NI Sampling: Accelerating Discrete Diffusion Sampling by Token Order Optimization

Enshu Liu

Menu

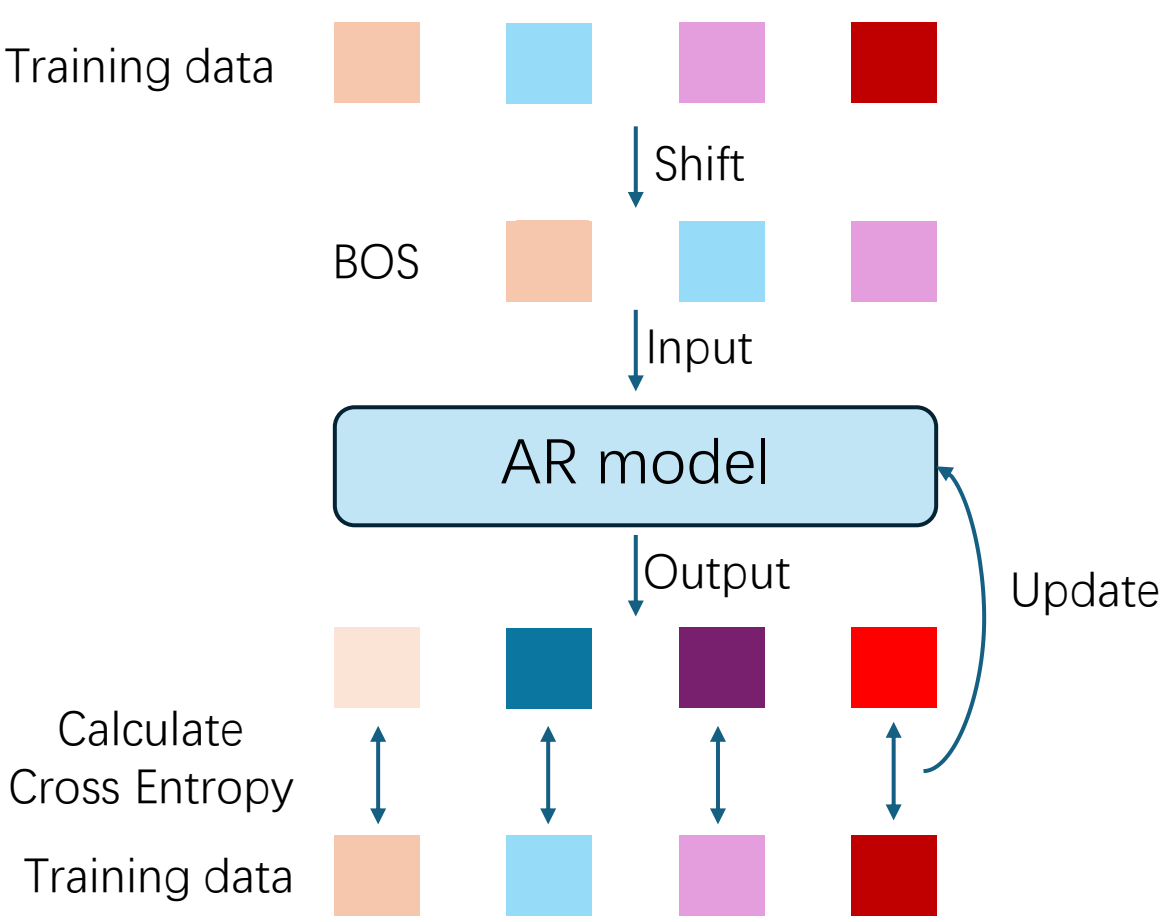
1. Background: Discrete Diffusion Model
2. Token Order Optimization

Menu

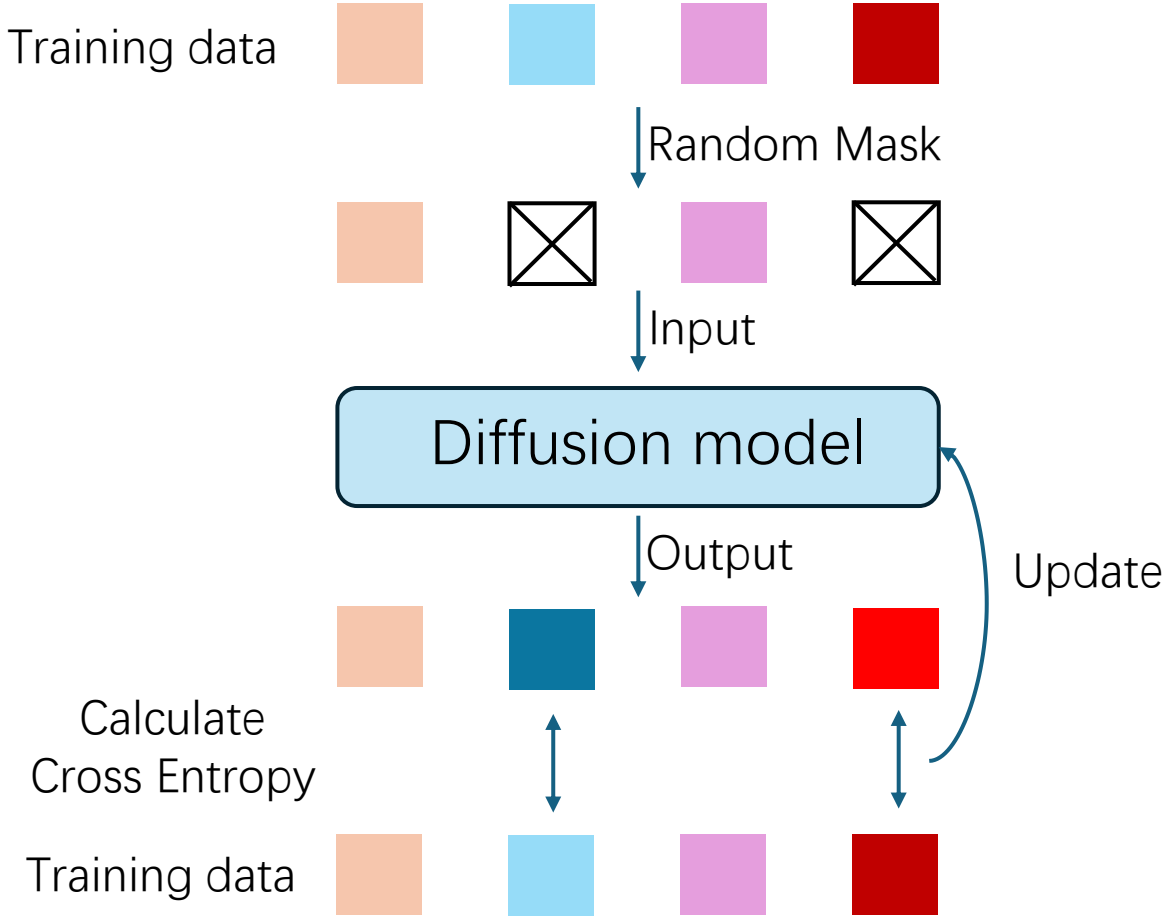
1. Background: Discrete Diffusion Model
2. Token Order Optimization

Discrete Diffusion Model: A new paradigm of Language Modeling

Training process of AR model

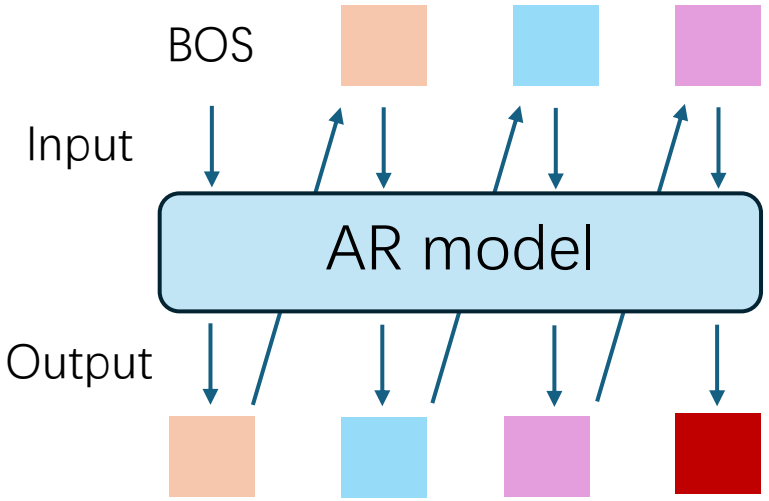


Training process of diffusion model

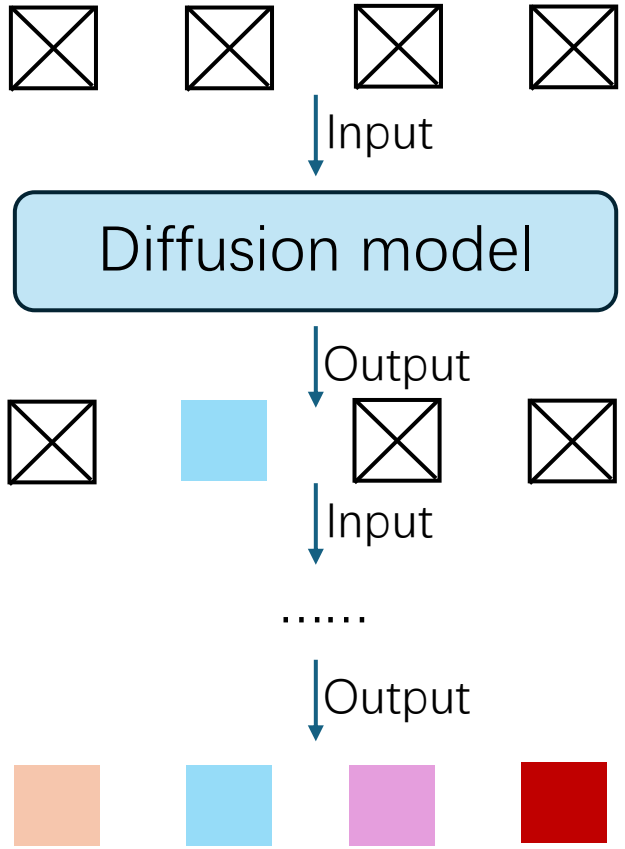


Discrete Diffusion Model: A new paradigm of Language Modeling

Generation process of AR model

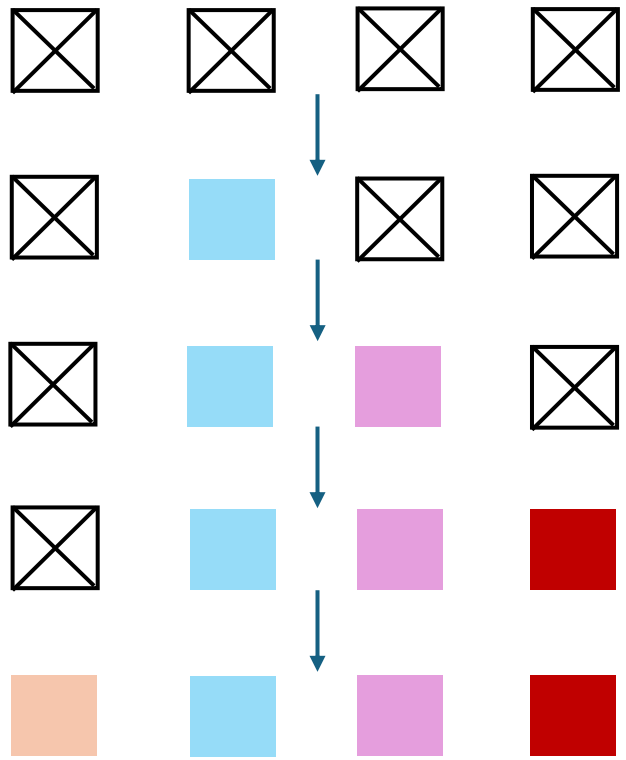


Generation process of diffusion model

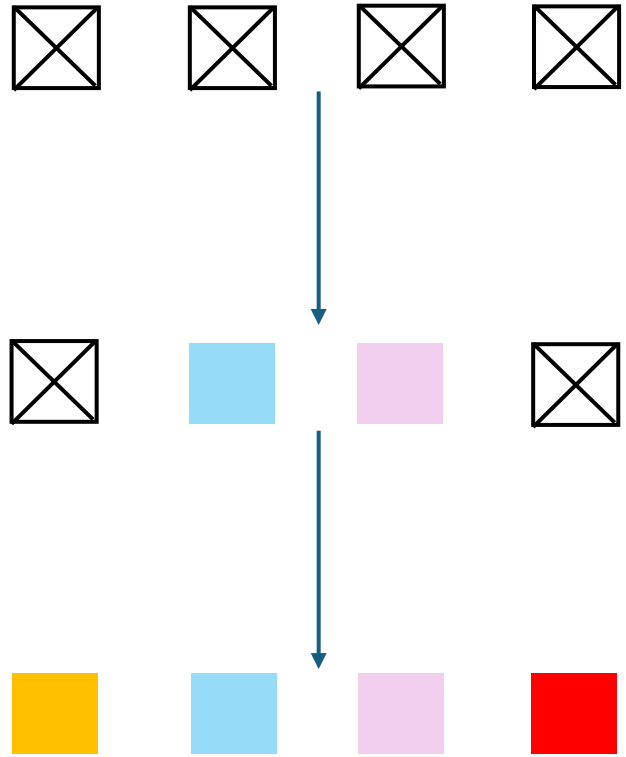


Property 1: Discrete Diffusion Model can Potentially Generate Multiple Tokens per Step

Default: 1 token per step

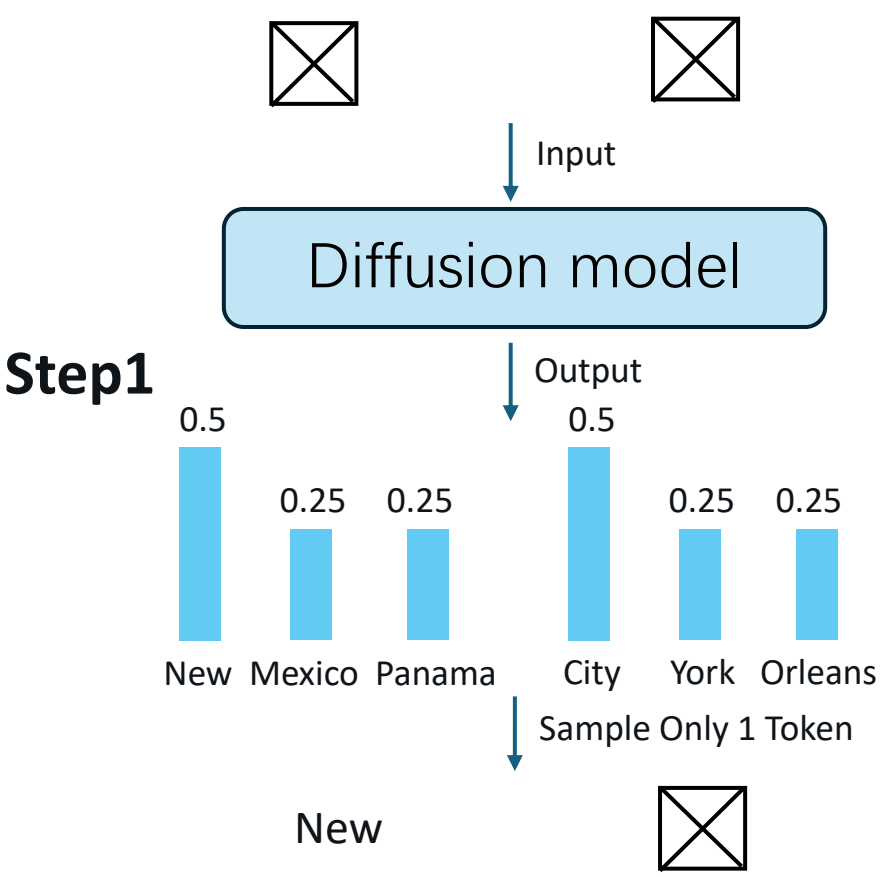


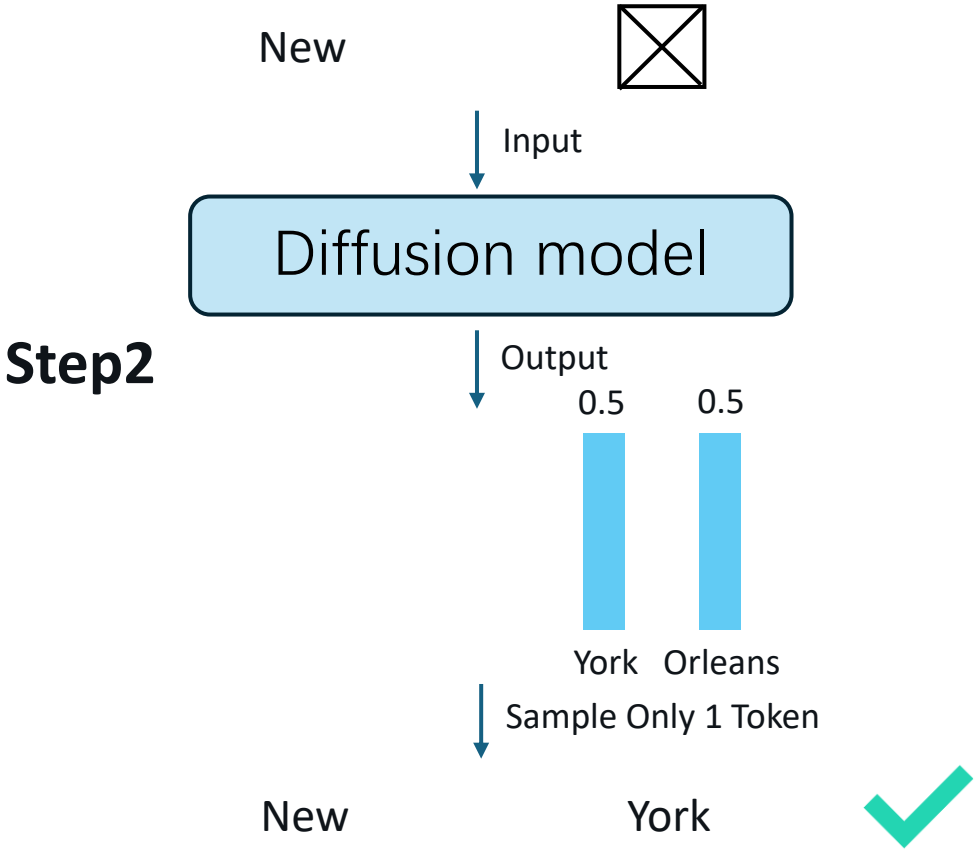
Multiple tokens per step



Generating Multiple Tokens per Step Causes Loss of Correspondence

Pick one city uniformly at random from the following four cities: New York, New Orleans, Mexico City, or Panama City.

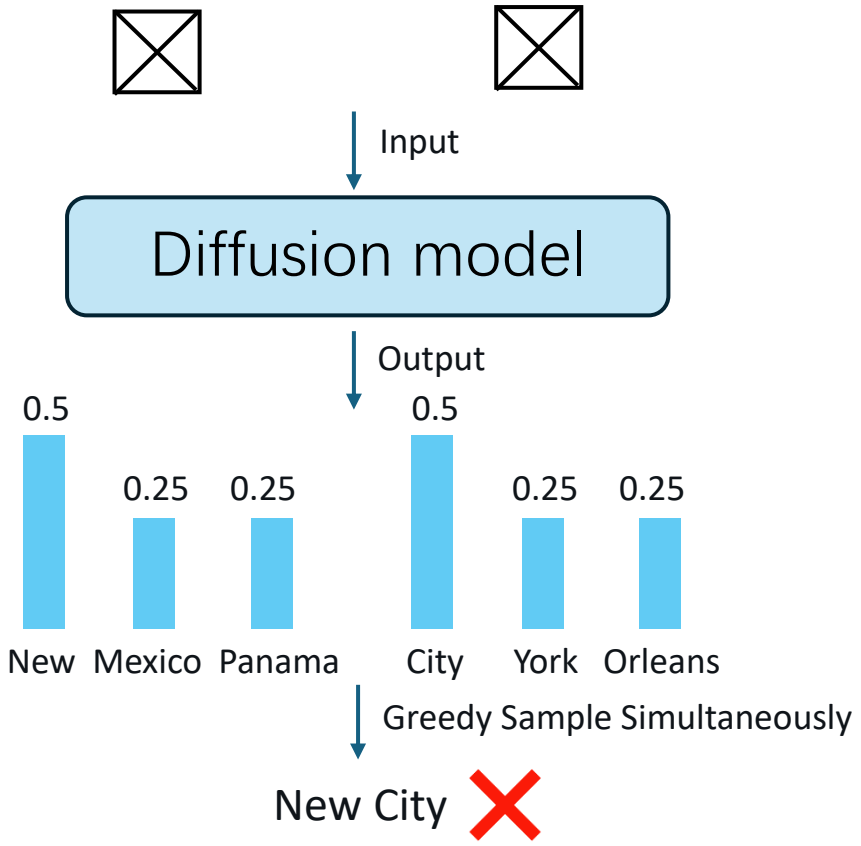




Sampling 1 token/step works well

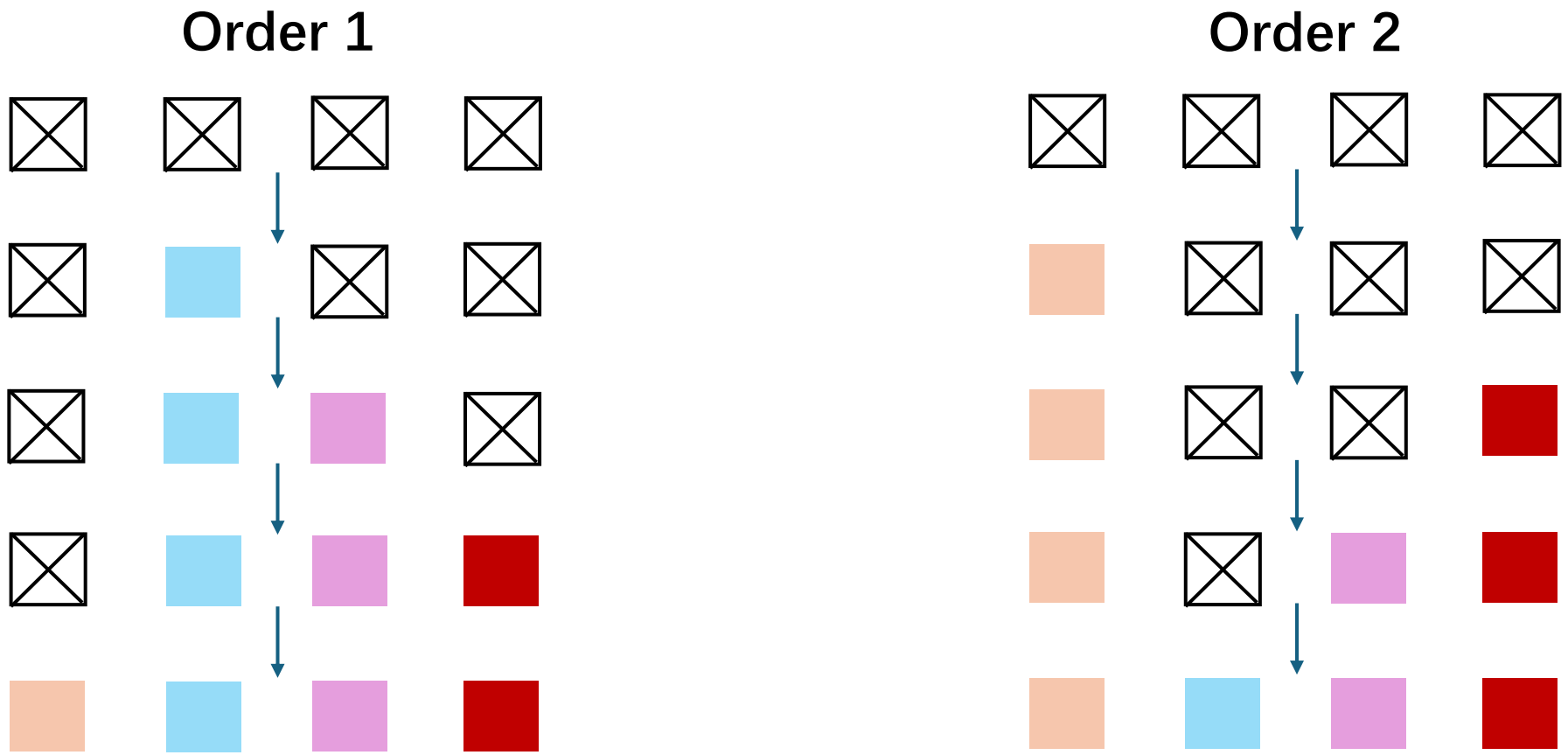
Generating Multiple Tokens per Step Causes Loss of Correspondence

Pick one city uniformly at random from the following four cities: New York, New Orleans, Mexico City, or Panama City.

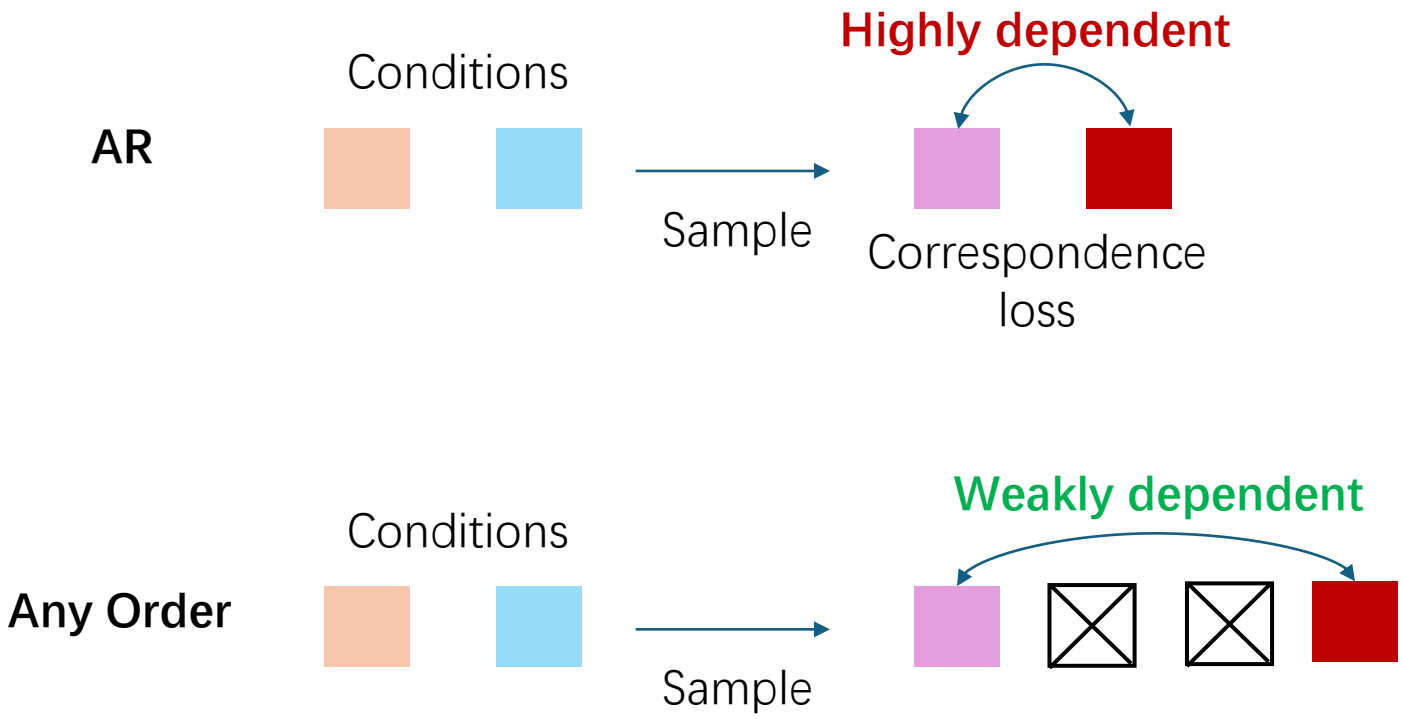


Parallel sampling replaces $P(Y_1, Y_2)$ with $P(Y_1)P(Y_2)$

Property 2: Discrete Diffusion Model can Generate with any Order



Any Order Property may Help Parallel Sampling



We want to fully utilize the “any order” property to enhance discrete diffusion model:

1. Higher Speed (i.e., more tokens per step)
2. Better Performance

Problem Definition: Token Order Optimization

We define a token order as **a partition of a permutation of {1,2,⋯,L}**, where L is the generation length

$$\pi = \left(\{i_1, i_2, \dots, i_{a_1}\}, \{i_{a_1+1}, \dots, i_{a_2}\}, \dots, \{i_{a_{n-1}+1}, \dots, i_L\}\right)$$

$(i_1, \dots, i_L)$ is a permutation of $\{1, \dots, L\}$

Given L and prompt C, we want to choose the best order which maximizes reward F:

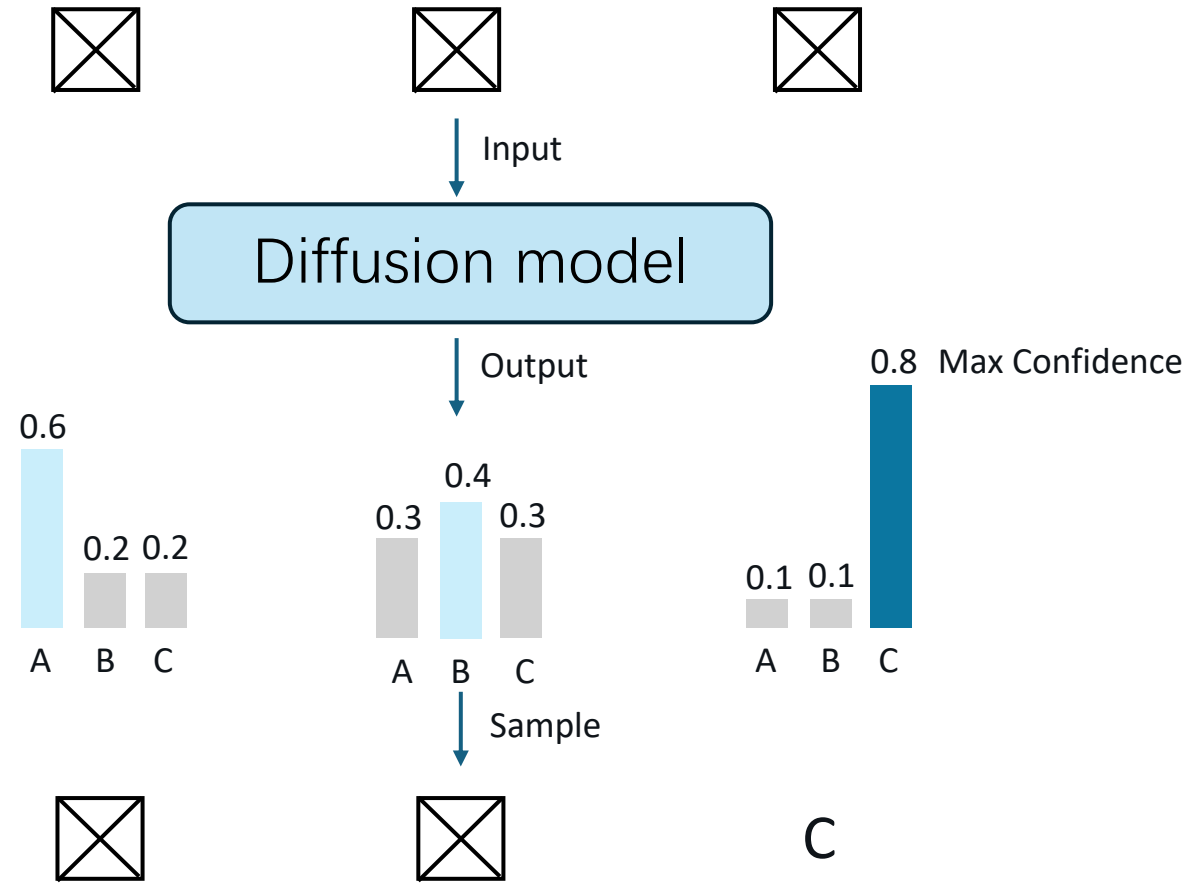
$$\pi^* = \underset{\pi}{\operatorname{argmax}} F(\pi, C, L)$$

Menu

1. Background: Discrete Diffusion Model
2. Token Order Optimization

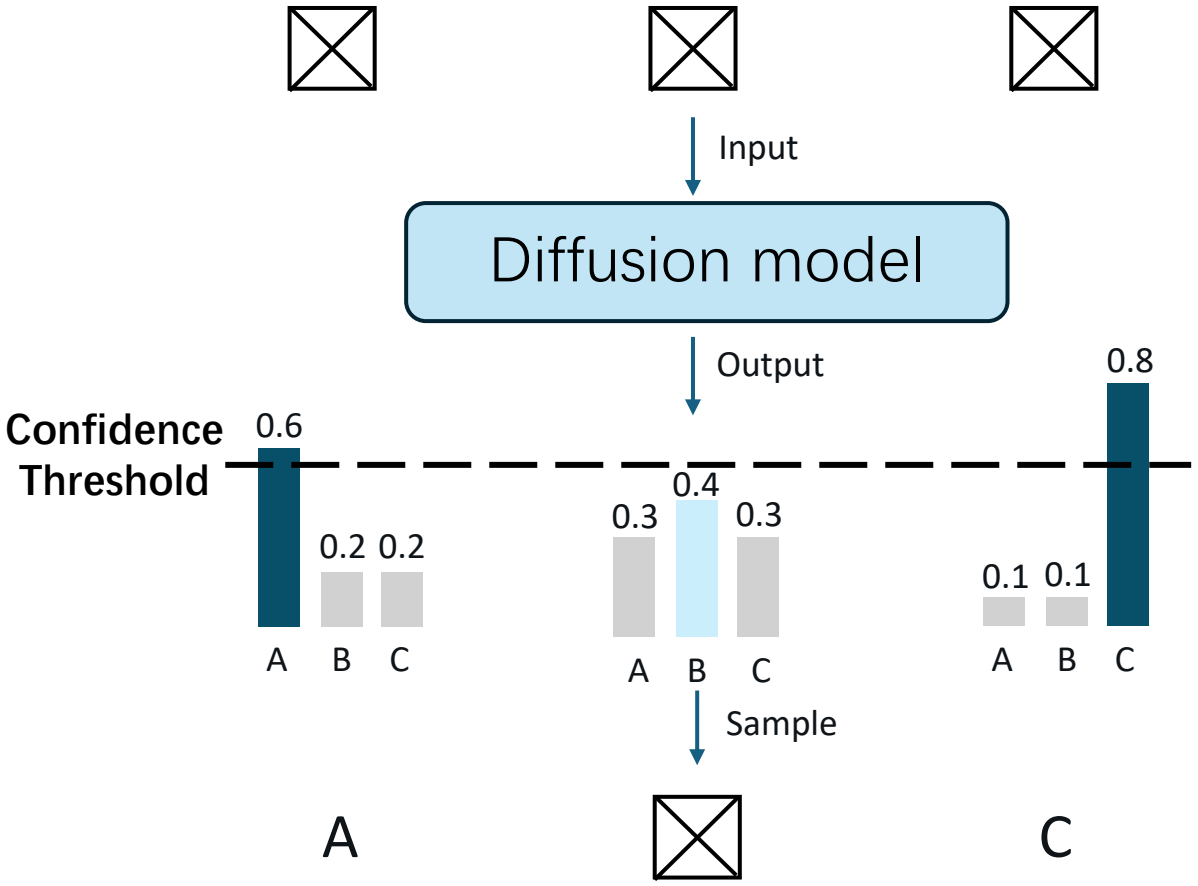
Existing Method of Token Order Selection: Max Confidence

Sample the token with the highest confidence at each step



Existing Method of Token Order Selection: Confidence Threshold

Sample all tokens whose confidence is larger than a threshold

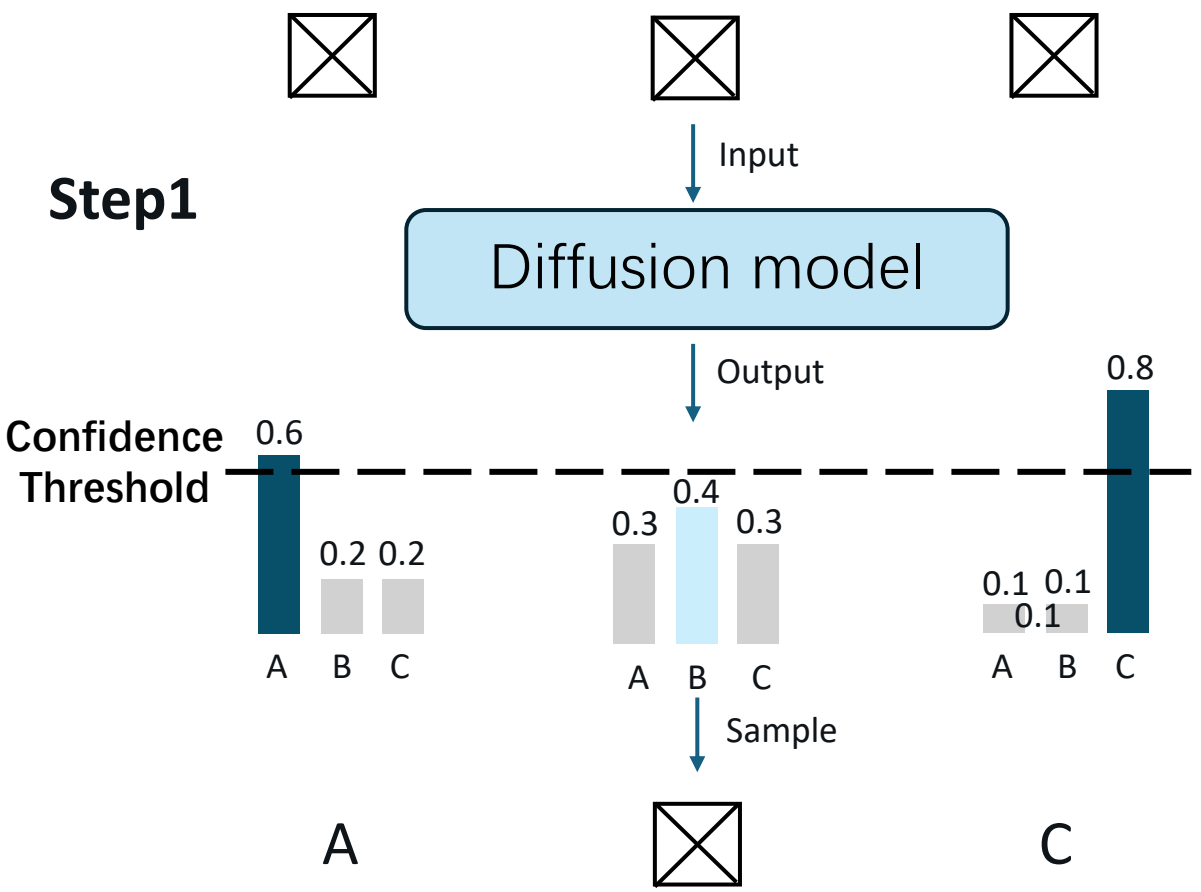


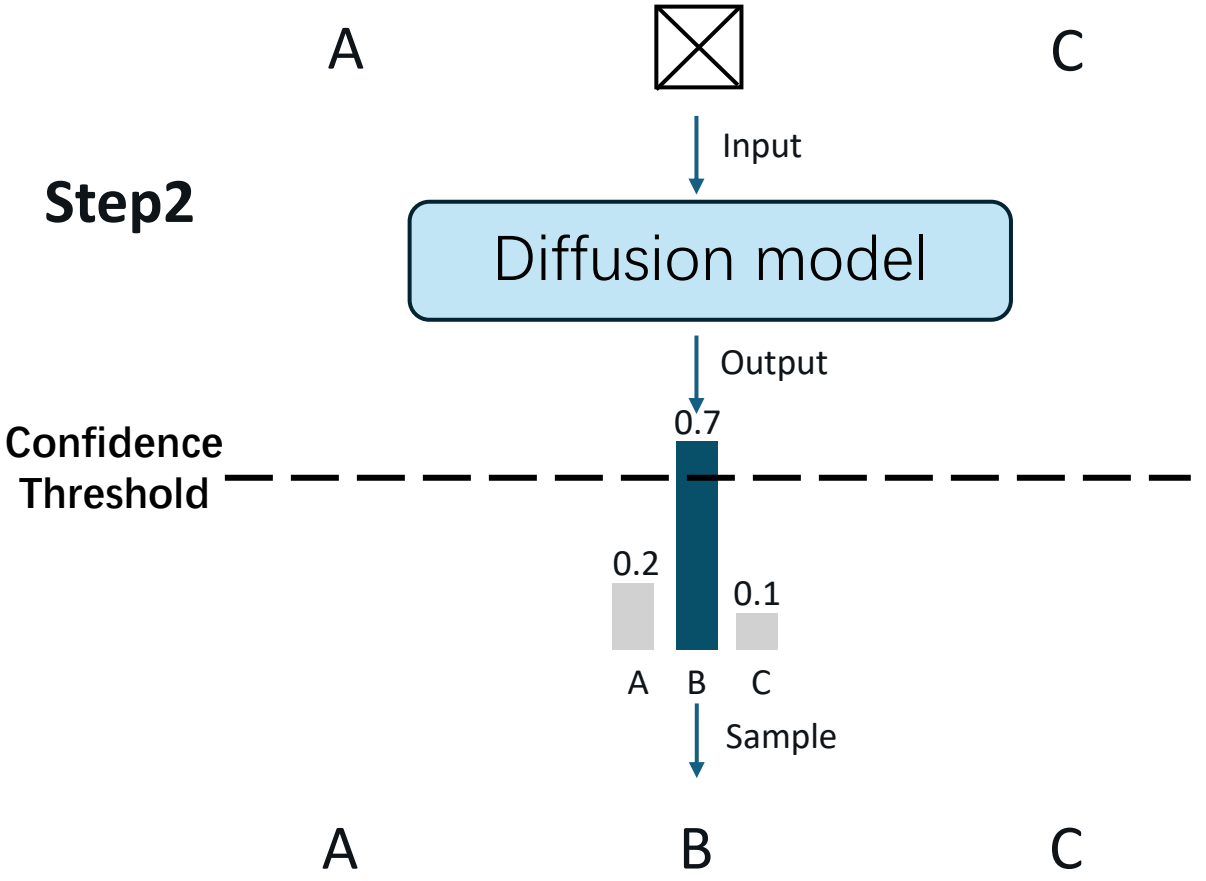
Intuition:

If the model is confident enough about certain tokens, greedily sampling them simultaneously incurs no correspondence loss.

Motivation of this Work

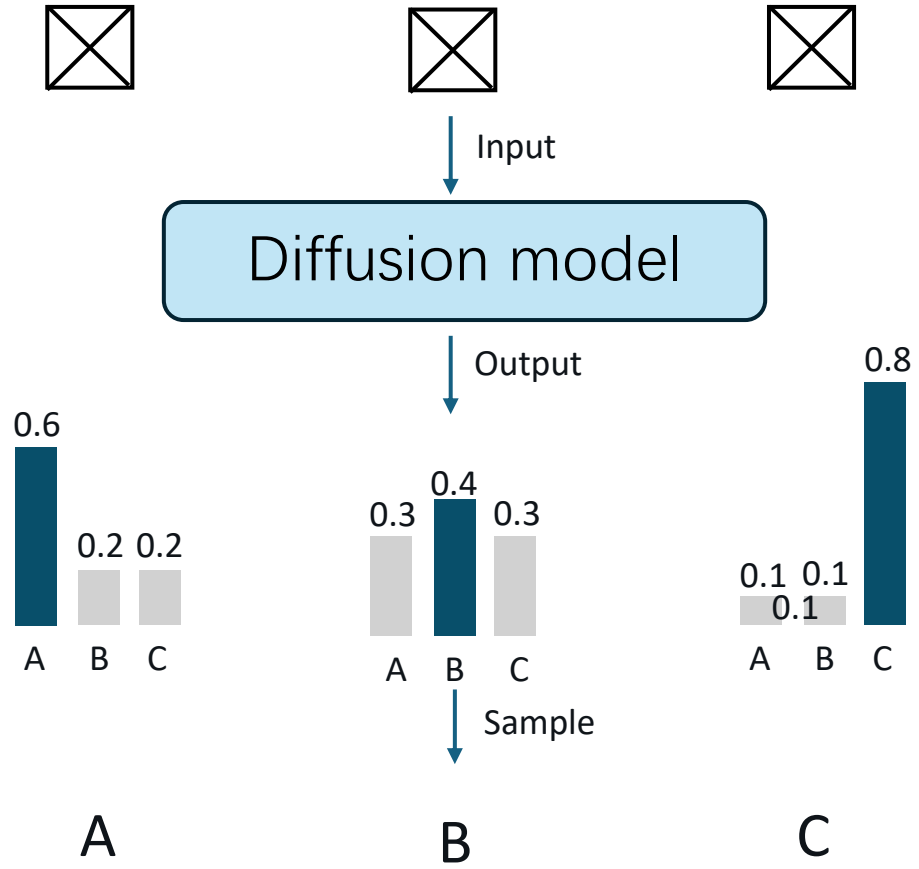
The model predicts many tokens **correctly** at each step but can't sample them due to **low confidence**.





Motivation of this Work

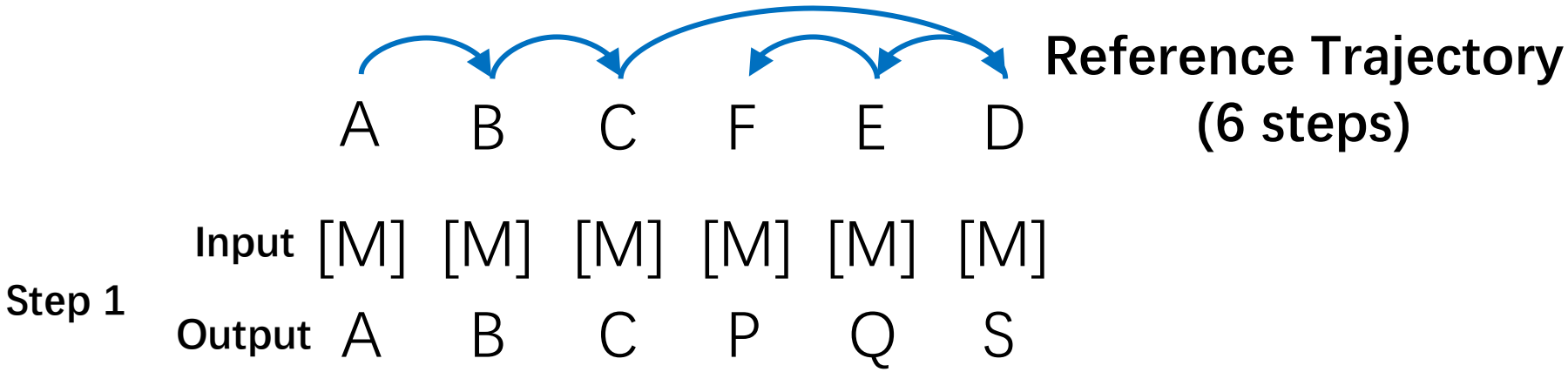
The model predicts many tokens **correctly** at each step but can't sample them due to **low confidence**.



Sample all correct tokens at each step can reduce the sampling step significantly.

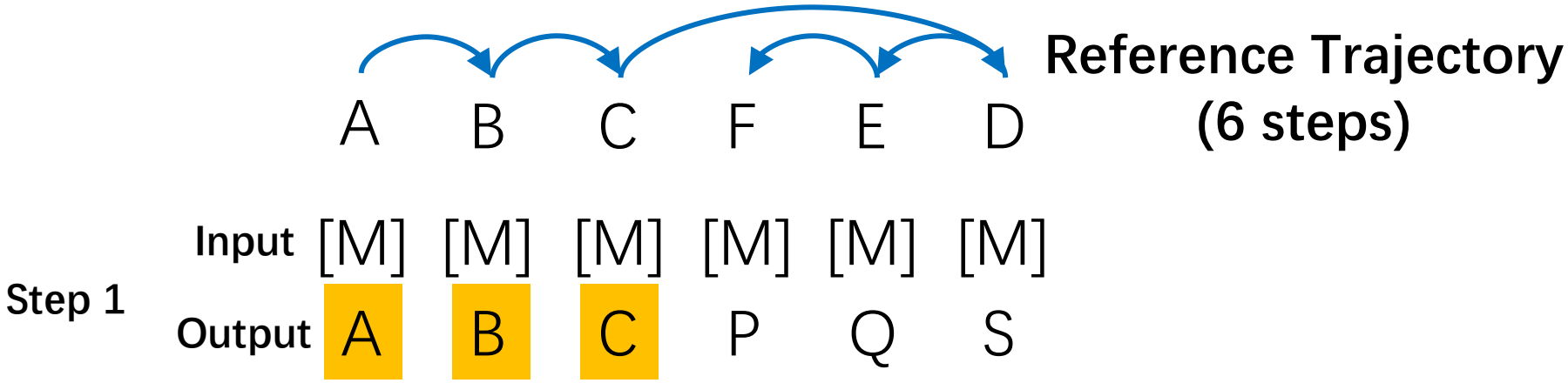
How to Identify Correct Tokens: Using Trajectory Preserving Order

Sample all tokens that are guaranteed not to alter the trajectory at each step.



How to Identify Correct Tokens: Using Trajectory Preserving Order

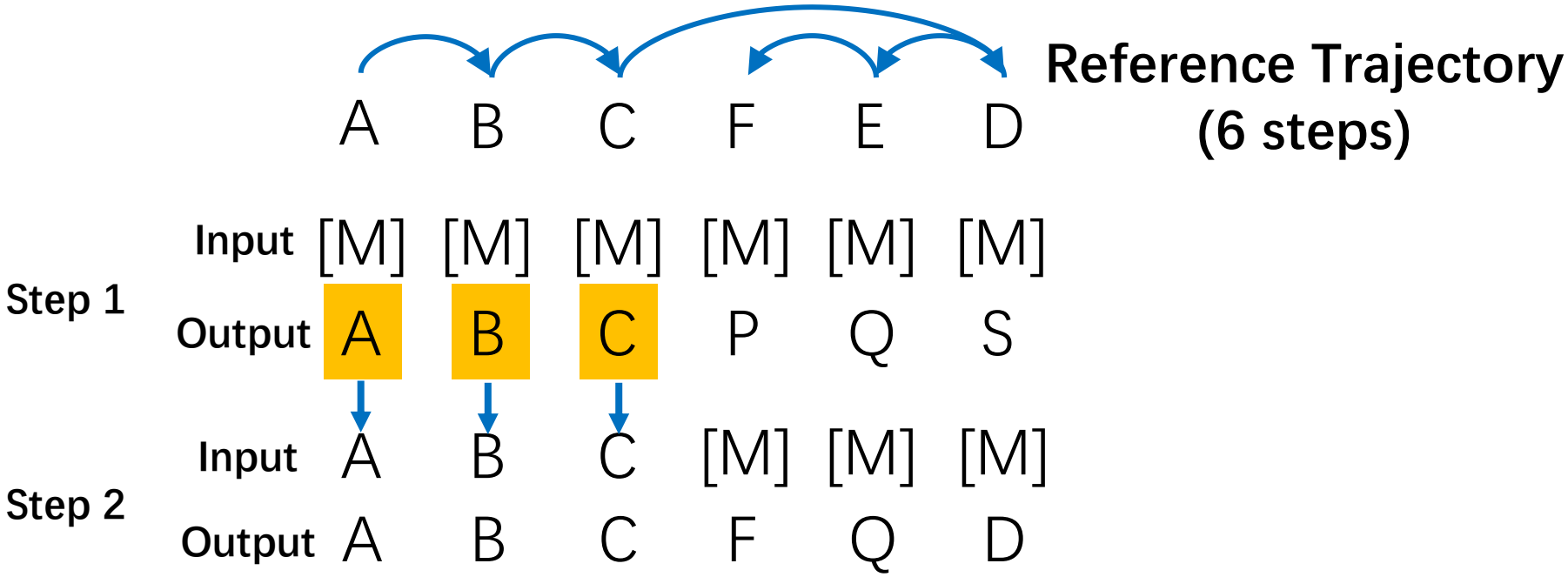
Sample all tokens that are guaranteed not to alter the trajectory at each step.



A, B, C are correctly predicted, and all of them follows the reference order, so they can be sampled

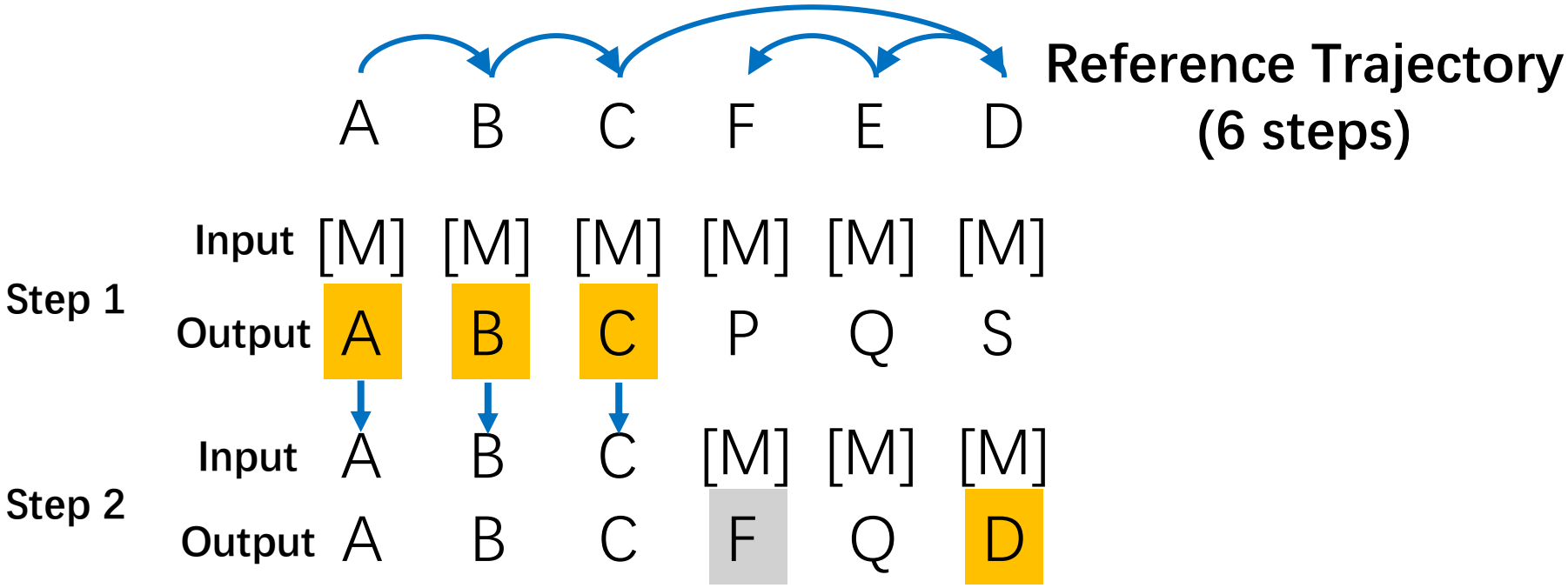
How to Identify Correct Tokens: Using Trajectory Preserving Order

Sample all tokens that are guaranteed not to alter the trajectory at each step.



How to Identify Correct Tokens: Using Trajectory Preserving Order

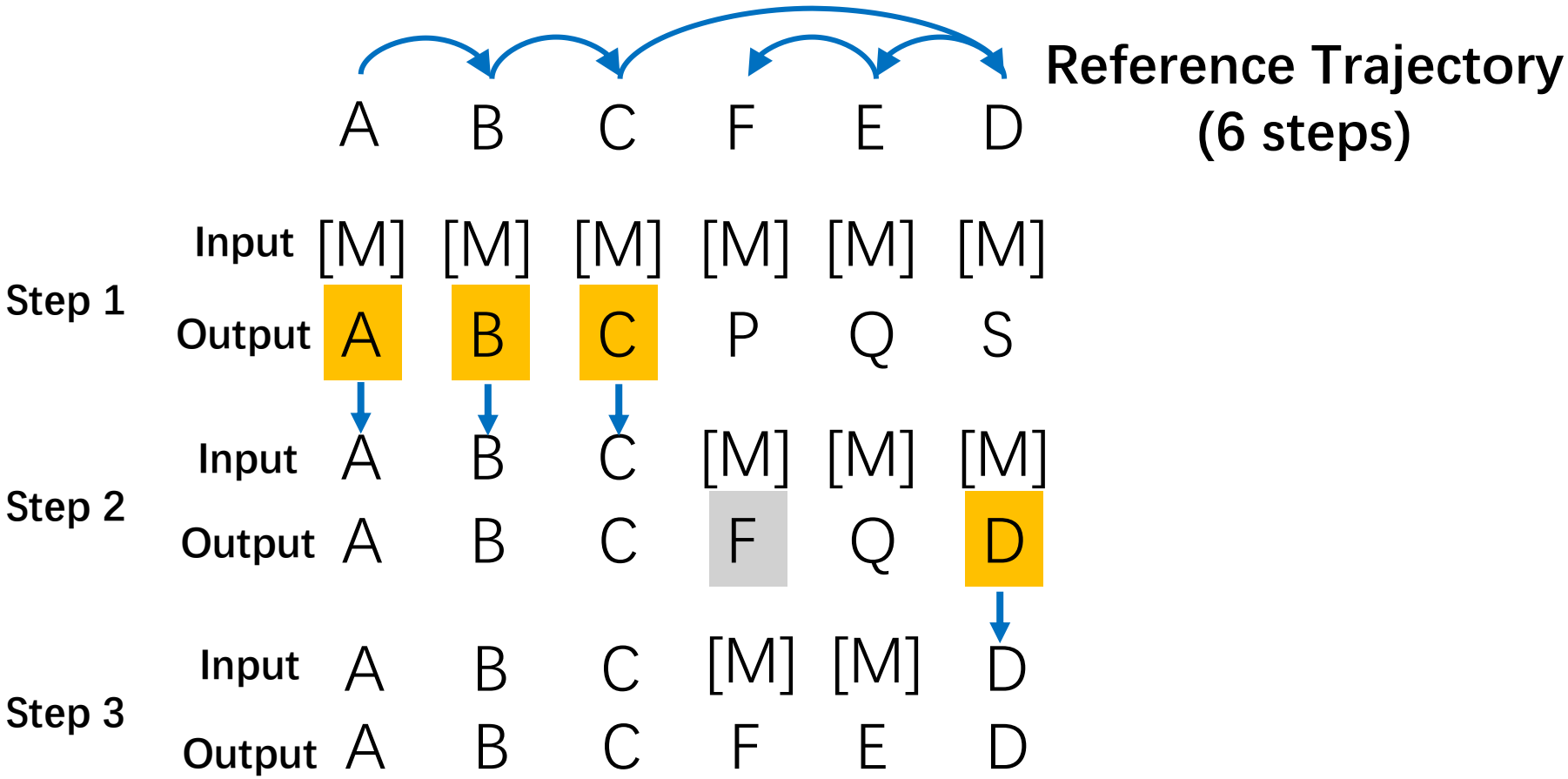
Sample all tokens that are guaranteed not to alter the trajectory at each step.



F, D are correctly predicted, but only D follows the reference order.
F does not follow the order because E is not correctly predicted

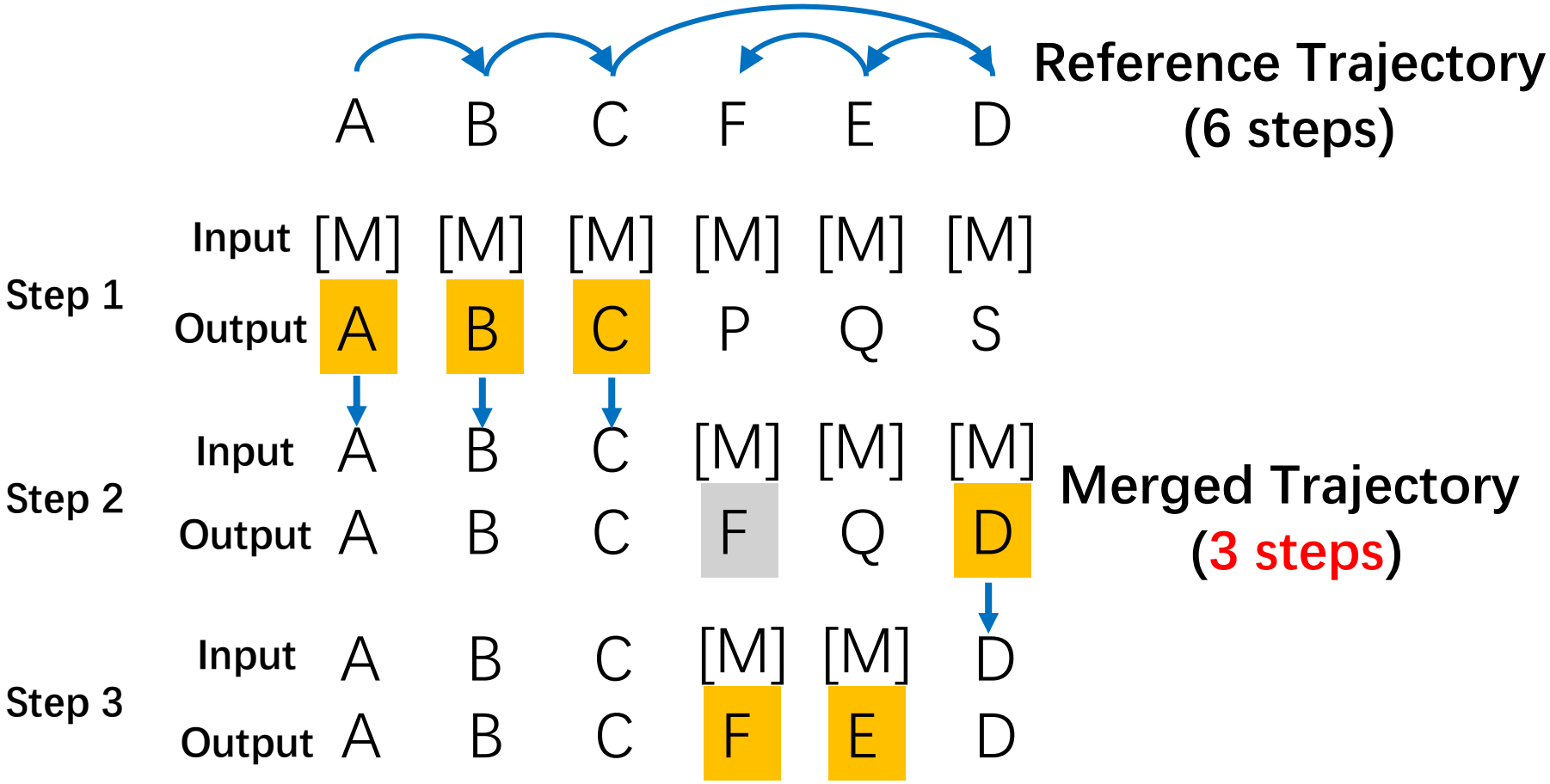
How to Identify Correct Tokens: Using Trajectory Preserving Order

Sample all tokens that are guaranteed not to alter the trajectory at each step.



How to Identify Correct Tokens: Using Trajectory Preserving Order

Sample all tokens that are guaranteed not to alter the trajectory at each step.

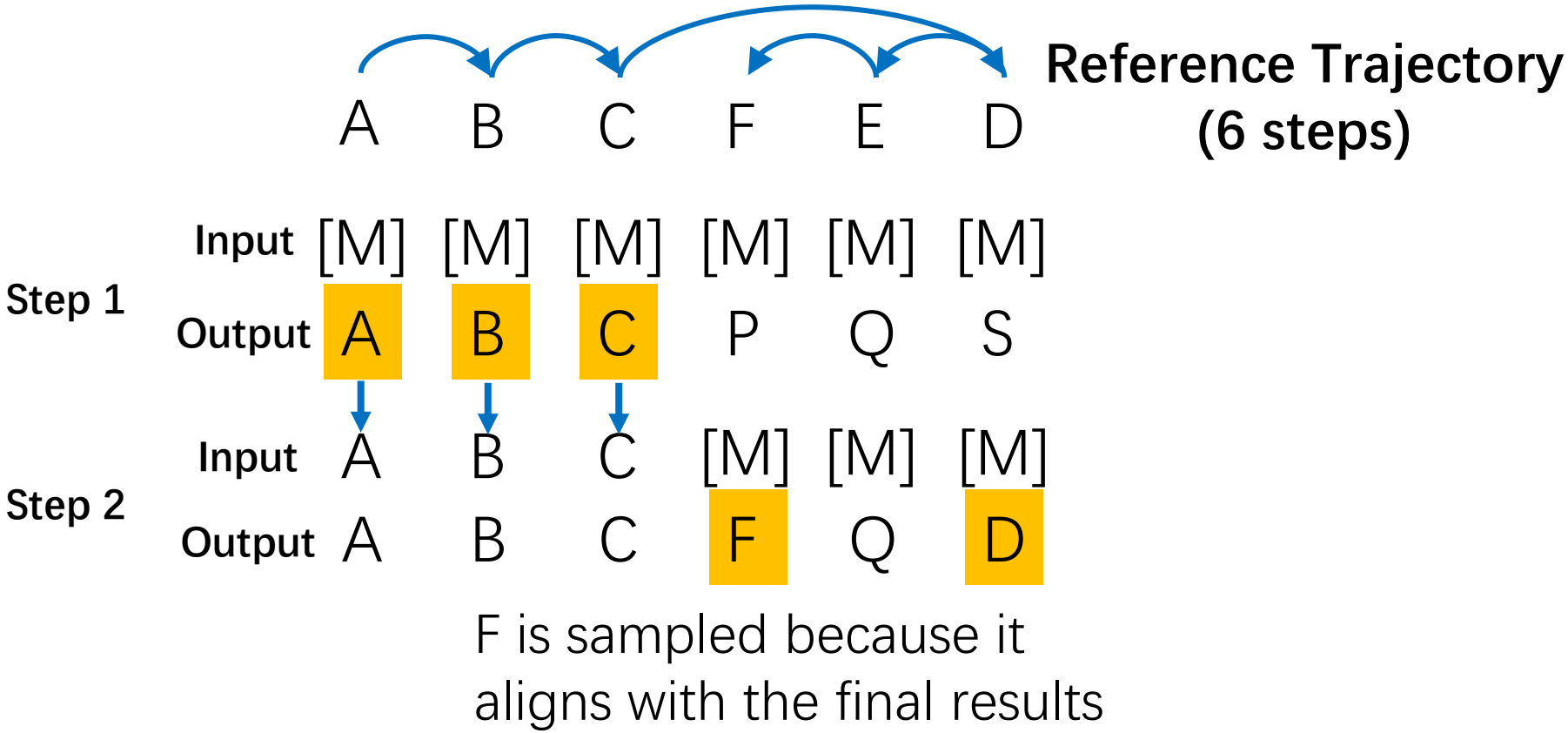


Potential Acceleration of Trajectory Preserving Order

Model	Method	GSM8K-256		MATH-256		MBPP-256	
		Acc	Step	Acc	Step	Acc	Step
LLaDA-8B-Instruct	Full-step	77.63	256	31.89	256	36.60	256
	Threshold	77.33	74.34 (3.4×)	31.68	95.98 (2.7×)	37.60	33.48 (7.6×)
	Traj-Preserving	77.63	22.36 (11.4×)	31.89	27.72 (9.2×)	36.60	10.52 (24.3×)
LLaDA-1.5	Full-step	81.05	256	33.18	256	38.40	256
	Threshold	81.58	72.92 (3.5×)	33.18	95.41 (2.7×)	38.40	41.89 (6.1×)
	Traj-Preserving	81.05	22.31 (11.5×)	33.18	28.39 (9.0×)	38.40	11.28 (22.7×)

A more Aggressive Method: Final Results Preserving Order

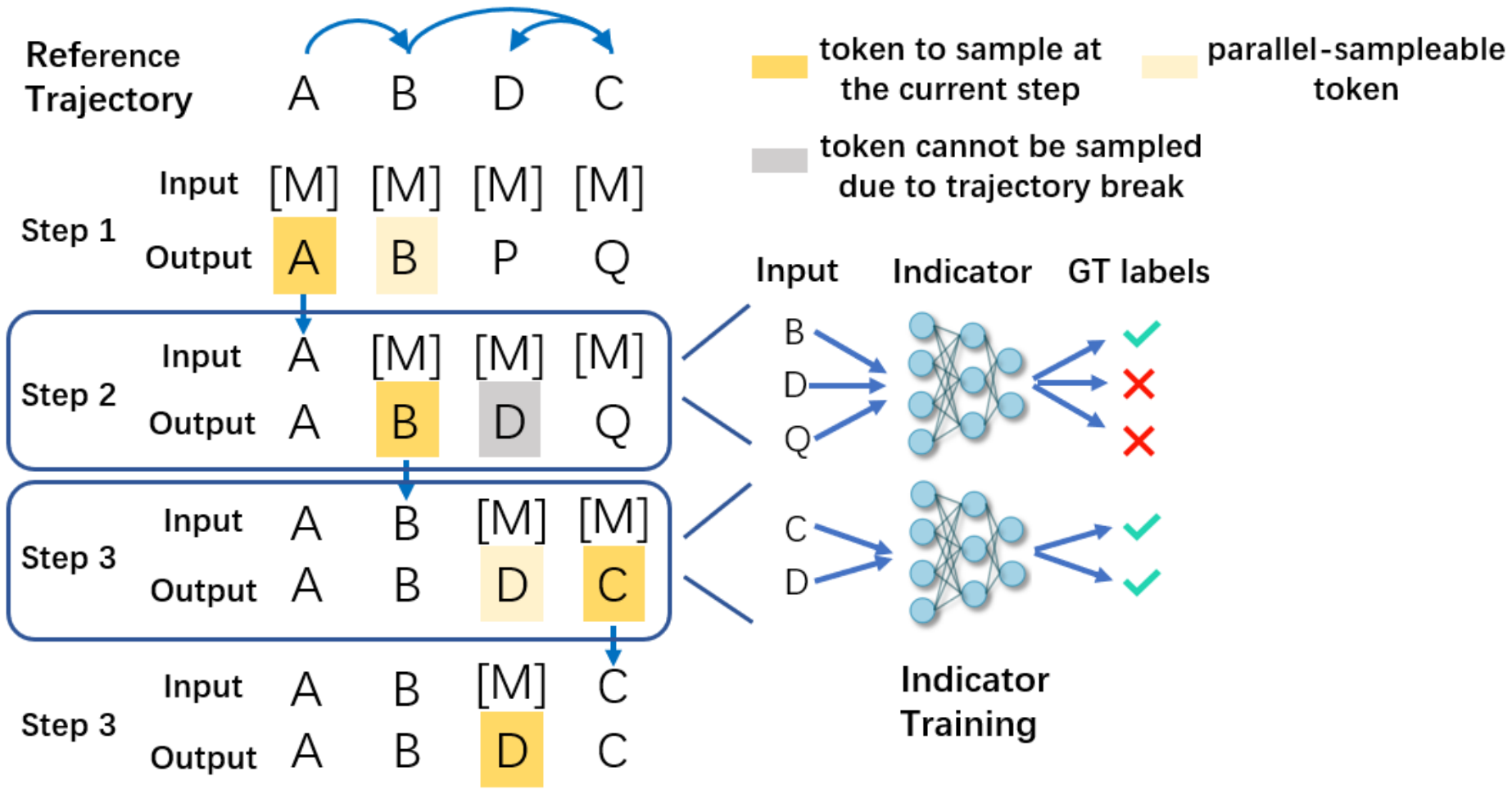
Sample all tokens aligned with the final results, regardless of the order.



Potential Acceleration of Final Results Preserving Order

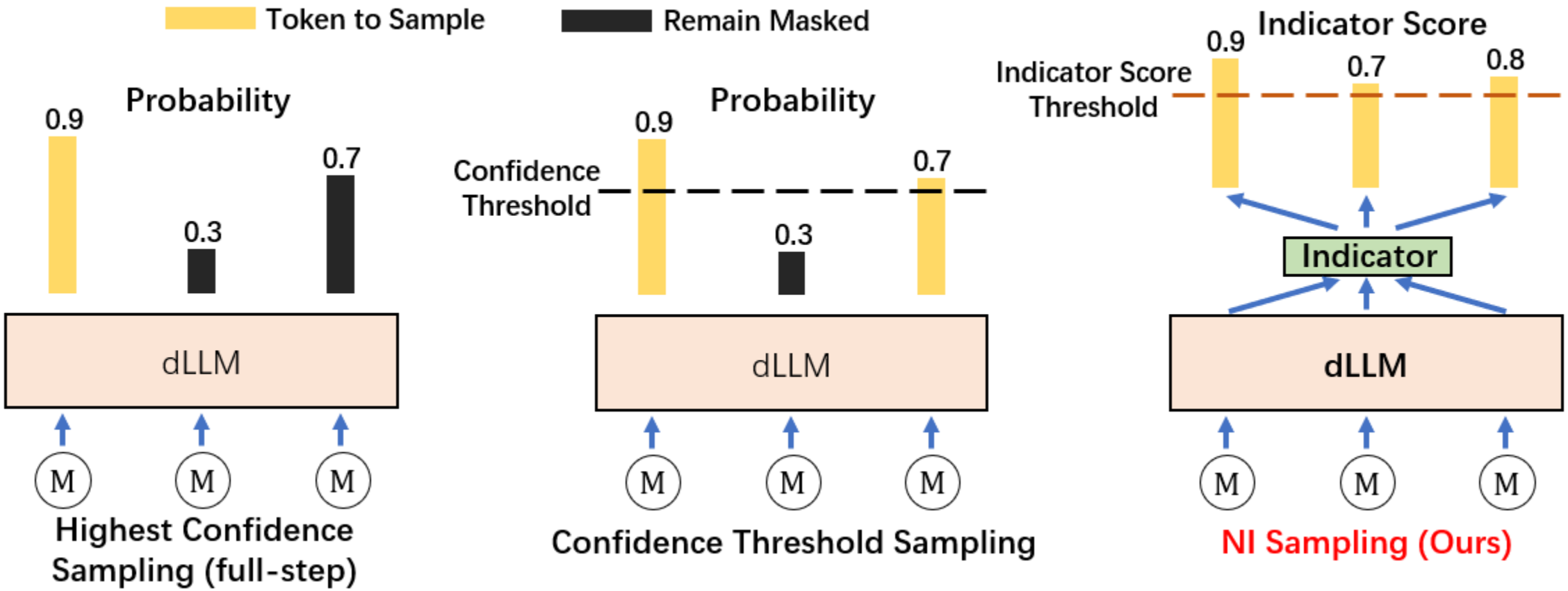
Method	LLaDA-8B-Instruct		LLaDA-1.5	
	Acc	Steps	Acc	Steps
Full-step	77.63	256	81.05	256
Threshold	77.33	74.34 (3.4×)	81.58	72.92 (3.5×)
Trajectory-Preserving	77.63	22.36 (11.4×)	81.05	22.31 (11.5×)
Final-Results-Preserving	77.78	6.95 (36.8×)	80.89	9.11 (28.1×)

Train a Neural Indicator to Judge whether a Token should be Sampled



Based on trajectory preserving order to label every masked position at each step, and train the indicator

Sample with the Neural Indicator



Like confidence threshold sampling, we can control the step-performance trade-off by adjusting score threshold

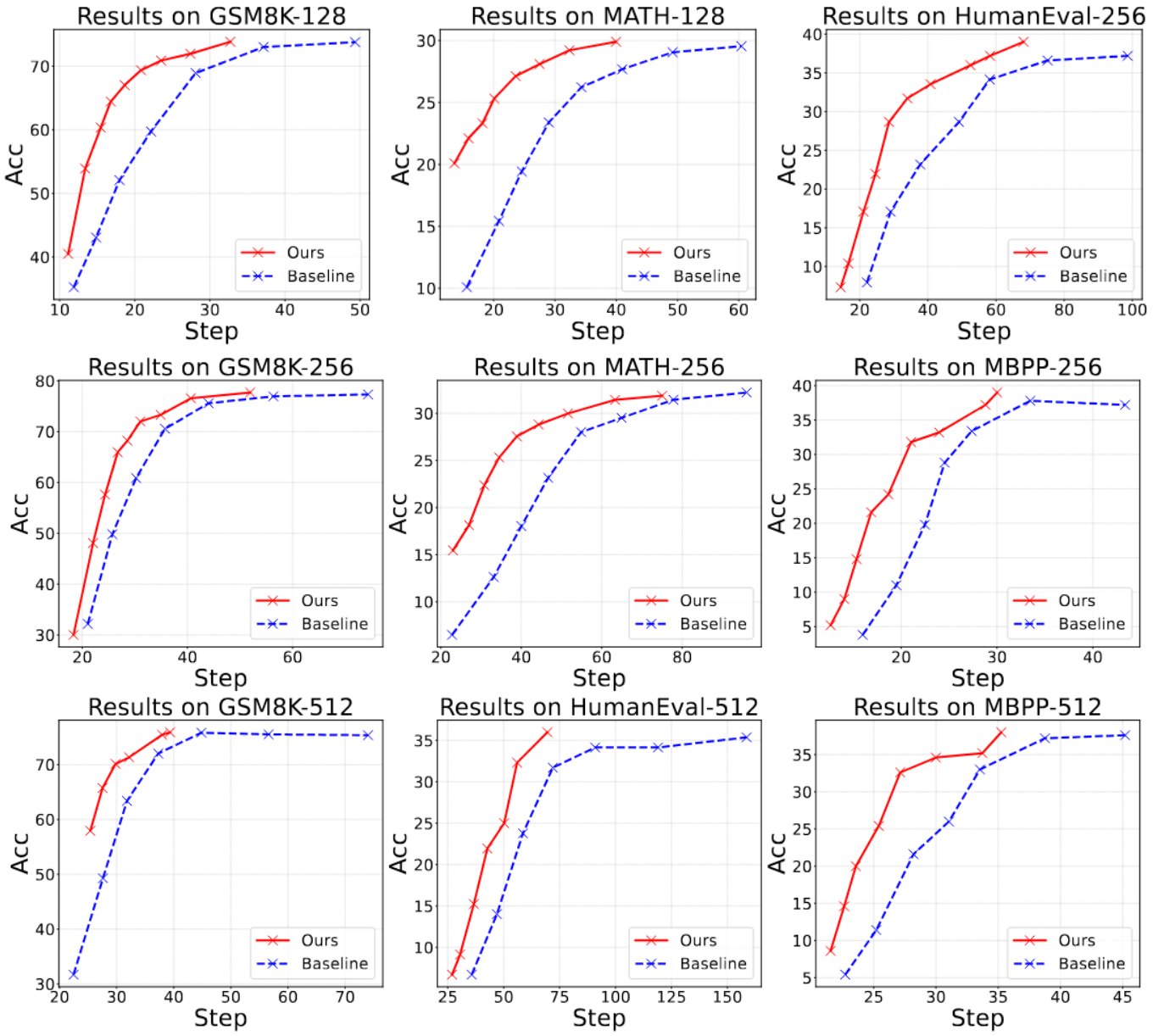
Speedup Results when Performance is Comparable with the Default Full-step Sampling

Dataset	Method	LLaDA-8B-Instruct			LLaDA 1.5		
		Acc	Steps	Token/s	Acc	Steps	Token/s
GSM8K-128	Full	73.92	128	20.4	76.65	128	19.7
	Threshold	73.77	49.35	53.3 (2.6×)	75.82	47.44	53.5 (2.7×)
	NI Sampling	73.84	32.84	80.0 (3.9×)	76.19	29.89	85.3 (4.3×)
GSM8K-256	Full	77.63	256	18.6	81.05	256	18.3
	Threshold	77.33	74.34	62.9 (3.4×)	81.58	72.92	65.0 (3.6×)
	NI Sampling	77.71	51.93	88.4 (4.8×)	82.18	48.41	94.8 (5.2×)
GSM8K-512	Full	74.83	512	14.4	80.67	512	15.0
	Threshold	75.81	44.85	170.6 (11.8×)	81.27	56.01	136.5 (9.1×)
	NI Sampling	75.44	37.98	197.7 (13.7×)	81.12	47.80	157.5 (10.5×)
MATH-128	Full	29.64	128	28.5	31.03	128	27.8
	Threshold	29.69	60.43	57.9 (2.0×)	30.93	58.71	60.9 (2.2×)
	NI Sampling	29.86	39.75	85.3 (3.0×)	30.51	38.16	90.5 (3.3×)
MATH-256	Full	31.89	256	25.0	33.18	256	25.0
	Threshold	31.68	95.98	66.7 (2.7×)	33.18	95.41	67.0 (2.7×)
	NI Sampling	31.86	75.00	81.5 (3.3×)	32.98	69.67	89.5 (3.6×)
HumanEval-256	Full	37.80	256	44.1	43.90	256	44.7
	Threshold	37.20	98.66	115.8 (2.6×)	42.68	100.7	113.3 (2.5×)
	NI Sampling	37.20	55.60	192.5 (4.4×)	42.68	64.40	168.4 (3.8×)
HumanEval-512	Full	35.37	512	31.5	40.85	512	32.1
	Threshold	35.37	158.6	100.8 (3.2×)	39.02	158.4	101.1 (3.1×)
	NI Sampling	35.98	69.54	219.3 (7.0×)	39.63	105.3	144.2 (4.5×)
MBPP-256	Full	36.60	256	22.8	38.40	256	23.3
	Threshold	37.60	33.48	176.0 (7.7×)	38.40	41.89	143.0 (6.1×)
	NI Sampling	39.00	30.02	192.8 (8.5×)	38.60	28.92	201.8 (8.7×)
MBPP-512	Full	36.80	512	19.1	37.80	512	18.7
	Threshold	37.60	38.75	250.2 (13.1×)	38.60	44.38	215.8 (11.5×)
	NI Sampling	38.00	35.25	263.9 (13.8×)	38.80	34.62	268.0 (14.3×)

Up to **2.2x** faster than confidence threshold sampling

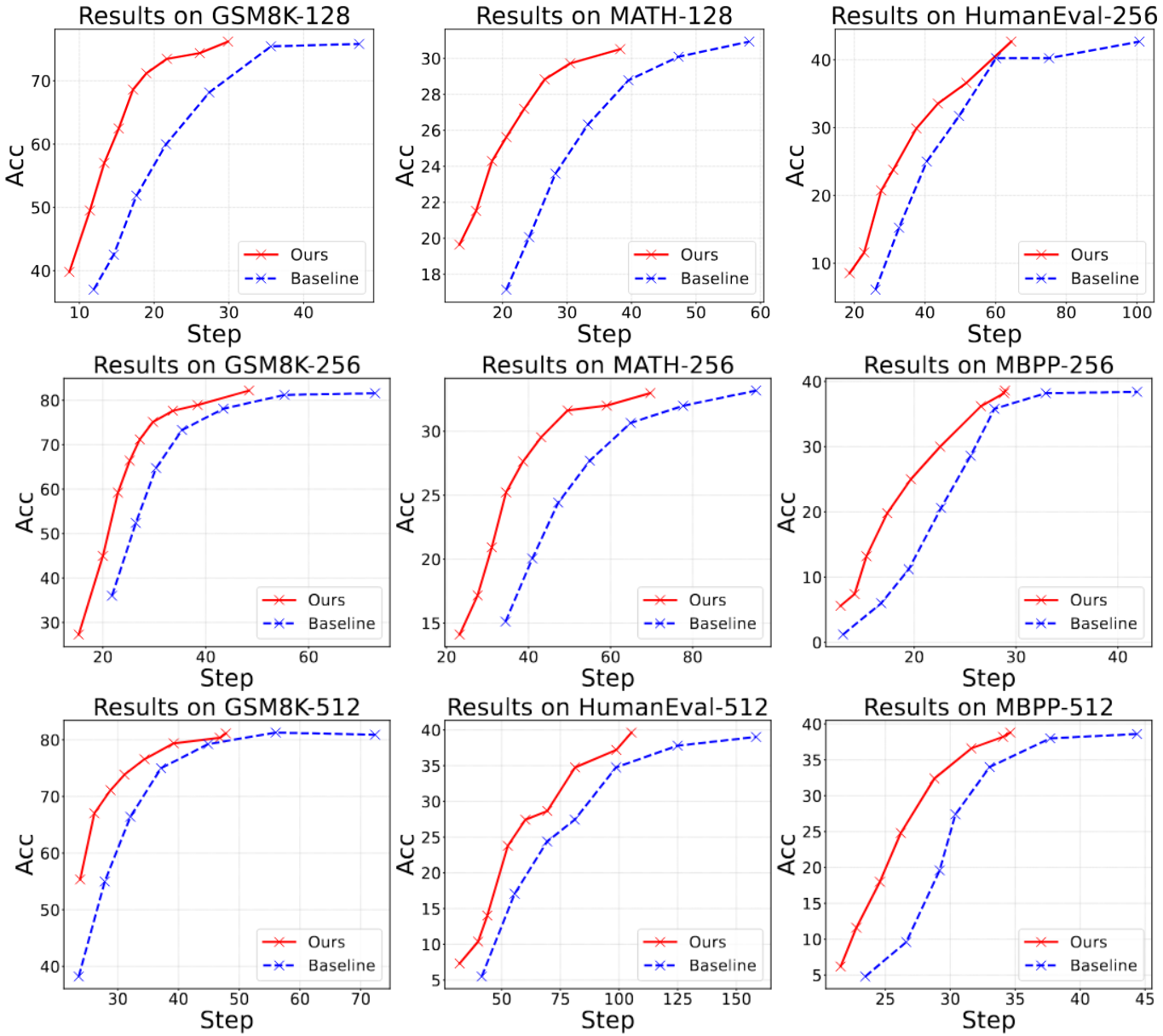
Trade-off Curves between Performance and Step

With LLaDA-8B-Instruct Model



Trade-off Curves between Performance and Step

With LLaDA-1.5 Model



Works Well with Random Sampling

With LLaDA-8B-Instruct Model

On HumanEval dataset

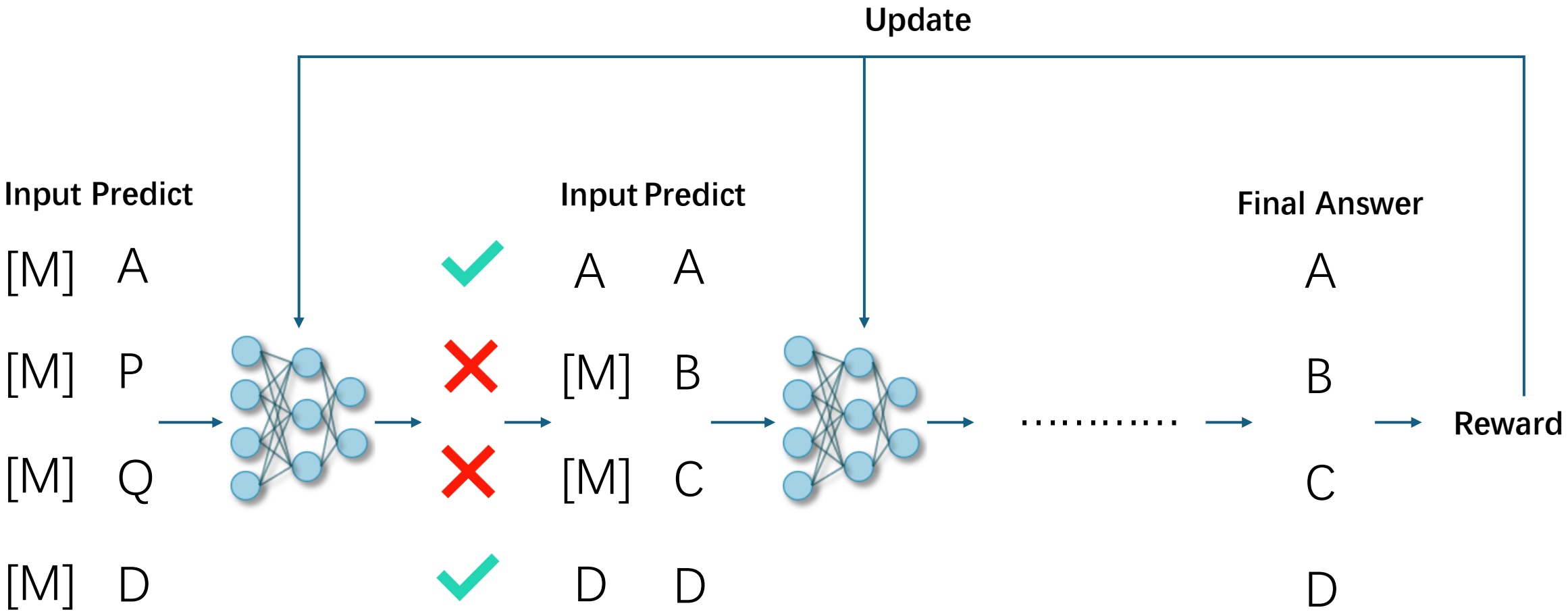
	Token/s	NFE	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16	**Slope↑**
Full-step	42.91	256	36.99	44.69	51.43	57.81	63.76	6.67
Threshold	110.82	99.13	36.82	44.35	51.02	57.40	63.10	6.56
Old Indicator	164.09	63.34	36.43	44.13	51.09	57.63	63.62	6.79
New Indicator	155.78	66.72	36.70	44.38	51.34	58.03	64.33	**6.89**

On GSM8K dataset

	Token/s	NFE	Pass@1	Pass@2	Pass@4	Pass@8	**Slope↑**
Full-step	19.11	256	76.95	84.83	89.99	93.48	5.47
Threshold	65.56	74.63	77.55	85.51	91.05	93.85	5.44
Old Indicator	86.74	54.70	77.78	85.97	91.05	93.70	5.28
New Indicator	82.36	57.68	77.33	84.91	90.83	94.47	**5.73**

Old Indicator: trained with data generated greedily
New Indicator: trained with data generated randomly

Follow-up Ideas: Use RL to Optimize the Neural Indicator



Use RL to Optimize the Token Selector

I would like to try sampling 1 token/step first to see how good it can be.
Selector gives scores for each token, and sample a position to unmask