96.8	1	74.1	✅
Qwen3-8B	/	1	234.2	98.3	90	1	93.63	/
d3LLM-LLaDA (8B dense)	0.5	1	1240.99	545.31	251.61	9.91	75.36	✅
	0.5	4	1310.18	551.87	249.98	8.56	75.12	/
d3LLM-Dream (7B dense)	0.4	1	586.77	280.48	125.57	4.89	80.89	✅
	0.4	4	676.81	281.82	127.85	4.22	80.76	/
LLaDA 2.0 mini (16B A1B)	0.95	1	401.27	241.94	179.86	2.42	92.49	✅
	0.95	4	507.22	275.25	218.57	2.25	92.95	/
LLaDA 2.1 mini (16B A1B)	0.95	1	520.81	270.24	247.66	3.04	92.65	/
	0.95	4	516.89	390.04	310.55	2.49	92.87	/

🚀 **3×–4.5×** speedup over AR model (Qwen) with SGLang engine.

- We also maintain a dLLM leaderboard using ***AUP score***: