



NEURIPS 2026 · SYDNEY / TMLR 2025

Reasoning Under 1 Billion

Memory-Augmented Reinforcement Learning for Large Language Models

Hung Le · Van Dai Do · Dung Nguyen · Svetha Venkatesh

Applied Artificial Intelligence Initiative, Deakin University



DEAKIN
UNIVERSITY

DEAKIN
**APPLIED ARTIFICIAL
INTELLIGENCE INITIATIVE**



THE PROBLEM

Small LLMs rarely get an answer right

GRPO SAMPLES 16 ANSWERS PER QUESTION

A large LLM gets many answers right.



An LLM under 1B often gets none right.



Filled dots are correct answers. Empty dots are wrong answers.

GRPO compares the answers to each question.

It learns only when some answers are right and some are wrong.

Small LLMs often get every answer wrong. Then the update teaches **nothing**.



THE PROBLEM

Sparse rewards make RL stall

01

Sparse reward

Most answers are wrong early in training. So most updates teach nothing.

02

Repeated mistakes

The model keeps sampling the same failed reasoning. It rarely tries a new path.

03

Training collapse

Some runs game the format rewards. Others shrink answers to digits or grow them to the length limit.

Small LLMs need a denser learning signal.



KEY IDEA

Remember what **worked**. Steer away from what **failed**.

SUCCESS MEMORY

It keeps correct answers. New answers that look like them earn more reward.

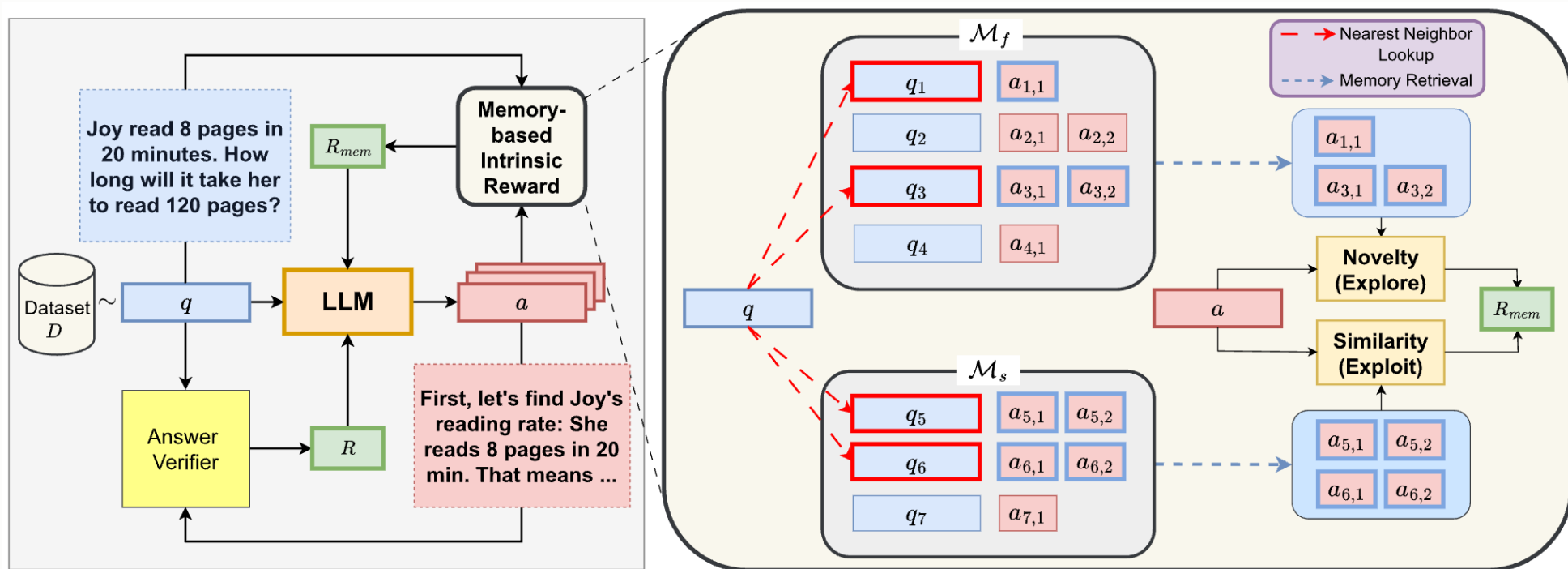
FAILURE MEMORY

It keeps wrong answers. New answers that differ from them earn more reward.

So every model's answer gets a training signal, even when none is correct.

METHOD

Memory-R+ adds a memory reward to GRPO



Left: the RL loop. The model gets an outcome reward and a memory reward. **Right:** one question looks up both memories, and the answer is compared with what they store.



Write: each answer goes to one memory

QUESTION q

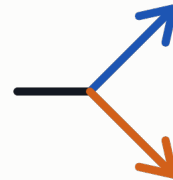
Joy reads 8 pages in 20 minutes. How long will 120 pages take?

16 SAMPLED ANSWERS

a1	a2	a3	a4
a5	a6	a7	a8
a9	a10	a11	a12
a13	a14	a15	a16

Three answers are correct. Thirteen are wrong.

The verifier
sorts each
answer.



SUCCESS MEMORY

q1			
q2			
q	a3	a7	a12 new

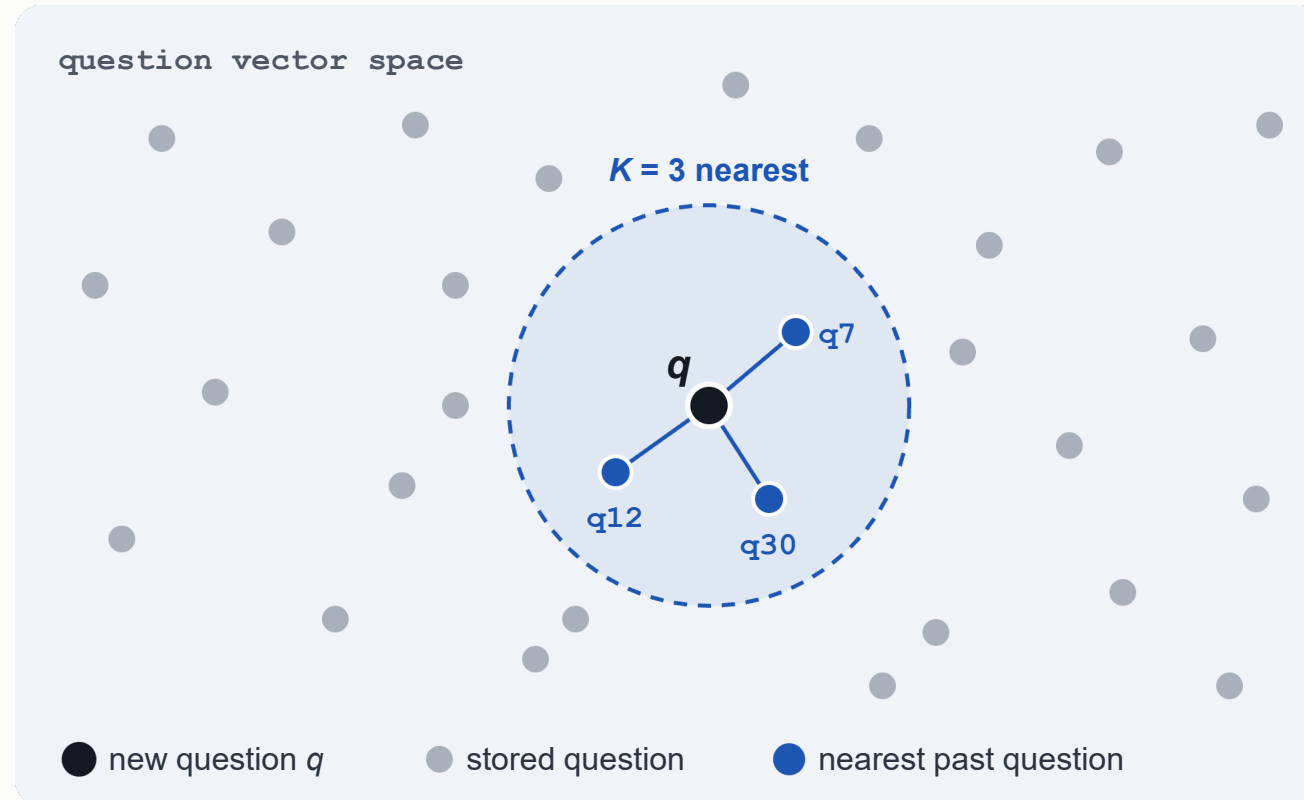
FAILURE MEMORY

q1						
q2						
q	a1	a2	a4	a5	a6	+ 8 more

Each memory keeps up to 100 answers per question. When a list is full, the oldest answer leaves.



Retrieve: A new question finds similar past questions



The encoder turns each question into a vector.

kNN finds the K closest past questions.

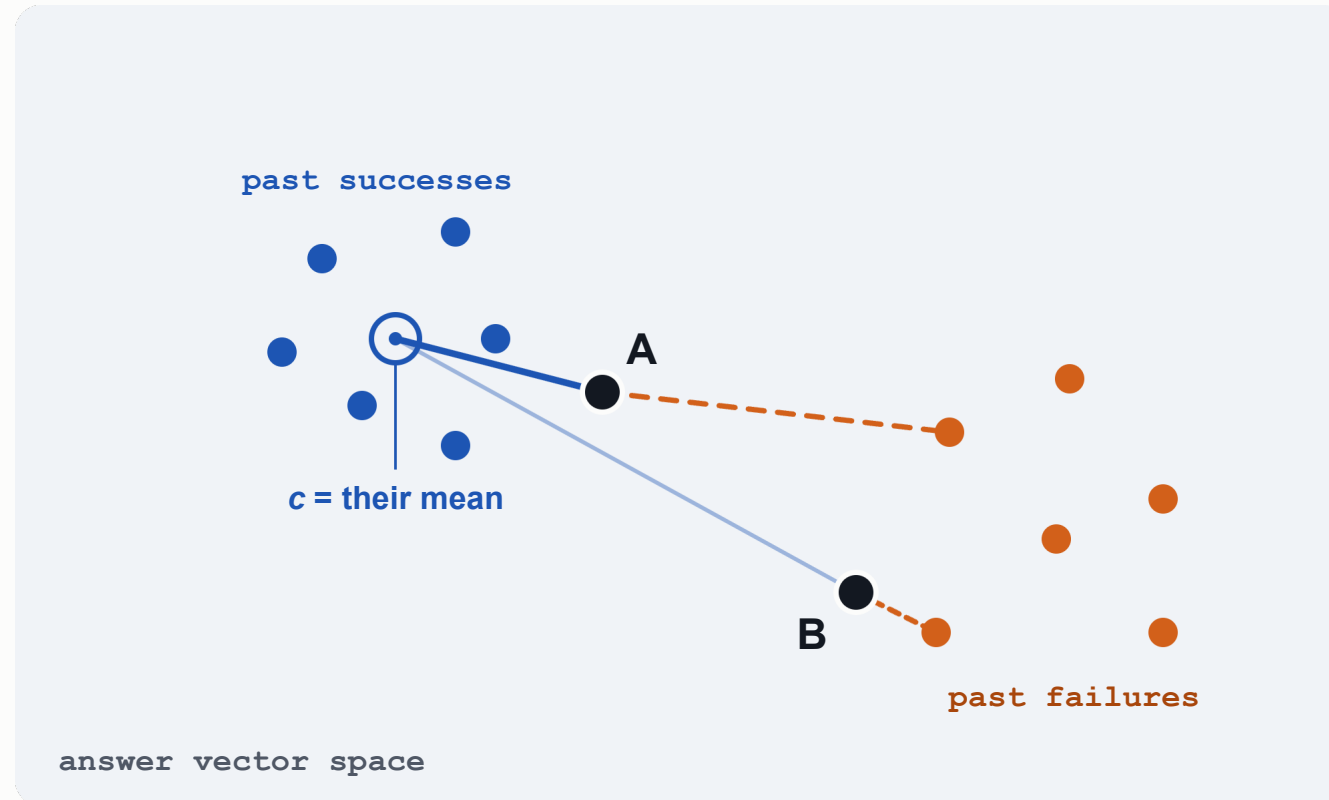
RETRIEVED SET B

q_7	■	■	■	■	■
q_{12}	■	■	■	■	■
q_{30}	■	■	■	■	

Blue squares are past successes. Orange squares are past failures.



Score: each answer is scored against both memories



Answer A

high reward

Exploit



Explore

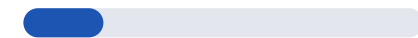


A is close to past successes. It is far from past failures.

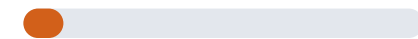
Answer B

low reward

Exploit



Explore



B is far from past successes. It sits next to a past failure.



Learn: memory gives a signal when all answers fail

Without memory (R1)

a1	a2	a3	a4	a5	a6	a7	a8
----	----	----	----	----	----	----	----

OUTCOME REWARD

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

ADVANTAGE

All advantages are 0.

Every reward is 0, so every advantage is 0. GRPO learns nothing.

With memory (Memory-R+)

a1	a2	a3	a4	a5	a6	a7	a8
----	----	----	----	----	----	----	----

MEMORY REWARD

0.8	0.3	0.6	0.1	0.9	0.4	0.2	0.7
-----	-----	-----	-----	-----	-----	-----	-----

ADVANTAGE



Answers near successes get a positive advantage. Answers near failures get a negative one.

Illustrative values. GRPO's advantage is each reward minus the group mean, divided by the group's spread.



METHOD: DETAILS ON REWARDS

Two rewards guide the model

SIMILARITY (EXPLOIT) · SUCCESS MEMORY

$$R_{\text{exploit}}(q, a) = -\|a - c(\mathcal{M}_s, q)\|$$

The reward is high when the answer is close to the mean of past successes.

NOVELTY (EXPLORE) · FAILURE MEMORY

$$R_{\text{explore}}(q, a) = 1 - \max_{a_j \in B(\mathcal{M}_f, q)} \cos(a, a_j)$$

The reward is high when the answer is far from its most similar past failure.

OBJECTIVE · OPTIMISED WITH GRPO

$$\max_{\theta} \mathbb{E}[R + R_{\text{mem}}], \quad R_{\text{mem}} = \beta_s \hat{R}_{\text{exploit}} + \beta_e \hat{R}_{\text{explore}}$$

Rewards are min-max normalised.
Both weights equal 1.



EXPERIMENTS

We test three small models

Models

We use three Instruct models: Qwen2.5-0.5B, Falcon3-1B and Llama-3.2-1B.

Training data

We train on GSM8K, on the harder AI-MO set and on Knights & Knaves puzzles.

Evaluation

We test on GSM8K, MATH-500 and AIME24. We report extractive-match accuracy (%).

Training

GRPO samples 16 answers per question, up to 200 tokens each. We average 3 seeds.

Baselines

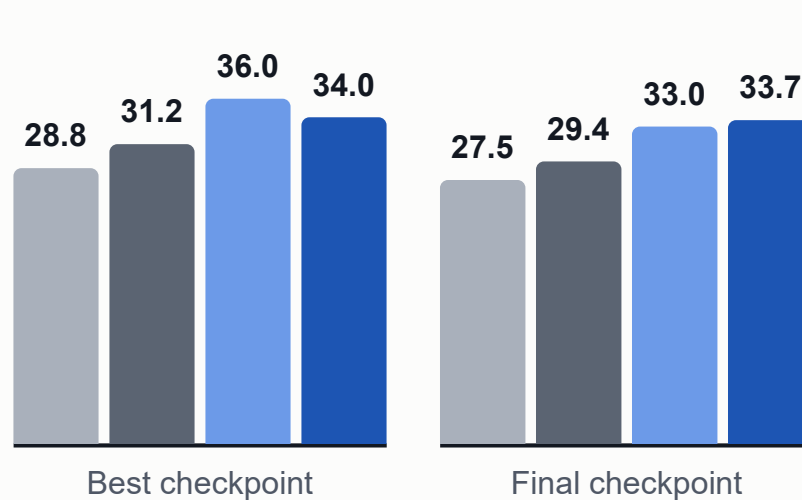
R1 uses correctness and format rewards. **Cosine** adds a length-based reward. **Memory-R** is our ablation. It uses only the exploit reward.



Memory-R and Memory-R+ lead on GSM8K

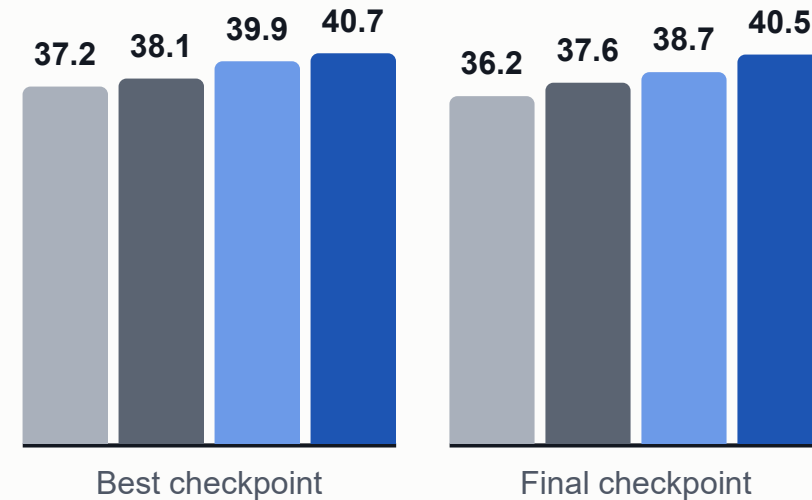
Both memory variants beat both baselines at the best and the final checkpoint. Falcon3-1B is on the next slide.

■ R1 ■ Cosine ■ Memory-R ■ Memory-R+



Qwen2.5-0.5B

The memory variants lead by 4.8 and 4.3 points.



Llama3.2-1B

The memory variants lead by 2.6 and 2.9 points.



Memory rewards beat R1 on every model

The table shows the best checkpoint after GSM8K training. Cosine leads on Falcon3-1B.

Qwen2.5-0.5B			Falcon3-1B		Llama3.2-1B	
Method	GSM8K	MATH-500	GSM8K	MATH-500	GSM8K	MATH-500
Base	27.8	20.0	32.9	12.2	26.3	17.4
R1	28.8 ± 7.5	18.9 ± 3.8	16.3 ± 1.7	10.8 ± 0.4	37.2 ± 1.3	19.0 ± 0.5
Cosine	31.2 ± 0.7	22.7 ± 1.0	37.4 ± 1.3 *	17.0 ± 0.9 *	38.1 ± 1.8	21.2 ± 0.3 *
Memory-R	36.0 ± 2.6 *	23.7 ± 1.3	36.3 ± 0.6	14.1 ± 0.9	39.9 ± 1.2	21.1 ± 0.7
Memory-R+	34.0 ± 2.3	24.4 ± 0.6 *	34.8 ± 0.5	16.9 ± 1.3	40.7 ± 1.1 *	20.6 ± 1.4

* marks the best trained method in each column. Scores are mean ± std over 3 seeds. AIME24 stays near zero for every method, so we leave it out here.



The gains hold on harder problems

We train Qwen2.5-0.5B on the harder AI-MO set and test after each run.

Memory-R+ beats R1 and Cosine by about:

+5

points on GSM8K

+4

points on MATH-500

+1

point on AIME24

More neighbours help. Accuracy (%) for each K:

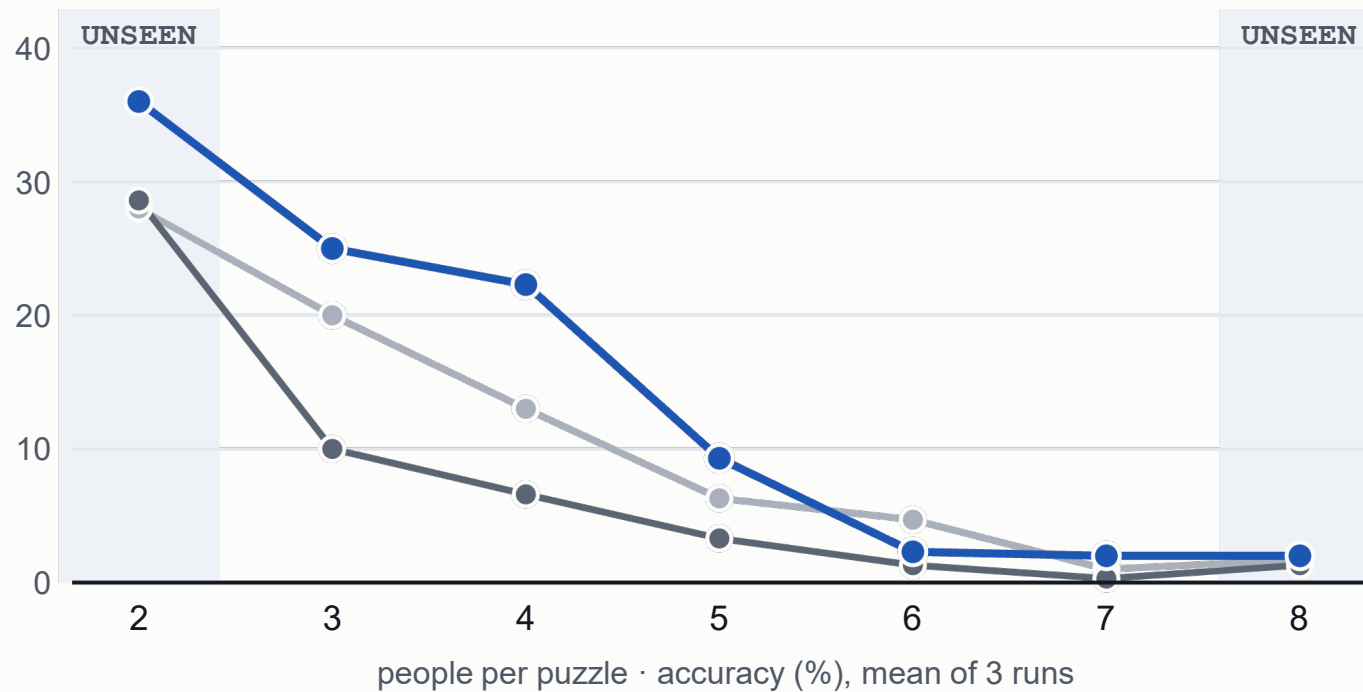
Score	K = 1	K = 10	K = 20	K = 30
GSM8K, best	37.6	38.7 *	38.2	38.4
GSM8K, final	36.5	37.8 *	37.5	36.8
MATH-500, best	25.5	25.9 *	25.8	25.8
MATH-500, final	22.7	23.7	25.3 *	25.0

* marks the best K in each row. Each score is the mean of 3 runs. On AIME24, K = 20 scores 2.2%, and the other values stay under 1%. We use K = 20 for the main AI-MO runs.



The gains carry over to logic puzzles

We train Qwen2.5-0.5B on Knights & Knaves puzzles with 3 to 7 people. Puzzles with 2 or 8 people are unseen.



Average accuracy (%)

— **14.3** Memory-R+

— **10.7** R1

— **7.3** Cosine

Memory-R+ is best at 6 of the 7 sizes. It also wins at both unseen sizes.



Memory-R+ trains stably

Baselines can collapse

R1 · FALCON3-1B

It chases format rewards, and correctness stalls. GSM8K drops from 32.9 to 16.3.

R1 · QWEN2.5-0.5B

Its answers shrink to bare digits. Correctness does not improve.

COSINE · QWEN2.5-0.5B

Its answers grow to the 200-token limit early. Rewards stop improving.

Memory-R+ never collapsed in any of our runs.

Its answers are also richer

We read 100 geometry answers from Qwen2.5-0.5B. We count the answers that check their own work.

26

6

Memory-R+

Base model

Diversity (×100)	Lexical	Semantic
Base	27.3	9.0
R1	24.2	8.6
Memory-R+	27.6	9.3

R1 loses diversity. Memory-R+ keeps it.



TAKEAWAYS

Small models reason better when they remember

- 01 Memory gives a dense reward. Every answer gets a score, even when none is correct.
- 02 Memory-R and Memory-R+ beat R1 on every model. Memory-R+ gains up to 14.4 GSM8K points over the base model.
- 03 The gains hold on harder maths and on logic puzzles. Training stays stable.

Limits

We test only maths and logic puzzles. We test models up to 1B parameters. AIME24 stays near zero for every method.



Thank you

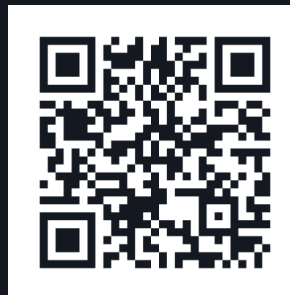
Questions are welcome. The paper and the code are open.

thai.le@deakin.edu.au



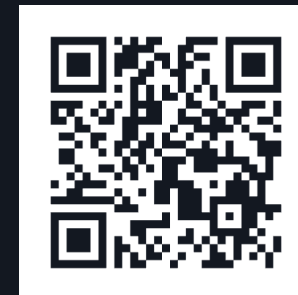
DEAKIN
UNIVERSITY

DEAKIN
**APPLIED ARTIFICIAL
INTELLIGENCE INITIATIVE**



Paper

OpenReview · TMLR 2025



Code

github.com/thaihungle/Memory-R