Test-Time Scaling with Diffusion Language Models via Reward-Guided Stitching

Roy Miles Aysim Toker Andreea-Maria Oncescu Jiankang Deng Ismail Elezi

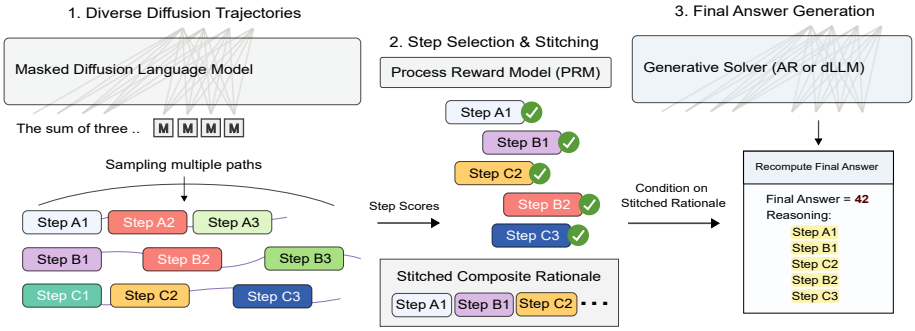
MVP Lab

NeurIPS 2026

Motivation

- LLMs benefit from sampling many chains of thought at test time.
- Existing aggregation (e.g. majority vote, best-of-N) works at the *trajectory* level.
- Partially correct attempts are thrown away, along with their useful intermediate steps.
- Masked diffusion LMs make sampling many diverse traces cheap.
- **Idea:** recombine the best *steps* across traces instead of picking one whole trace.

Overview: Stitching Noisy Diffusion Thoughts



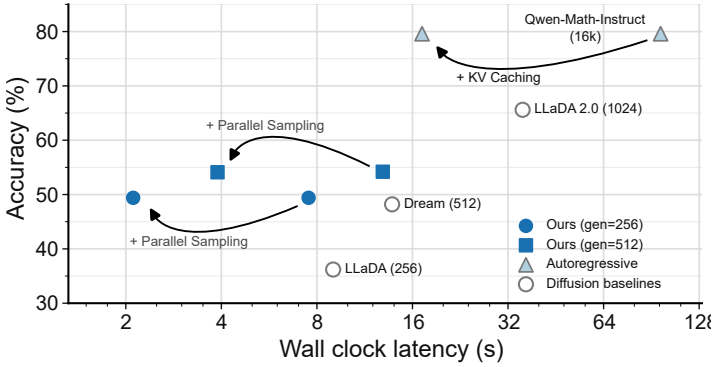
Source: paper.pdf, Figure 1, p. 2

- Diffusion explores diverse reasoning paths; a PRM scores each step.
- Best steps are stitched into one rationale; a solver recomputes only the final answer.
- Modular and training-free: no unified hybrid architecture needed.

Method

1. **Explore:** sample many low-cost reasoning traces with a masked diffusion LM (LLaDA / Dream).
2. **Evaluate:** score each intermediate step with an off-the-shelf process reward model.
3. **Stitch:** combine the highest-scoring steps across traces into a composite rationale.
4. **Solve:** an autoregressive solver conditions on the rationale and produces the final answer.
5. **Coding:** the same pipeline extends to code generation tasks.

Accuracy vs. latency (MATH500)



Source: paper.pdf, Figure 2, p. 4

- Compared against autoregressive and diffusion baselines.
- Stitching reaches higher accuracy at lower latency.

Main results

		Coding				Math		Avg
Model Arch	**Size**	**HumanEval**	**HumanEval+**	**MBPP**	**MBPP+**	**GSM8K**	**Minerva Math**	
Qwen3	4B	57.32	50.61	67.00	80.69	77.48	47.10	63.37
Qwen3	8B	64.63	56.71	69.40	83.07	81.80	52.94	68.09
LLaDA	8B	32.32	27.44	40.80	51.85	70.96	27.30	41.78
↳ w/ conf sampling	8B	40.24	37.80	41.89	55.82	78.77	36.54	48.51
Dream	7B	54.88	49.39	56.80	74.60	77.18	39.60	58.74
↳ w/ conf sampling	7B	59.76	57.32	59.14	71.96	81.96	42.78	62.15
Block Diff	4B†	56.10	51.22	54.60	69.84	82.87	47.02	60.27
TiDAR (Trust AR)	8B‡	55.49	52.44	65.40	79.63	79.83	50.58	63.90
TiDAR (Trust Diff)	8B‡	57.93	55.49	65.40	80.95	80.44	51.64	65.31
Ours*	8B	**70.37**	**64.02**	**74.61**	**81.75**	**91.51**	**53.20**	**72.38**

Source: paper.pdf, Table 1, p. 6

- Baselines: Qwen3, LLaDA, Dream, Block Diffusion, TiDAR.
- Up to 23.8% higher average accuracy; up to 1.8× lower latency.

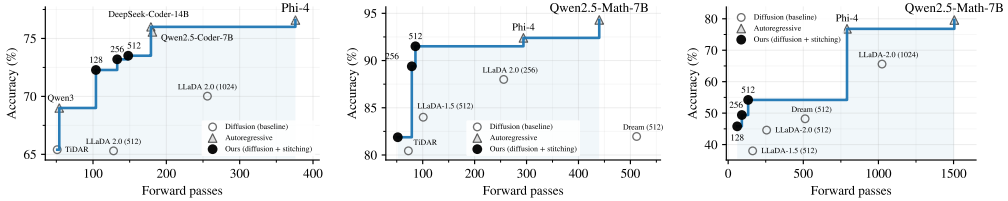
Comparison with autoregressive test-time scaling

Setting	Len.	GSM8K Acc. (%)	GSM8K Lat. (s)	MATH500 Acc. (%)	MATH500 Lat. (s)	Mem. (GB)
Ours (Qwen PRM)	–	91.3	2.585	54.3	4.735	42.8 / 42.7
Ours (Skywork PRM)	–	89.4	2.421	51.8	3.910	32.3 / 32.2
Greedy						
Qwen2.5-Math-7B	256	32.0	2.545	18.0	4.989	14.4 / 14.4
	512	84.0	3.250	49.4	8.869	14.4 / 14.4
	1024 / 768	90.0	3.829	64.2	9.849	14.5 / 14.5
Self-consistency ($N = 4$)						
Qwen2.5-Math-7B	256	34.0	2.798	20.0	5.310	14.4 / 14.4
	512	90.0	3.932	56.4	9.210	14.4 / 14.5
	1024 / 768	92.0	4.361	69.6	10.130	14.5 / 14.5
PRM best-of-4						
Qwen2.5-Math-7B	256	34.0	2.990	18.0	5.311	27.9 / 28.0
	512	88.0	4.064	62.0	9.098	28.0 / 28.1
	1024 / 768	92.0	4.239	69.8	10.989	28.3 / 28.3

Source: paper.pdf, Table 2, p. 6

- Accuracy and latency on GSM8K and MATH500.

Compute efficiency



(a) MBPP (b) GSM8K (c) MATH500

Source: paper.pdf, Figure 3, p. 6

- Pareto fronts at generation lengths 128, 256 and 512.

Why the solver matters

Setting	GSM8K		MATH500	
	Acc	#steps	Acc	#steps
Stitching (4 traces)				
↳ AR	91.5	96.9	54.2	135.5
↳ dLLM $\gamma = 0.7$	90.1	106.3	50.2	132.6
↳ dLLM $\gamma = 0.9$	90.3	141.6	52.0	186.5
Stitching (8 traces)				
↳ AR	91.4	110.8	55.0	141.1
↳ dLLM $\gamma = 0.7$	91.4	115.8	52.6	138.6
↳ dLLM $\gamma = 0.9$	92.5	151.8	54.4	198.2

Source: paper.pdf, Table 3, p. 8

- The final solver turns imperfect stitched rationales into accurate answers.

Ablation: stitching strategies

Setting	GSM8K	MATH500
Baseline		
LLaDA	78.8	37.6
without Solver or PRM		
Majority vote [31]	85.1	42.0
with Solver and PRM		
All CoT	86.6	46.0
Best CoT	90.1	49.2
with Solver, PRM, and Stitching		
Above confidence	90.4	52.0
Best CoT and above confidence	**91.5**	**54.2**

Source: paper.pdf, Table 4, p. 8

- Step-level recombination helps most on harder problems.

Ablations: confidence threshold and number of generations

Confidence threshold

Confidence	GSM8K		MATH500	
	Acc	#steps	Acc	#steps
0.4	87.6	76.6	46.8	103.4
0.5	89.1	79.8	49.2	108.7
0.6	90.5	86.8	52.0	119.8
0.7	91.5	96.9	54.2	135.5
0.8	91.7	111.3	54.4	158.5

Source: paper.pdf, Table 5, p. 8

Number of generations

	GSM8K	MATH500	Avg
1	82.4	46.0	64.2
2	88.1	49.2	68.7
4	91.5	54.2	72.9
6	91.4	54.3	72.9
8	91.3	55.0	73.2

Source: paper.pdf, Table 6, p. 8

- Sensitivity to the diffusion sampling confidence threshold.
- Scaling with the number of diffusion generations.

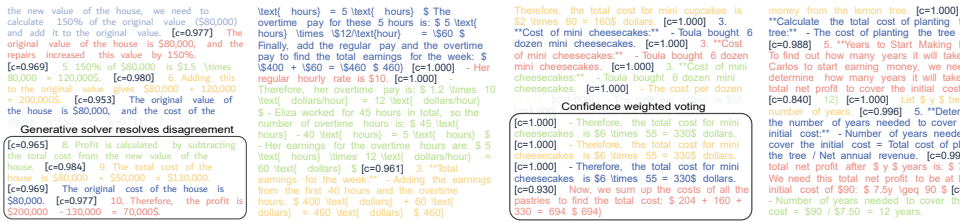
Ablation: choice of models

Model	GSM8K		MATH500	
	Acc	#steps	Acc	#steps
Qwen-Math-Instruct	94.3	440	79.6	1504
Phi-4	92.4	295	76.8	791
LLaDA-1.5	84.0	101	38.0	163
↳ w/ stitching	**90.8**	87	**53.8**	134
Dream	82.6	512	48.2	512
↳ w/ stitching	**85.4**	521	**53.2**	523
LLaDA 2.0	88.0	256	44.6	256
↳ w/ stitching	**89.6**	266	**61.4**	267

Source: paper.pdf, Table 7, p. 9

- Pipeline is modular: explorer and solver can be swapped.

Qualitative examples



Source: paper.pdf, Figure 4, p. 7

- Disagreements between traces are resolved at the step level.
- PRM confidence scores guide the final prediction.

Limitations

- Relies on the quality of an off-the-shelf process reward model.
- Failure cases occur when the PRM scores wrong steps highly.

Conclusion

- Reuse step-level reasoning from many cheap diffusion-sampled traces.
- Separate exploration (diffusion), evaluation (PRM) and synthesis (AR solver).
- Training-free, with gains in both accuracy and latency.
- Code: `https://github.com/roymiles/diffusion-stitching`