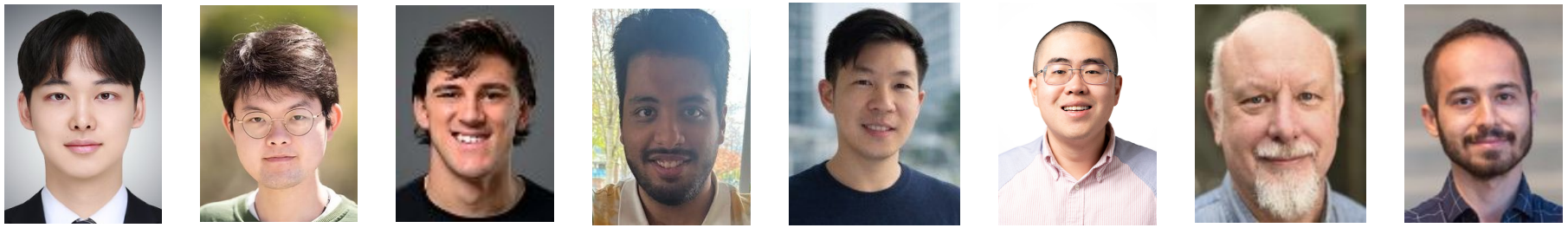
CDLM: Consistency Diffusion Language Models For Faster Sampling



Minseo Kim[1], Chenfeng Xu[2,3], Coleman Hooper[2], Harman Singh[2], Ben Athiwaratkun[3], Ce Zhang[3],

Kurt Keutzer[2], Amir Gholami[2]

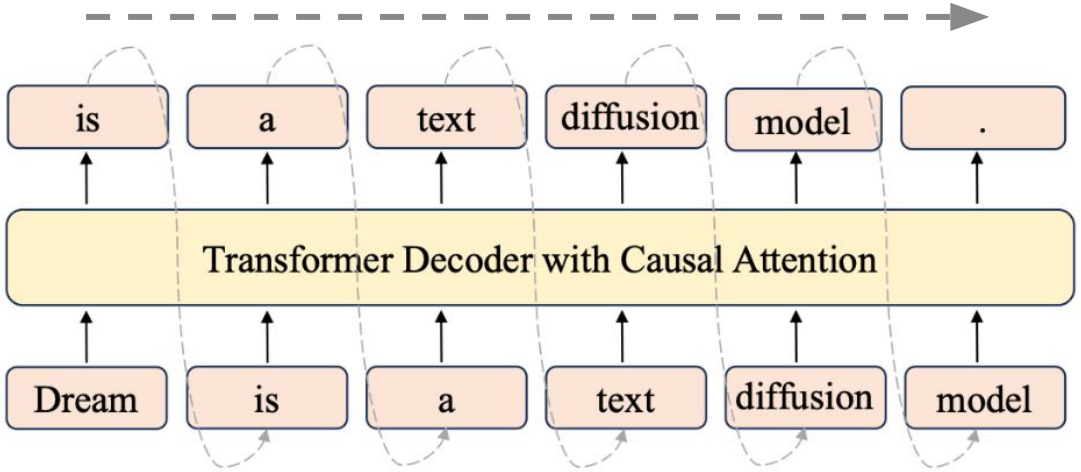
[1]Seoul National University, [2]UC Berkeley, [3]Together AI

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

MLSys

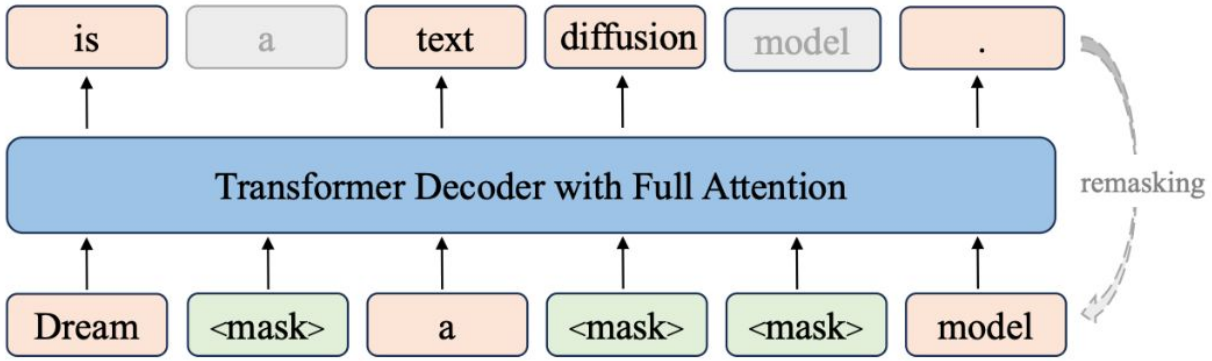
Why DLMs Are Promising

Left-to-right generation



(a) Autoregressive Modeling

Parallel denoising with bidirectional context

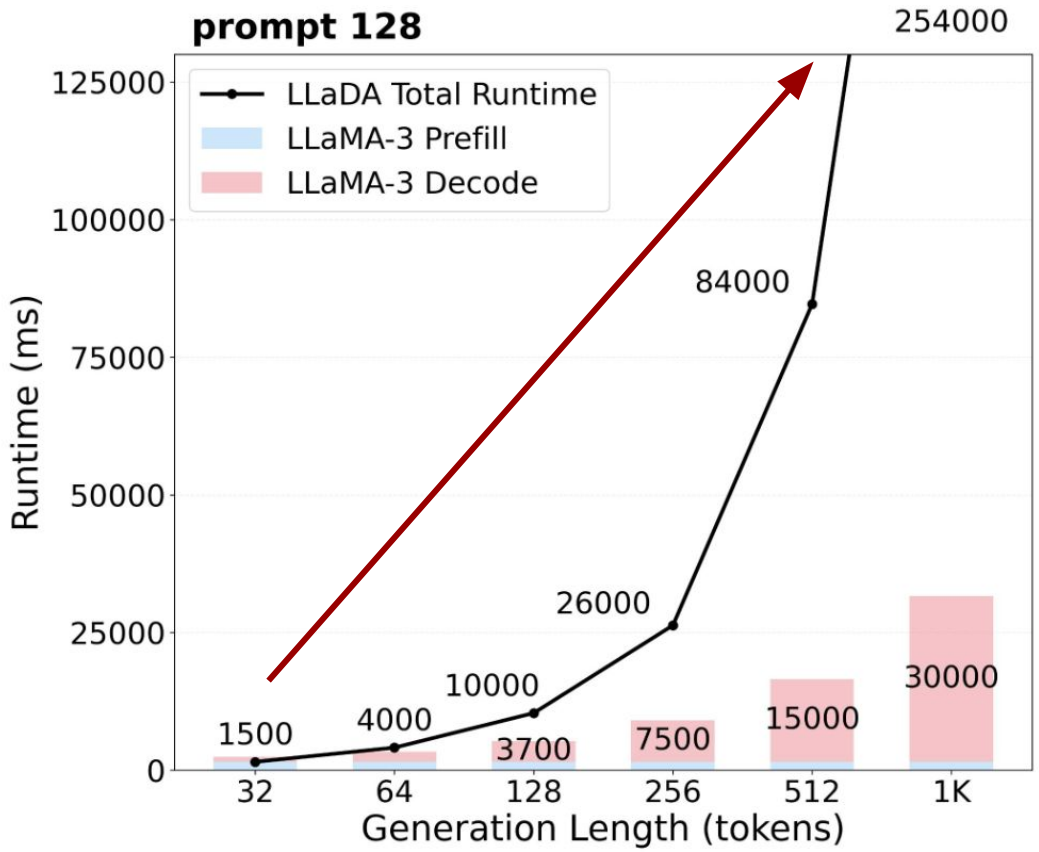


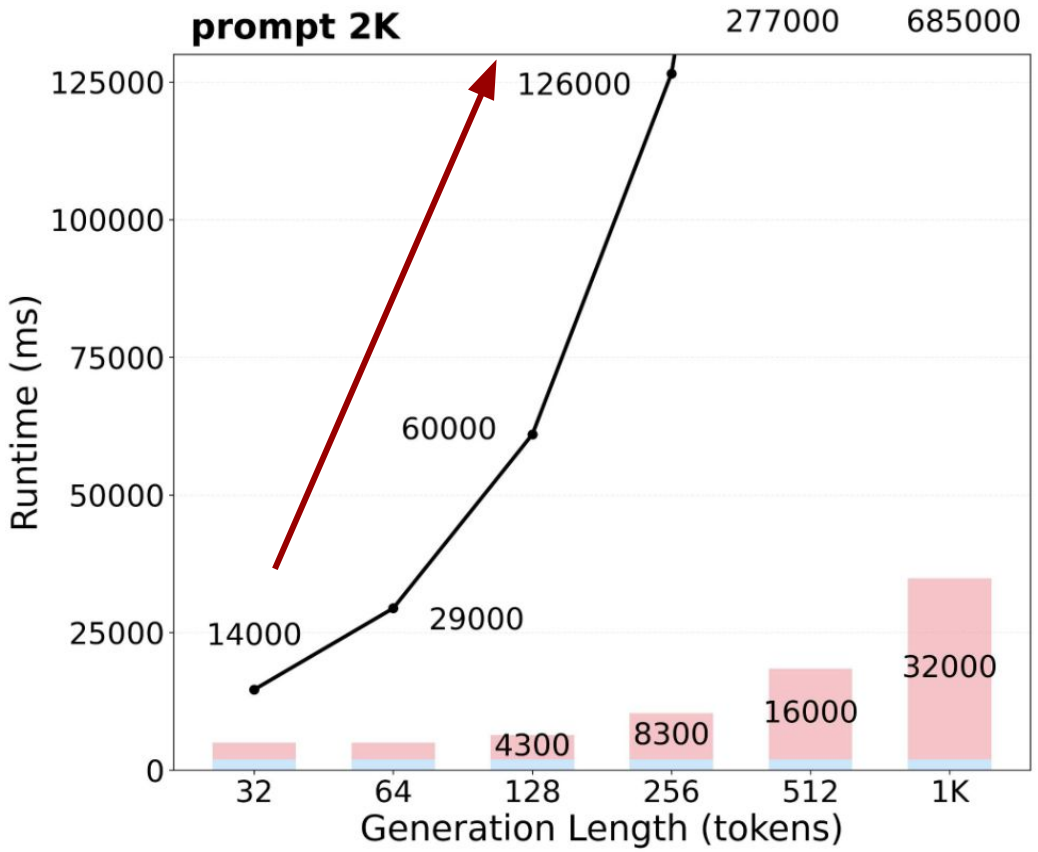
(b) Diffusion Modeling in Dream

Ye J, Xie Z, Zheng L, Gao J, Wu Z, Jiang X, Li Z, Kong L. Dream 7B: Diffusion Large Language Models. arXiv preprint. 2025.

But Open-Source DLMs Are Still Slow

* LLaDA-8B-Instruct vs. LLaMA-3-8B-Instruct, batch size 1, steps = generation length

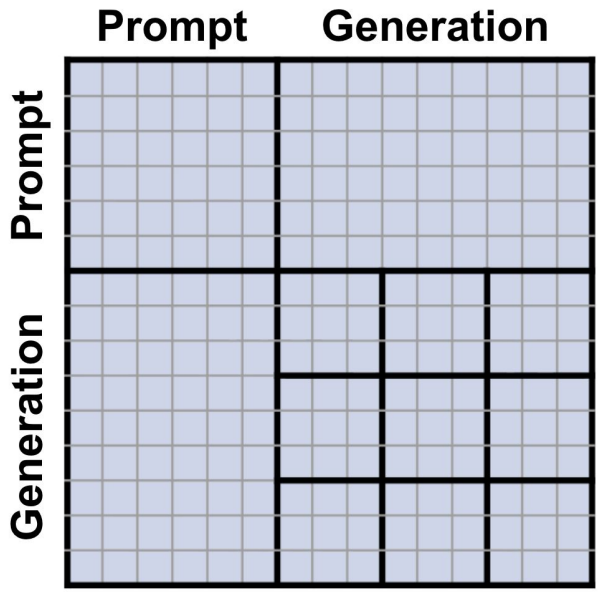




Kim M, Hooper C, Tomar A, Xu C, Farajtabar M, Mahoney MW, Keutzer K, Gholami A. Beyond Next-Token Prediction: A Performance Characterization of Diffusion versus Autoregressive Language Models. arXiv preprint. 2025.

Key Bottlenecks

(1) Cache Incompatibility



Fully bidirectional attention is incompatible with standard KV caching.

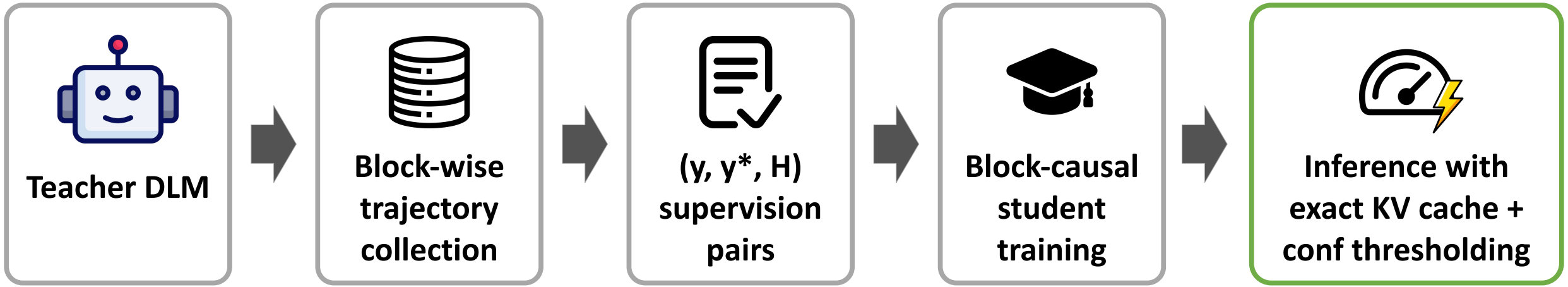
(2) Excessive Refinement Steps

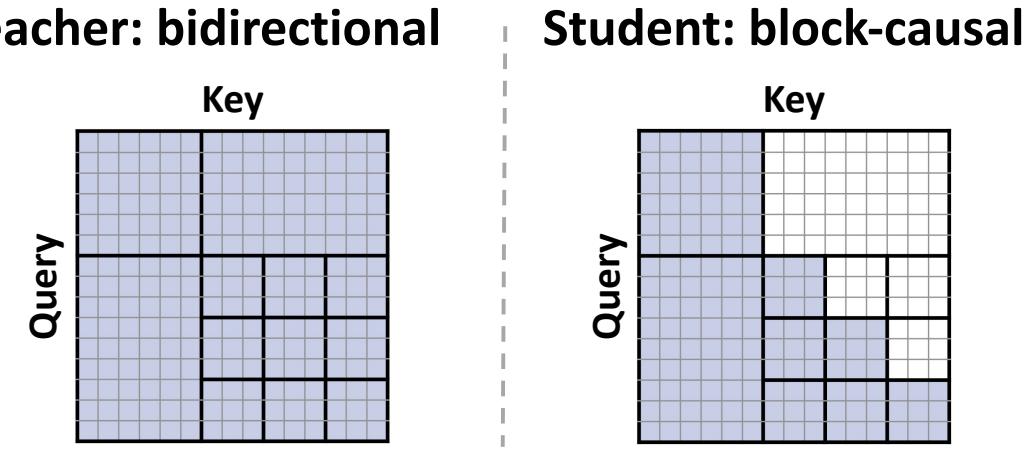
Step 0	British	munchkin	cats	[MASK]	[MASK]	[MASK]	[MASK]	[MASK]	[MASK]
Step 1	British	munchkin	cats	[MASK]	[MASK]	and	[MASK]	[MASK]	[MASK]
Step 2	British	munchkin	cats	are	[MASK]	and	[MASK]	[MASK]	[MASK]
Step 3	British	munchkin	cats	are	calm	and	[MASK]	[MASK]	[MASK]
Step 4	British	munchkin	cats	are	calm	and	playful	[MASK]	[MASK]
Step 5	British	munchkin	cats	are	calm	and	playful	[MASK]	.
Step 6	British	munchkin	cats	are	calm	and	playful	pets	.

High-quality generation requires many refinement steps.

CDLM Overview

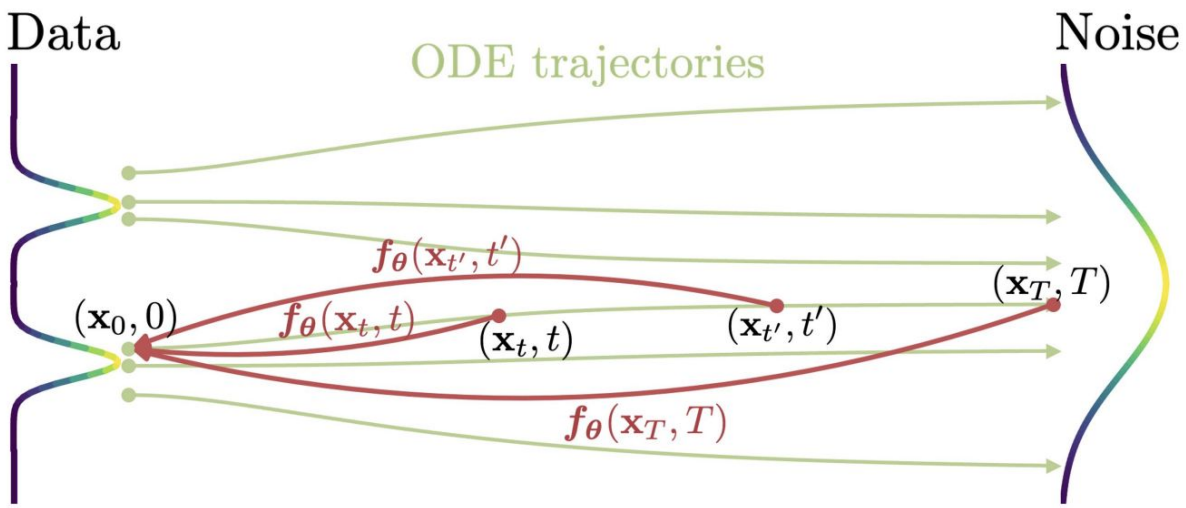
CDLM reduces refinement steps and enables KV caching by training a block-causal student on teacher trajectories.



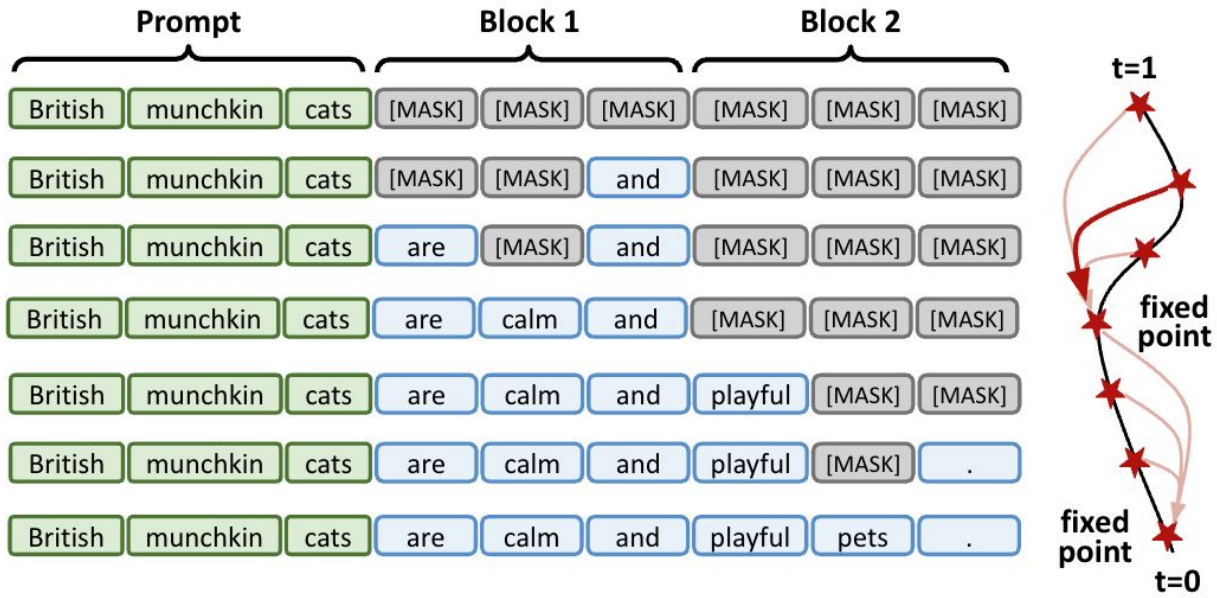


Can attend Cannot attend

Consistency Modeling Intuition



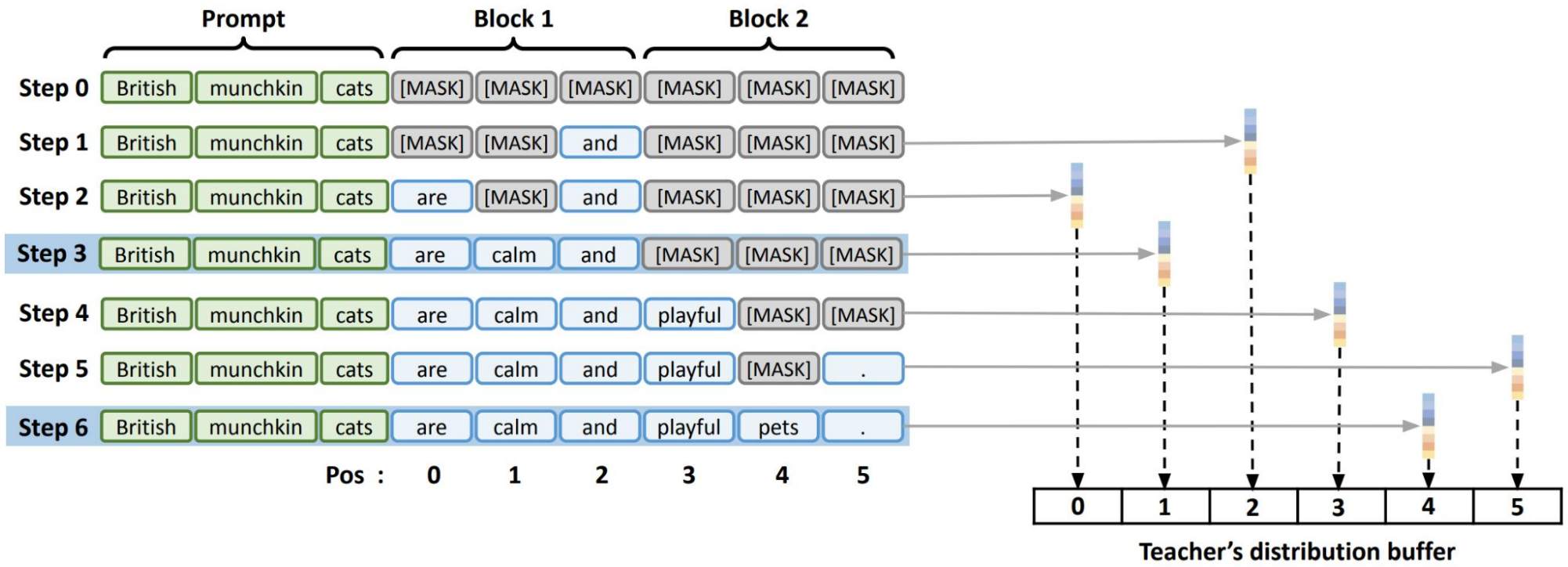
- ***Continuous data space***
- *Consistency between timesteps* ***on ODE trajectories***
- *Goal: map directly to* ***fully denoised data***



- ***Discrete token space***
- *Consistency over* ***teacher-generated denoising trajectories***
- *Goal: map directly to the* ***block-completion state*** *(inference-aligned)*

Song Y, Dhariwal P, Chen M, Sutskever I. Consistency Models. ICML. 2023.

Teacher Trajectory Collection



① Teacher decodes block-wise

Bidirectional attention, one token per step

② Store tokens and hidden states at finalization

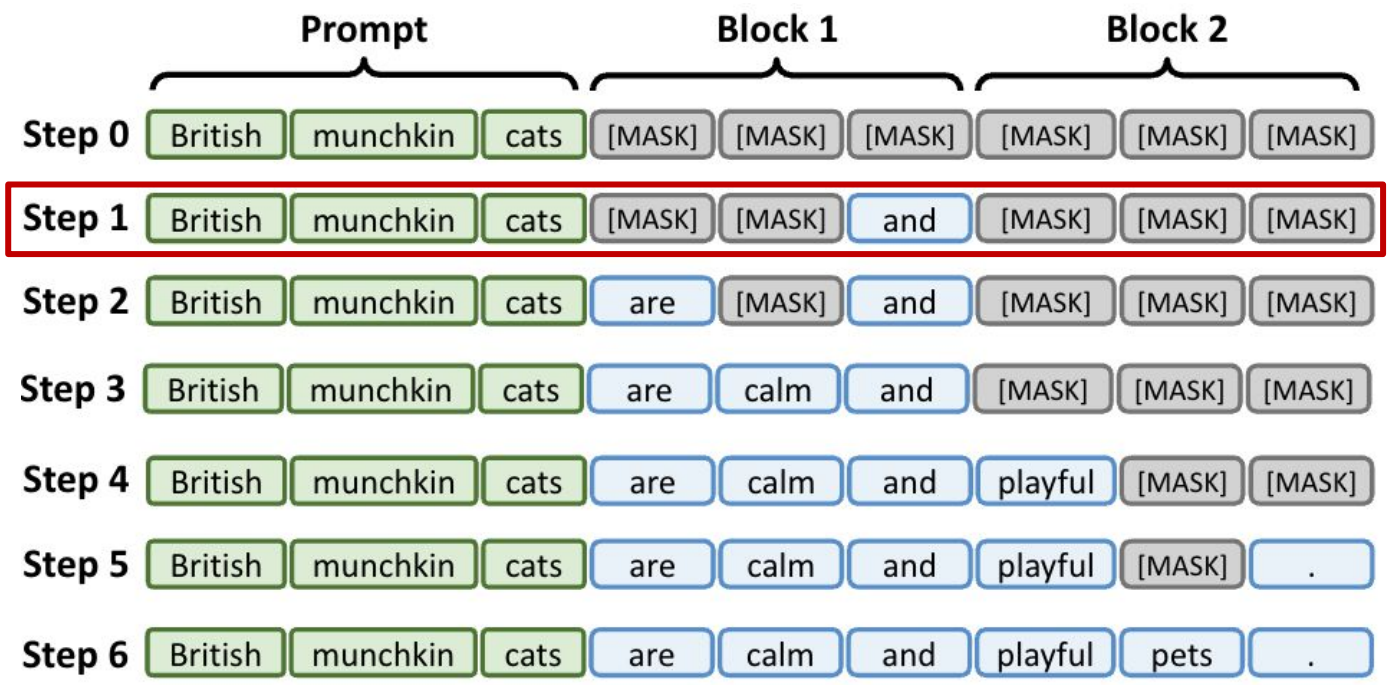
For white-box distillation

③ Use multiple sampling temperatures

Temperature affects token choices and reveal order

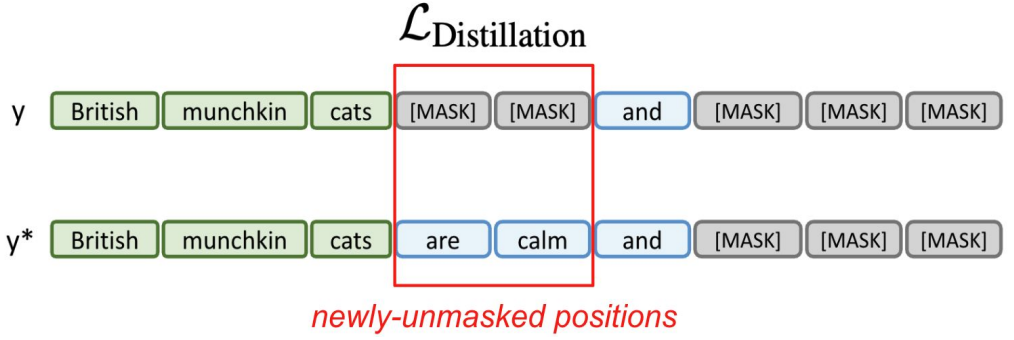
Training Objective: Distillation Loss

$$L = L_{distill} + w_{cons}\, L_{consistency} + w_{dlm}\, L_{DLM}$$



Sample an intermediate state y and its block-completion state y*

(a) Distillation loss

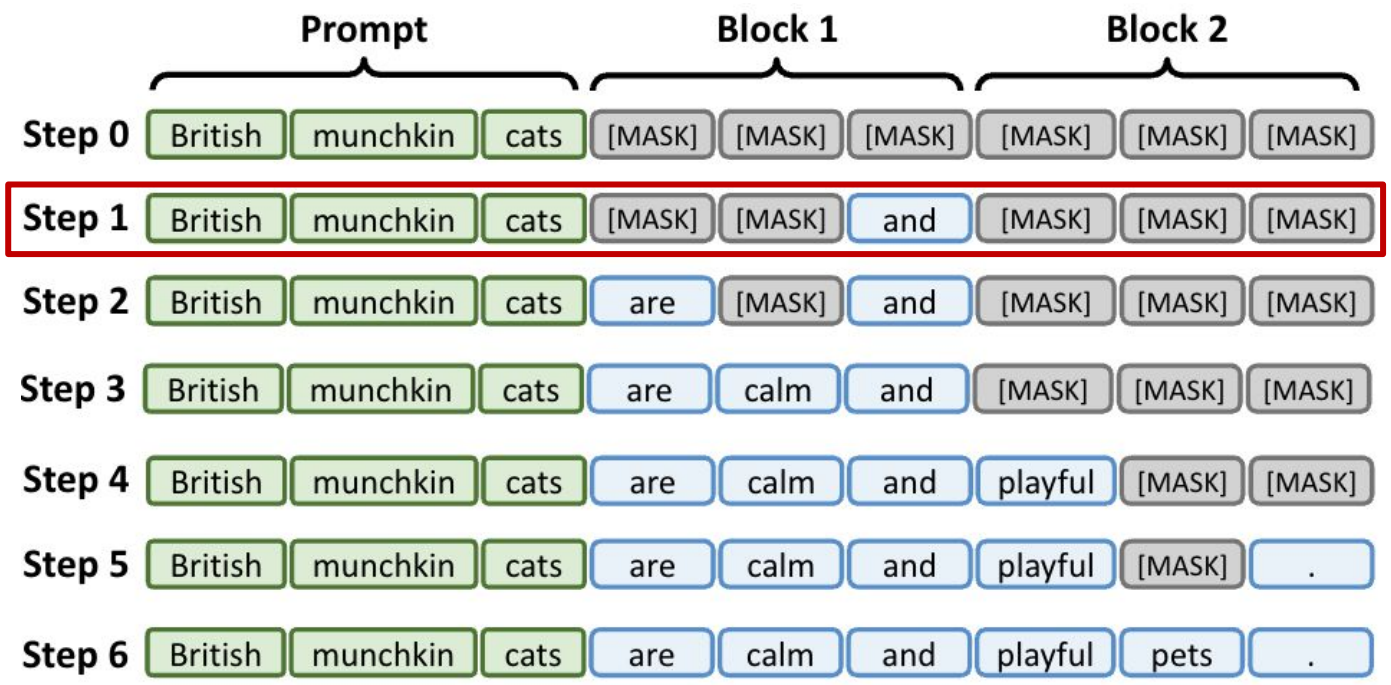


$$\mathcal{L}_{\text{Distillation}} = \mathbb{E}_{(x,\mathcal{T}_x,\mathbf{H}_x)\sim\mathcal{D}}\, \mathbb{E}_{y\sim\mathcal{T}_x}\left[\frac{1}{|\mathcal{U}_y|}\sum_{i\in\mathcal{U}_y} D_{\text{KL}}\left(p_i^{(T)} \,\|\, q_\phi(\cdot \mid y, x)_i\right)\right]$$

Intuition: Teaches the student to predict multiple new tokens at once.

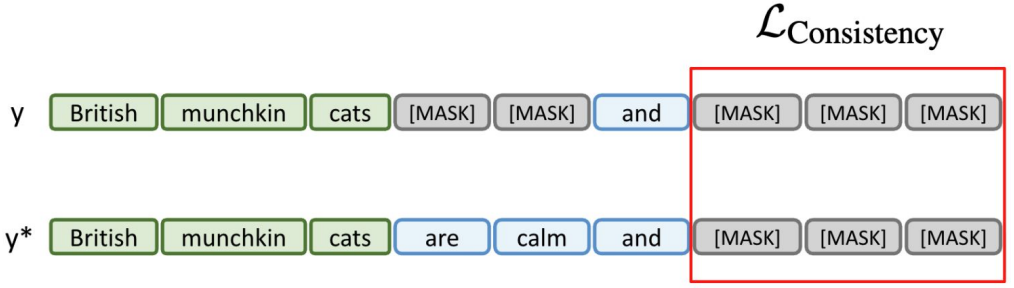
Training Objective: Consistency Loss

$$L = L_{distill} + w_{cons}\, L_{consistency} + w_{dlm}\, L_{DLM}$$



Sample an intermediate state y and its block-completion state y*

(b) Consistency loss

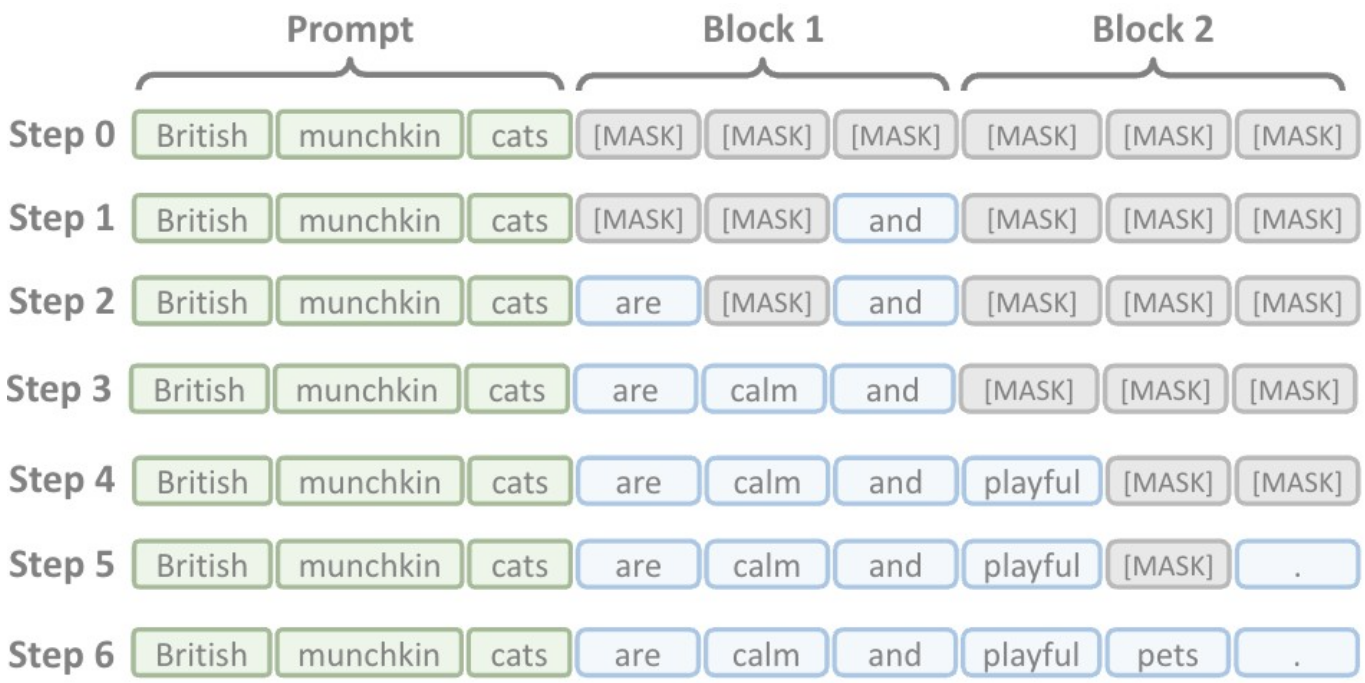


$$\mathcal{L}_{\text{Consistency}} = \mathbb{E}_{(x,\mathcal{T}_x)\sim\mathcal{D}}\, \mathbb{E}_{y\sim\mathcal{T}_x}\left[\frac{1}{|\mathcal{S}_y|}\sum_{i\in\mathcal{S}_y} D_{\mathrm{KL}}\Big(q_{\phi^-}(\cdot \mid y^\star, x)_i \;\|\; q_\phi(\cdot \mid y, x)_i\Big)\right]$$

Intuition: Stabilizes predictions between an intermediate state and its block-completion state.

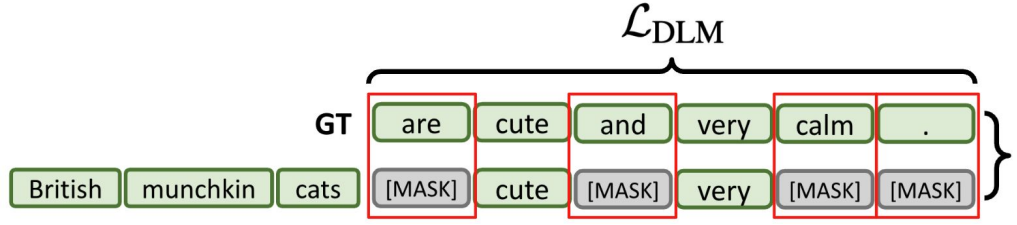
Training Objective: DLM Loss

$$L = L_{distill} + w_{cons}\, L_{consistency} + w_{dlm}\, L_{DLM}$$



Randomly mask ground-truth tokens

(c) DLM loss

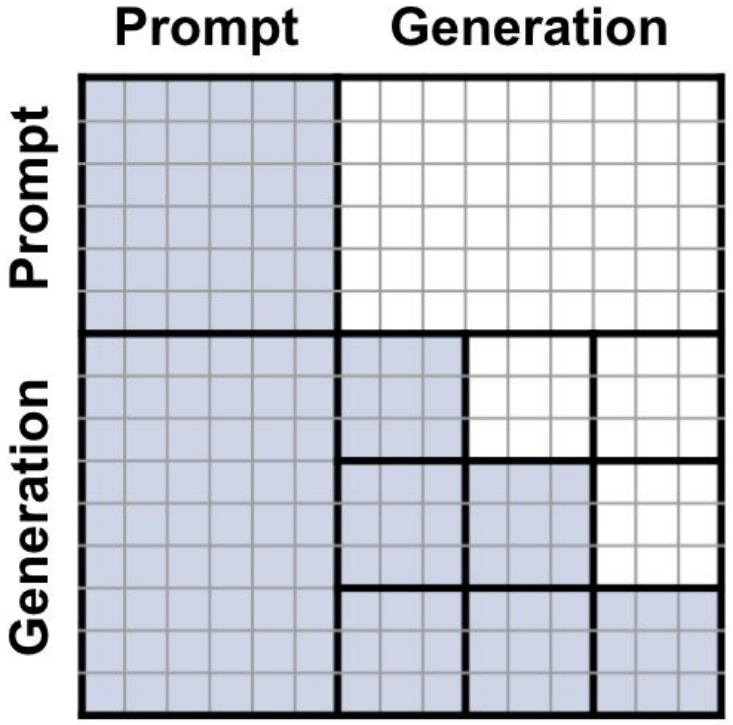


$$\mathcal{L}_{\text{DLM}} = -\mathbb{E}_{(x,\hat{\mathbf{y}})\sim\mathcal{D}}\,\mathbb{E}_t\left[\frac{1}{t}\sum_{i=1}^{L_g}\mathbf{1}\left[\hat{y}_{t,i} = \texttt{[MASK]}\right]\log q_\phi\left(\hat{y}_i \mid \hat{\mathbf{y}}_t, x\right)\right]$$

Intuition: Preserves the original masked denoising ability.

Inference

- ***Block-wise causality***
  - Decode block-by-block and cache finalized blocks
- ***Parallel decoding***
  - Within each block, finalize tokens in parallel using confidence thresholds



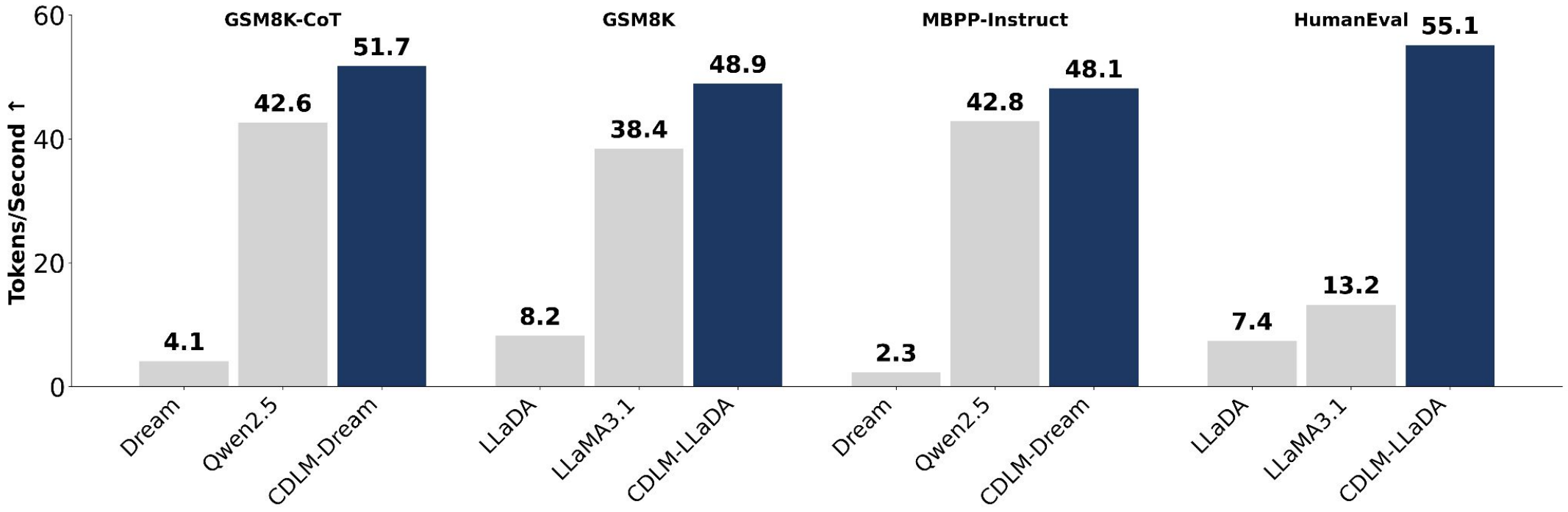
Main Results: Latency and Steps

3.6×-14.5× lower latency	**3.4×-7.9×** fewer refinement steps	**Competitive** accuracy on math and coding tasks

Benchmark	Method	Latency(s) ↓	Total Steps ↓	Score ↑
GSM8K-CoT (8-shot)	Dream	23.5 (×1.0)	256.0 (×1.0)	79.1
	Fast-dLLM-Dream	2.5 (×9.4)	60.8 (×4.2)	77.3
	CDLM-Dream	**2.1** (×11.2)	**44.1** (×5.8)	78.8
HumanEval-Instruct (0-shot)	Dream	13.4 (×1.0)	256.0 (×1.0)	48.2
	Fast-dLLM-Dream	2.5 (×5.4)	71.6 (×3.6)	46.3
	CDLM-Dream	**2.2** (×6.1)	**49.6** (×5.2)	50.0
MATH (4-shot)	LLaDA	25.7 (×1.0)	256.0 (×1.0)	24.1
	Fast-dLLM-LLaDA	5.0 (×5.1)	107.0 (×2.4)	32.5
	CDLM-LLaDA	**4.2** (×6.1)	**75.3** (×3.4)	28.3
MBPP-Instruct (0-shot)	LLaDA	11.4 (×1.0)	256.0 (×1.0)	40.8
	Fast-dLLM-LLaDA	3.4 (×3.4)	66.9 (×3.8)	35.0
	CDLM-LLaDA	**3.2** (×3.6)	**58.0** (×4.4)	38.4

* Fast-dLLM uses parallel decoding + dual-cache KV. dLLM-Cache and Fast-dLLM parallel-only are omitted.

Main Results: Throughput vs. AR Baselines



Each CDLM step is more expensive than an AR step (matrix-matrix multiplication), but finalizes multiple tokens in parallel.

Why Training Matters: Naive Truncation Fails

GSM8K results

Method	Latency (s) ↓	Steps ↓	Score ↑
Dream-7B-Instruct	4.4	48	41.8
CDLM–Dream (ours)	2.1	44.1	78.8
LLaDA-8B-Instruct	6.0	56	60.3
CDLM–LLaDA (ours)	3.3	57.7	73.9

Naively truncating steps hurts quality; CDLM reduces steps safely.

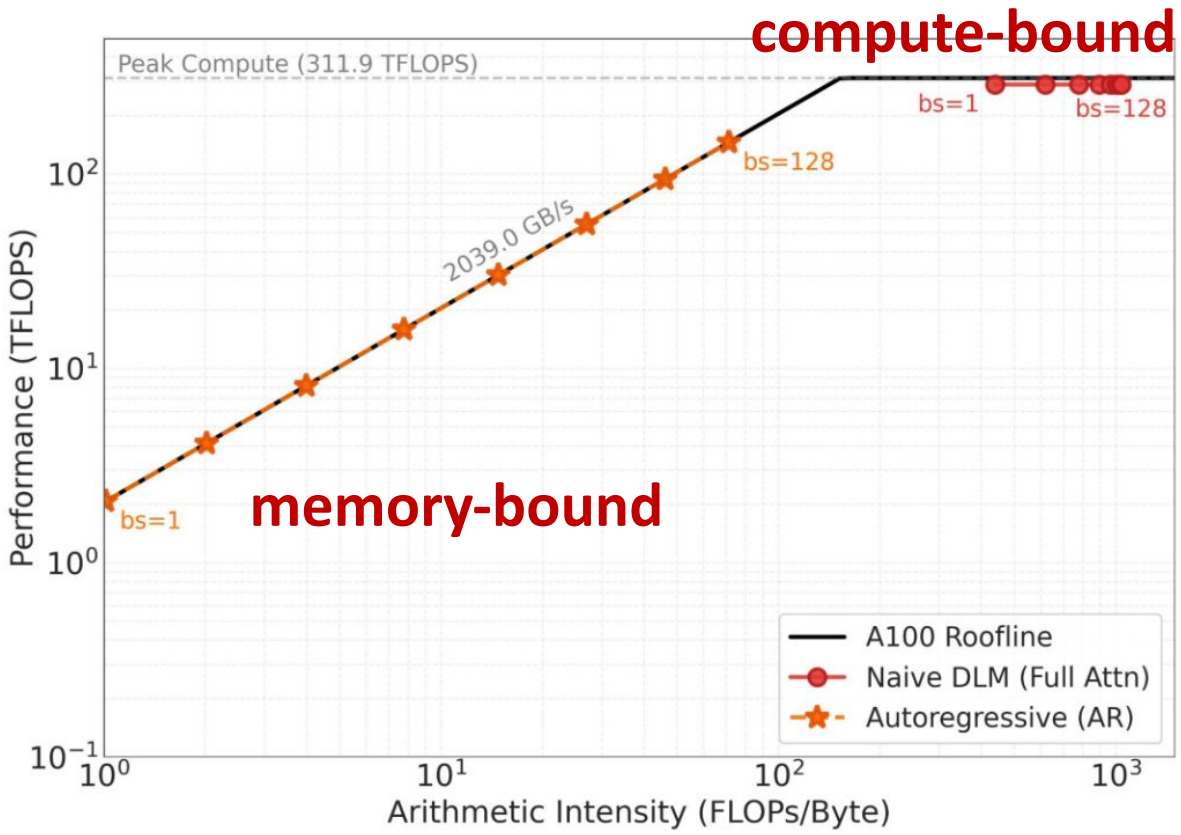
CDLM roughly halves latency at similar step counts via KV caching.

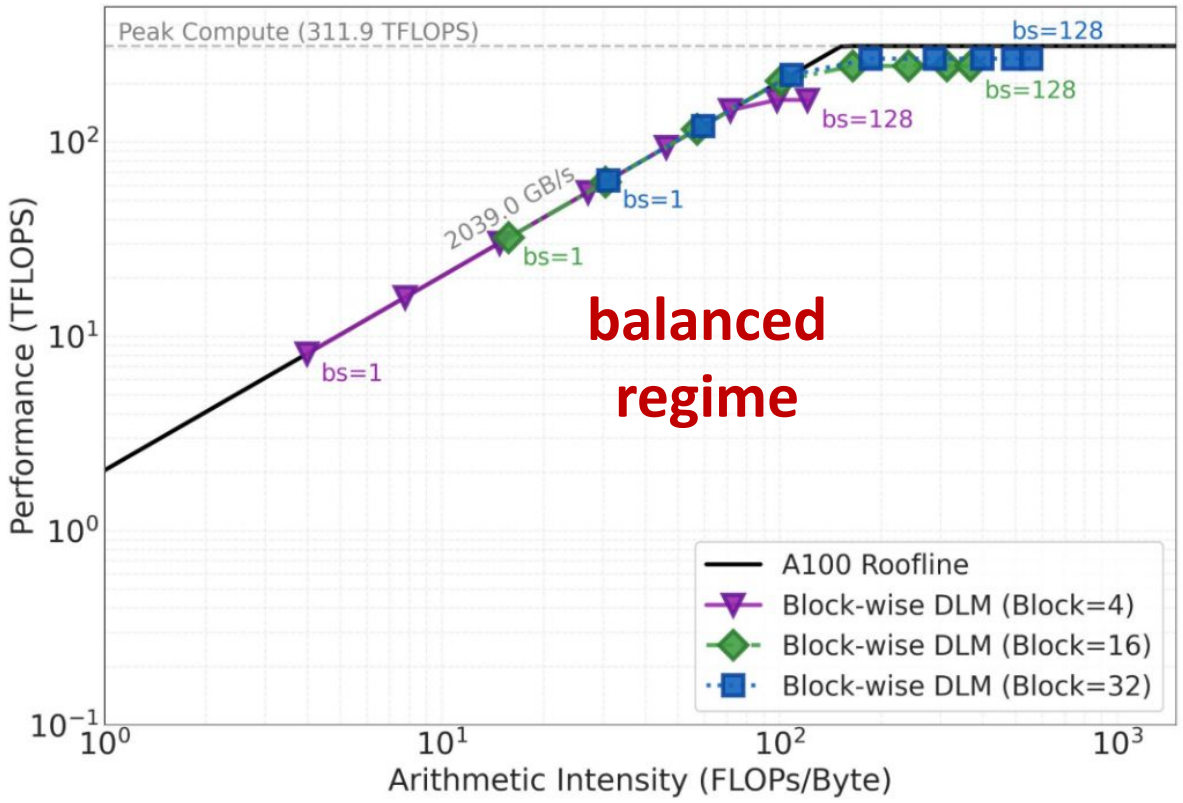
Ablation: Why Three Losses?

L_distill	L_consistency	L_DLM	Score ↑	Convergence
✔			weaker	fast
	✔		collapse	
✔	✔		good, less robust	fast
✔	✔	✔	strong	fast

Distillation loss anchors training; consistency loss stabilizes jumps;
DLM loss preserves task ability.

System-Level Scalability: Arithmetic Intensity





Block-wise DLMs occupy a middle ground between AR and vanilla DLMs: better low-batch compute utilization without saturating compute immediately.

Future Directions

- ***Scale the trajectory corpus with broader task coverage***
- ***Distill from stronger DLM teachers***
- ***Extend to longer generation budgets***

Thank you for listening!

Paper

Code

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

MLSys