@ NeurIPS 2026

Towards Superhuman Intelligence: Self-Evolvement through Online Verified Data Synthesis

T³RL: Tool Verification for Test-Time Reinforcement Learning

Ruotong Liao*, Nikolai Röhrich*, Xiaohan Wang, Yuhui Zhang, Yasaman Samadzadeh, Volker Tresp, Serena Yeung-Levy
https://arxiv.org/html/2603.02203v1

The era of experience

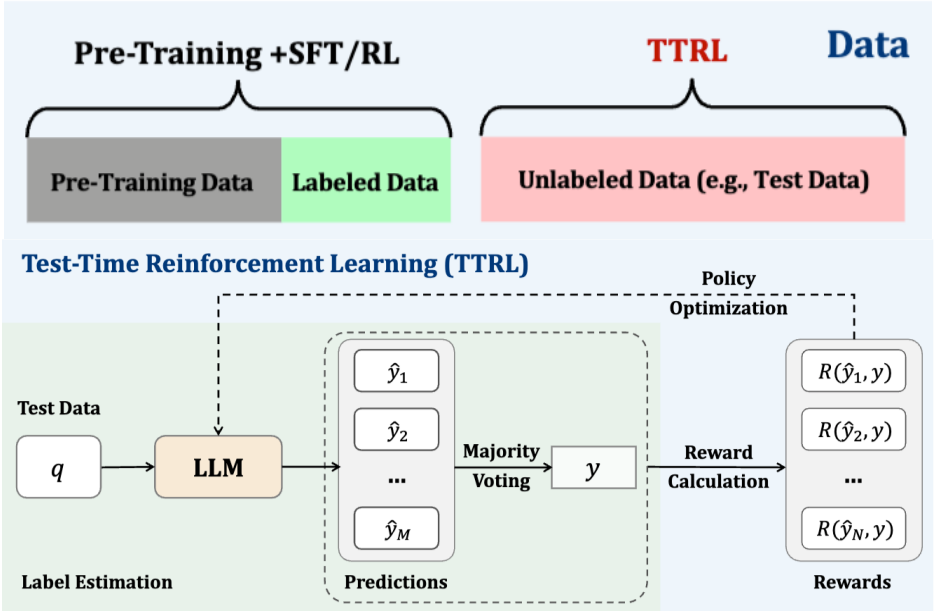
Human data is approaching its limit.

Silver & Sutton,

The next era of agents will learn continually from "streams of experience" - in mathematics, coding, and science, where high-quality human data is nearly consumed.

Can Language Model train itself?

Test Time Scaling: Large Language Models **know more** than they generate.



Paper source:
Silver & Sutton, Welcome to the Era of Experience, 2025.
Yuxin Zuo, Test Time Reinforcement Learning, 2025

The Accumulated Errors in Test Time Scaling

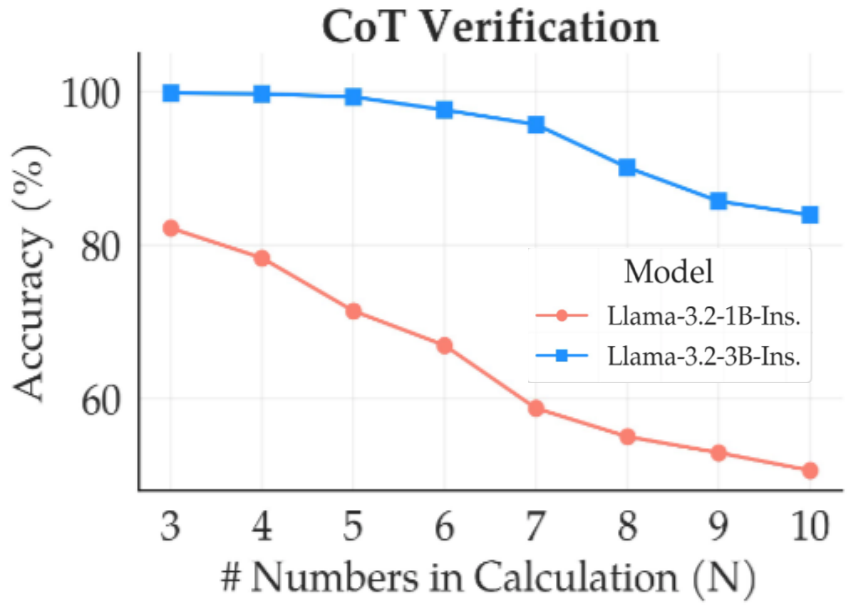
Compute $99^2 + 99 + 1$ in your head.

LLM

[...] We can rewrite $99^2 + 99 + 1$ as $99^2 + 99 + 1 = (99 + 1)^2 = 100^2 = 10000$. So, the final answer is 10000.

[...] $99^2 = (100 - 1)^2 = 100^2 - 2 * 100 * 1 + 1^2 = 10000 - 200 + 1 = 9801$. Now, we add 99 to 9801 to get 9900. Finally, we add 1 to get 9901.

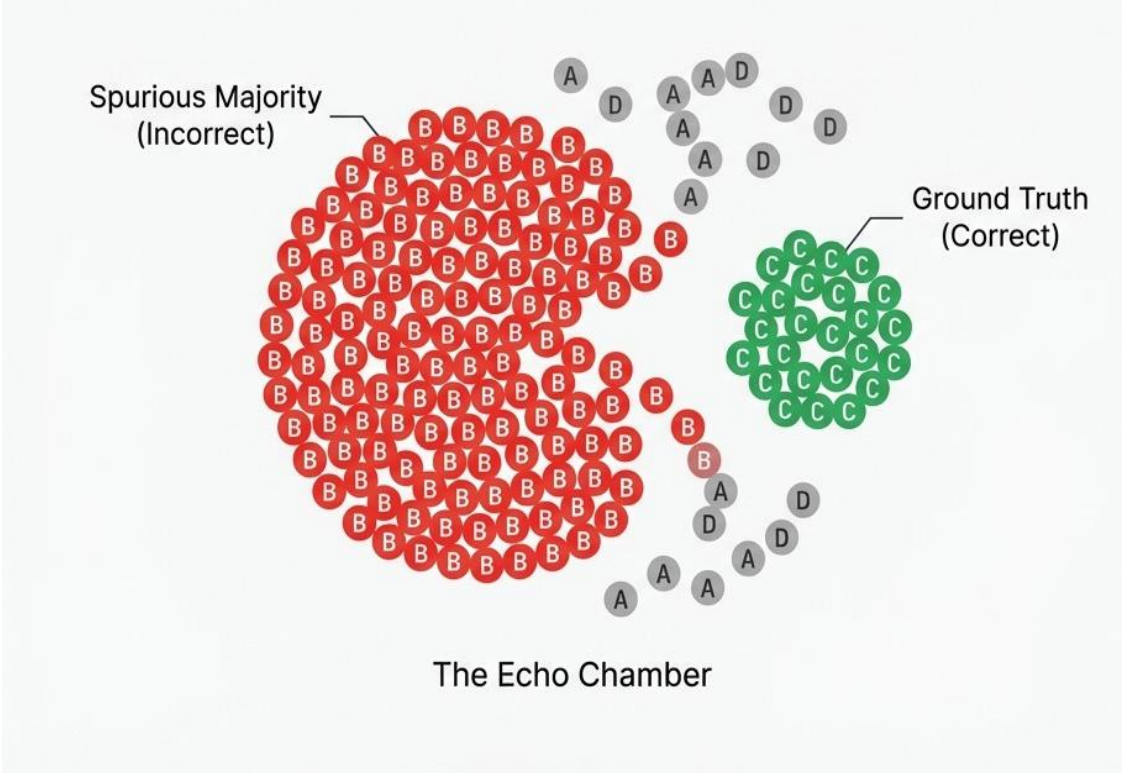
Let's do this step-by-step to ensure accuracy. [...] The final answer to the expression 99^2 + 99 + 1 is 10000.



Paper source: Minki Kang, T1: TOOL-INTEGRATED VERIFICATION FORTEST-TIME COMPUTE SCALING INSMALL LANGUAGE MODELS, 2026

The Echo Chamber creates a Reinforced Error

Popularity is not truth

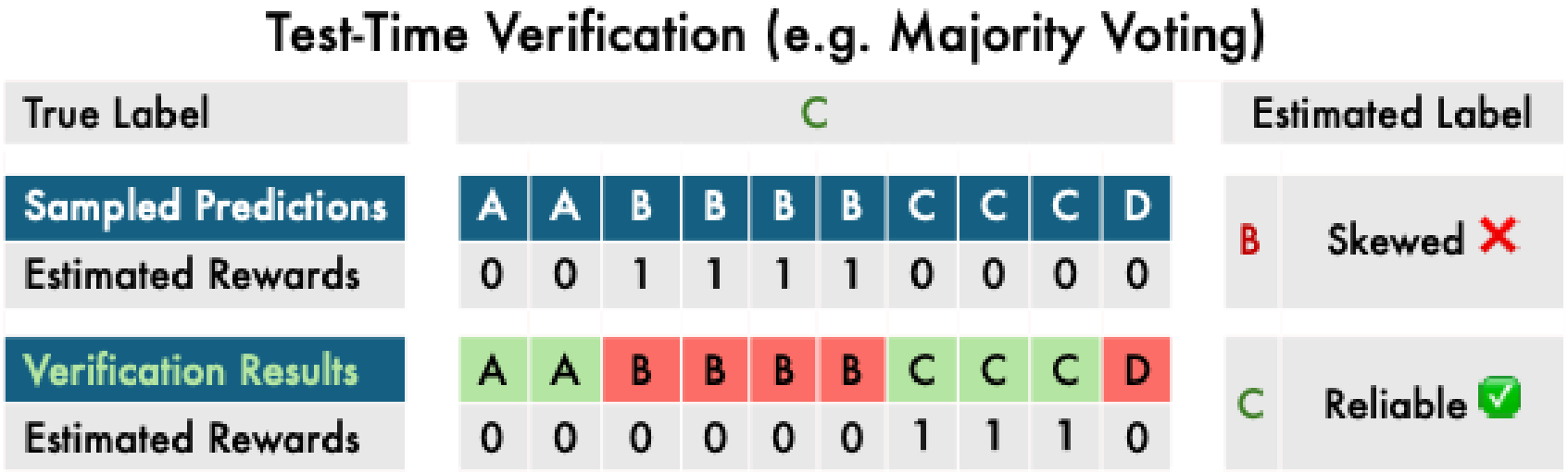


Vicious Cycle

True Label	C									Estimated Label
1st Round	False-Popular Predictions									
Sampled Predictions	A	A	B	B	B	B	C	C	C	D	B is good!
Estimated Rewards	0	0	1	1	1	1	0	0	0	0	
⋮											
Nth Round	Cycle of Reinforced Bias									
Sampled Predictions	A	B	B	B	B	B	B	B	C	D	B is really good!
Estimated Rewards	0	1	1	1	1	1	1	1	0	0	

Ruotong Liao, et al. Tool Verification for Test Time Reinforcement Learning. arxiv.org/html/2603.02203v1

The next scaling axis: Verification



The era of experience needs grounded rewards

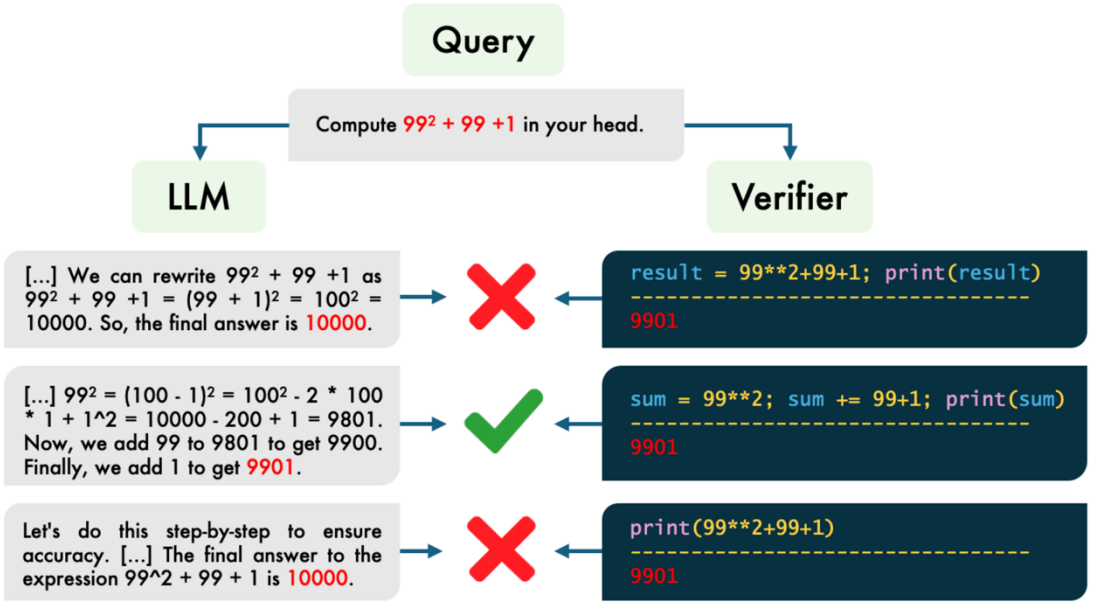
Human data is approaching its limit.

Rewards must come from the environment.

*"Rewards will be grounded in **their experience of the environment,** rather than coming from human prejudgement."*

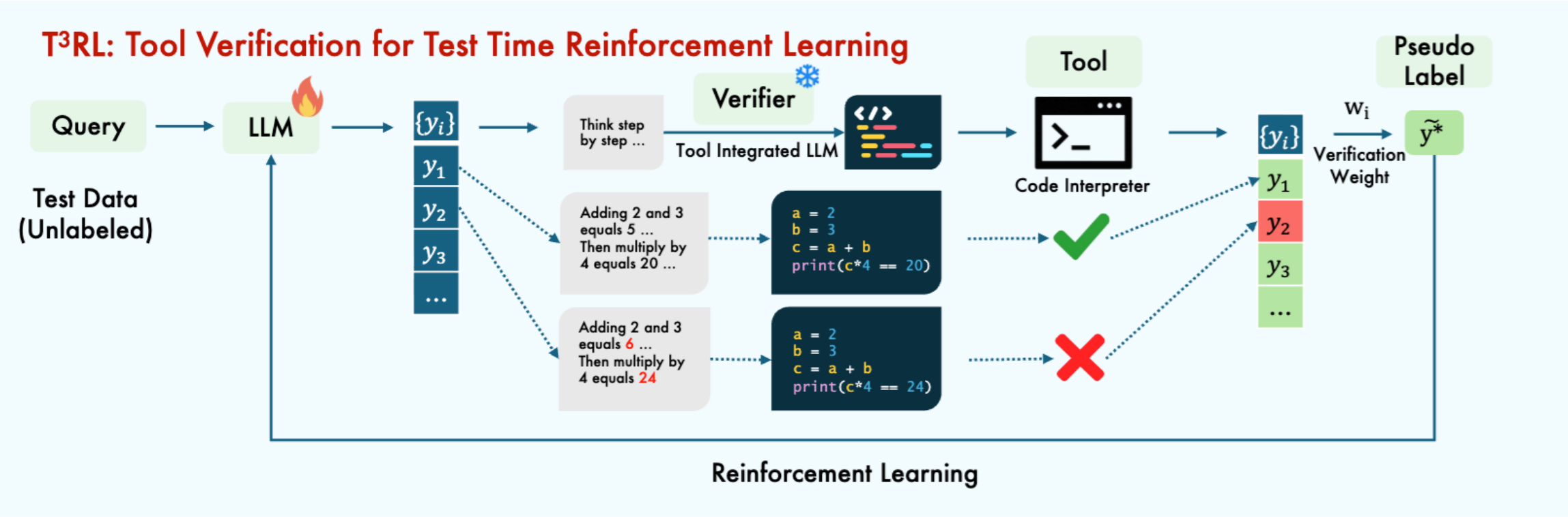
Test-Time Verification (e.g. Majority Voting)

True Label	C										Estimated Label	
Sampled Predictions	A	A	B	B	B	B	C	C	C	D	B	Skewed ✗
Estimated Rewards	0	0	1	1	1	1	0	0	0	0		
Verification Results	A	A	B	B	B	B	C	C	C	D	C	Reliable ✓
Estimated Rewards	0	0	0	0	0	0	1	1	1	0		



Paper source:
Silver & Sutton, Welcome to the Era of Experience, 2025.
Yuxin Zuo, Test Time Reinforcement Learning, 2025

Ruotong Liao, et al. Tool Verification for Test Time Reinforcement Learning. arxiv.org/html/2603.02203v1



Verification-aware consensus

$$w_i = (1 - v_i) \cdot 1 + v_i \cdot \omega,$$

$$\tilde{y}^* = \arg\max_{a \in \mathcal{A}} \sum_{i=1}^{N} w_i \cdot \mathbb{1}[\hat{a}_i = a].$$

Verified reward and training objective

$$r_i^{\mathrm{v}} = \mathbb{1}[\hat{a}_i = \tilde{y}^*].$$

$$\max_{\theta} \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[r^{\mathrm{v}}(y, \tilde{y}^*)],$$

$$\theta \leftarrow \theta + \eta \nabla_\theta \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[r^{\mathrm{v}}(y, \tilde{y}^*)],$$

T³RL breaks the loop by verifying before it votes

1 Sample

Generate **N reasoning rollouts** on an unlabeled test input.

2 Translate

An external verifier converts checkable steps into executable python calculation.

3 Execute

A code interpreter returns deterministic evidence.

4 Aggregate

Verified rollouts receive more vote mass in the pseudo-label.

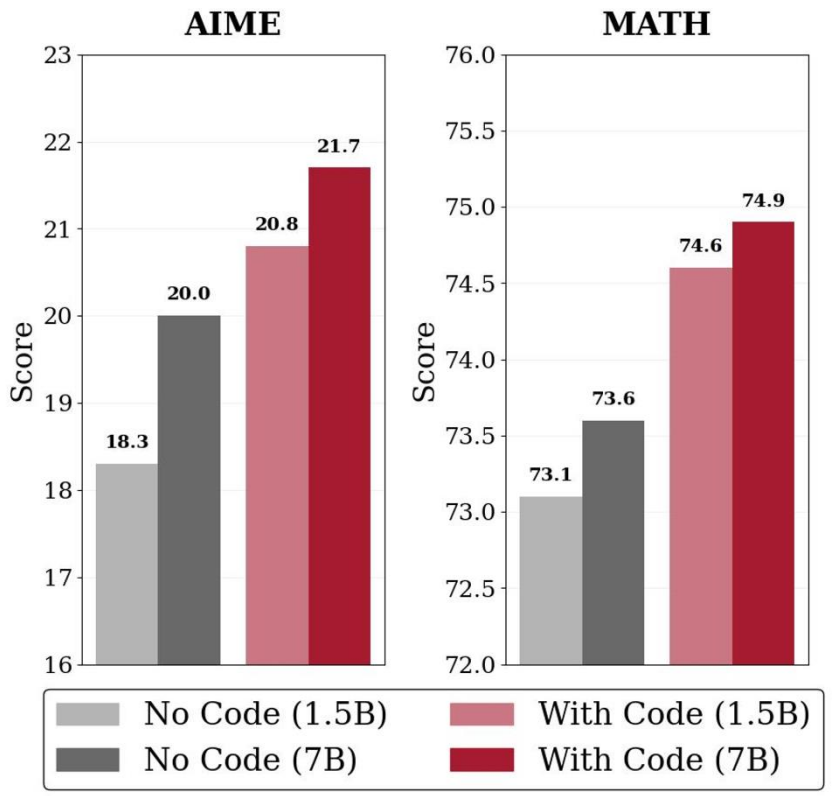
Unverified rollout = 1 vote

Verified rollout gets upweighted votes

a weighted pseudo-label

Ruotong Liao, et al. Tool Verification for Test Time Reinforcement Learning. arxiv.org/html/2603.02203v1

Tool evidence adds value beyond LLM self verification



	Math 1.5B	Llama 3B	Qwen3 4B
TTRL	15.8	13.3	36.4
Self	16.7	16.9	39.8
Ext.	**20.8**	**17.1**	**40.0**

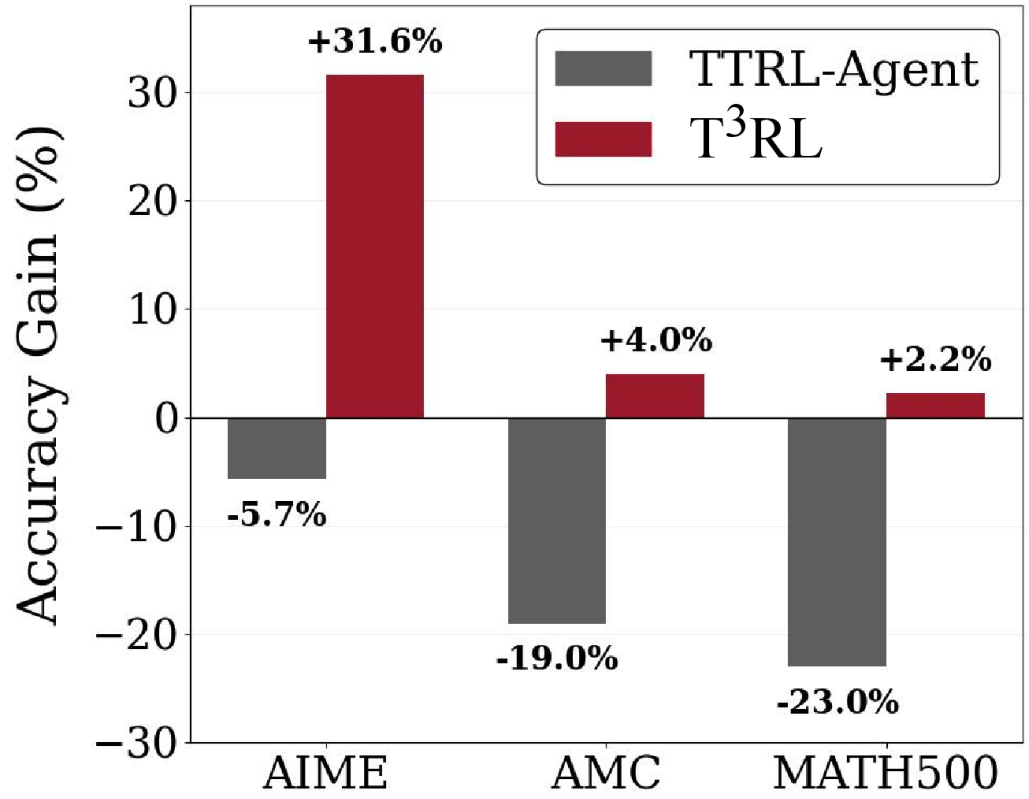
LLM-only helps

Self-verification improves the pseudo-label before any tool is executed.

Tool Verification helps more.

Ruotong Liao, et al. Tool Verification for Test Time Reinforcement Learning. arxiv.org/html/2603.02203v1

Is Tool Calling Enough? Position Matters.



Act with Tool inside the policy

Reasoning errors and tool-use errors are mixed, then amplified by self-consensus.

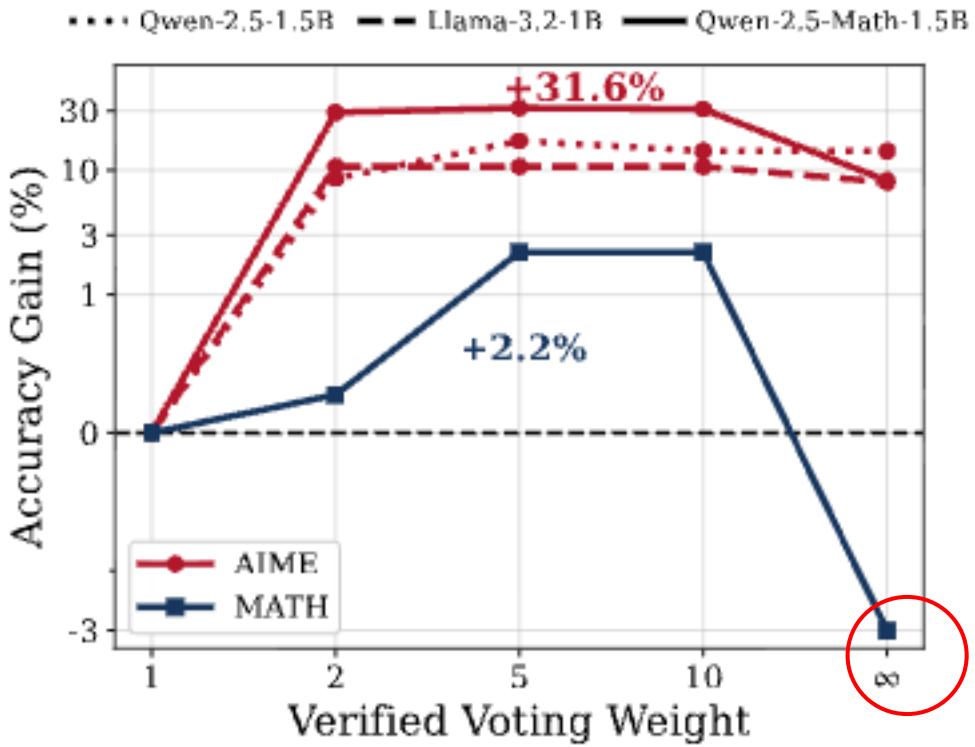
Verify with Tool outside the policy

Decouple Error Signal from **Tool Calling** and **Reasoning**
Decouple **Generation** from **Verification**

T³RL = verified online data synthesis

Ruotong Liao, et al. Tool Verification for Test Time Reinforcement Learning. arxiv.org/html/2603.02203v1

Verification must stay soft and the verifier should be competent



Hard filtering

Moderate weight wins

Hard filtering becomes brittle to verifier imperfections.

Verifier failure

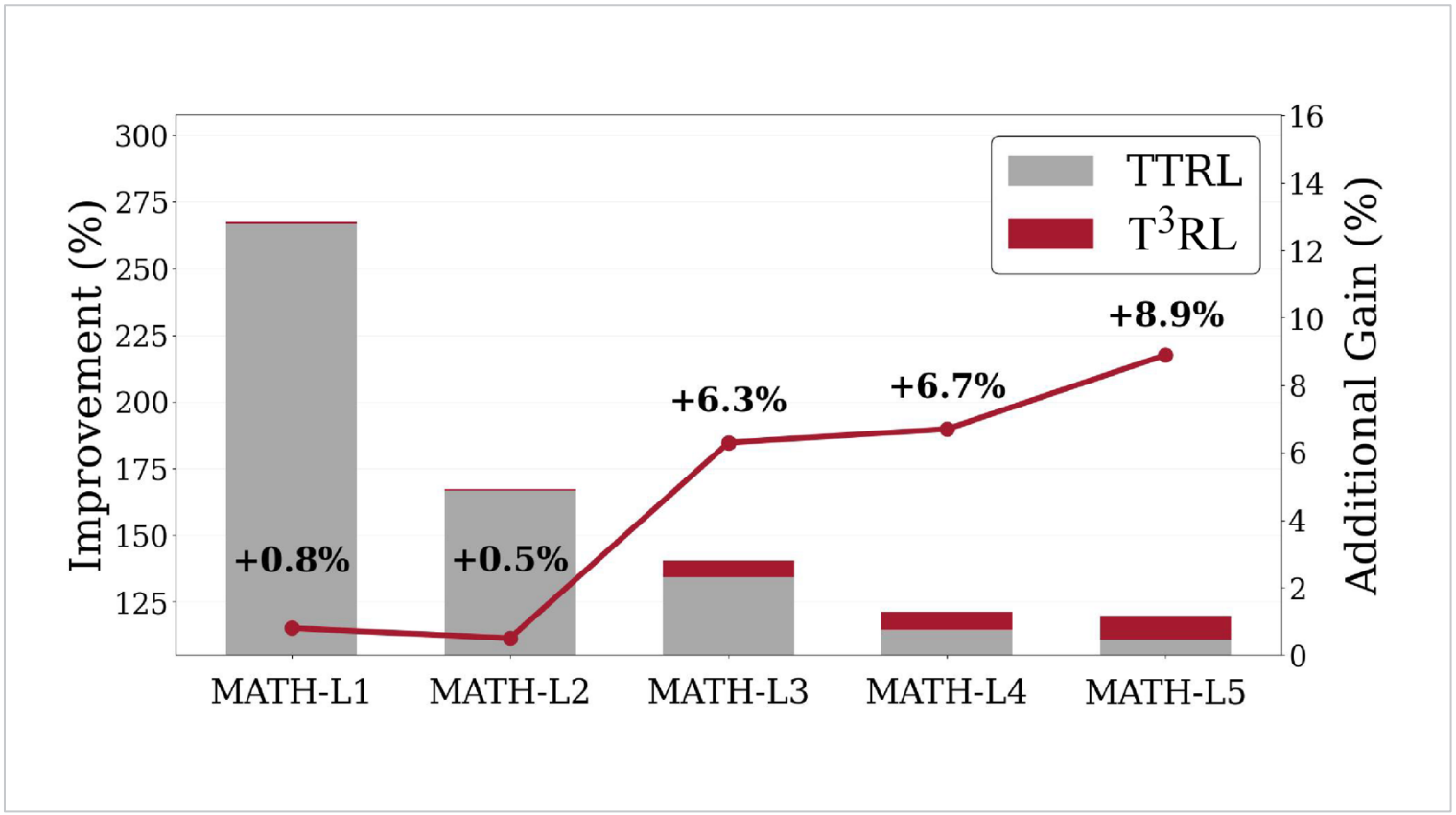
Policy	Verifier	Acc.	Prec.	Rec.	Δ AIME
Math-1.5B	Coder-1.5B	73.2	25.0	51.5	+5.0
Llama-3.2-3B	Coder-1.5B	73.3	39.4	59.1	+3.8
Qwen-3-4B	Coder-1.5B	75.0	38.2	40.7	+3.6

$$w_i = (1 - v_i) \cdot 1 + v_i \cdot \omega,$$

$$\tilde{y}^* = \arg\max_{a \in \mathcal{A}} \sum_{i=1}^{N} w_i \cdot \mathbb{1}[\hat{a}_i = a].$$

Ruotong Liao, et al. Tool Verification for Test Time Reinforcement Learning. arxiv.org/html/2603.02203v1

Verification matters most where reasoning is hardest



AIME 2024

15.8 -> 20.8

+31.6%

MATH-500 L5

largest extra gain within MATH

+8.9%

Longer reasoning chains accumulate more tool-detectable errors.

Ruotong Liao, et al. Tool Verification for Test Time Reinforcement Learning. arxiv.org/html/2603.02203v1

Test Time Compute, Wisely
Verification beats brute-force scaling for test time training

(a) AIME training runs; training costs include policy and verifier; arrows are vs. TTRL@64.

Method (rollout)	AIME ↑	Tokens ↓	Time (s) ↓	TFLOPs ↓
TTRL@64	15.8	365K	324	7.9K
T³RL@16	**20.0** (↑27%)	**147K** (↓60%)	**126** (↓61%)	**3.0K** (↓63%)
T³RL@32	20.8 (↑32%)	293K (↓20%)	218 (↓33%)	5.9K (↓25%)
T³RL@64	20.8 (↑32%)	586K (↑61%)	406 (↑26%)	11.8K (↑50%)

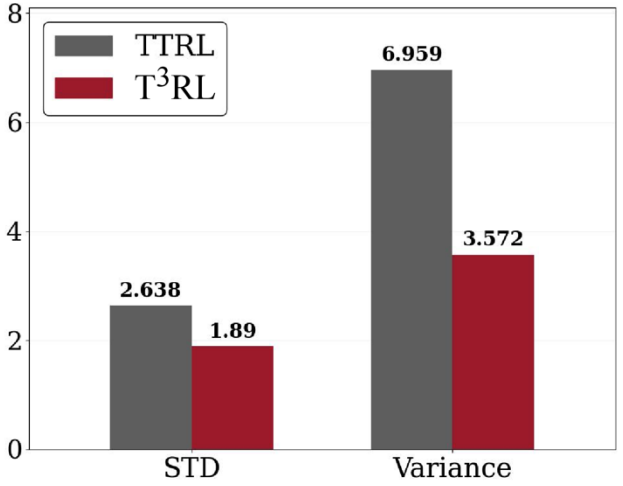
N=16 beats TTRL at N=64 with half training computation

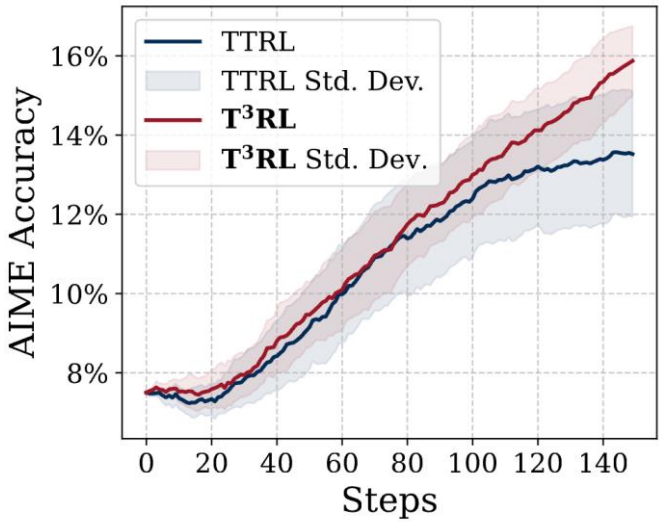
(b) Inference costs; arrows are vs. TTRL.

Method	Inf. Tokens ↓	AIME	AMC	MATH
Base	1,630	7.7 (↓63%)	28.6 (↓44%)	32.7 (↓56%)
TTRL	1,630	15.8 (↓24%)	48.9 (↓4%)	73.0 (↓2%)
T³RL (ours)	**1,630**	**20.8**	**50.9**	**74.6**
Test-time majority (N=16)	61K (↑37×)	19.2 (↓8%)	49.4 (↓3%)	65.0 (↓13%)
Tool-verified majority (N=16)	96K (↑59×)	20.8 (→0%)	51.5 (↑1%)	65.5 (↓12%)
T1 (N=16, V=4)	200K (↑123×)	20.0 (↓4%)	52.1 (↑2%)	66.3 (↓11%)
T1 + PRM (N=16, V=4)	200K (↑123×)	20.4 (↓2%)	52.7 (↑4%)	67.9 (↓9%)
Buffer-of-Thoughts	2,120 (↑30%)	9.6 (↓54%)	34.9 (↓31%)	38.6 (↓48%)
BoN-RR (N=64)	243K (↑148×)	21.0 (↑1%)	51.8 (↑2%)	51.7 (↓31%)

beats TTRL with 1% inference cost

Verification improves information quality per rollout.



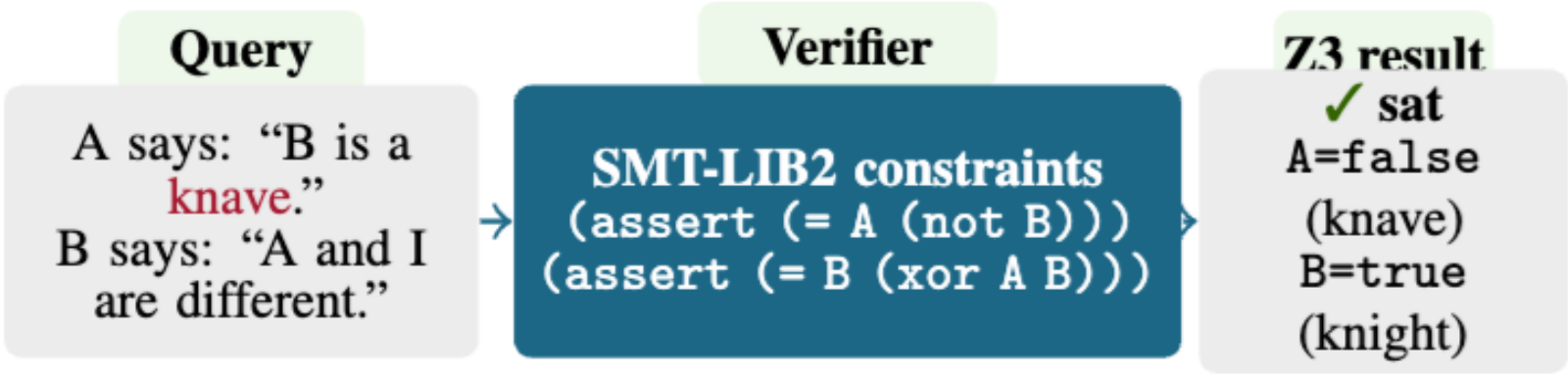


Stabilized Training

Tool-grounded rewards reduce run-to-run dispersion.

Paper source: Figures 8b and 9a; Section 6.1.

Generalization of Verification beyond Math



Puzzle	Base	TTRL	T^{3}RL	Δ vs. TTRL
n=2	34.5 ± 6.5	63.0 ± 0.0	**73.0 ± 1.2**	**+10.0**
n=3	22.2 ± 3.8	45.0 ± 2.0	**68.0 ± 0.8**	**+23.0**
n=4	11.8 ± 2.8	12.8 ± 0.5	**19.5 ± 0.5**	**+6.7**

N-character puzzles from Knights-and-Knaves solved by SMT-LIB2 and checked with Z3

TAKEAWAY

Self-improving AI with a verified self-distillation loop that generates synthetic training data online

Verified rollouts become training data itself after external evidence reshapes the reward.

CONSENSUS	TOOLS	RL
proposes candidates	**supply evidence**	**learns from verified modes**

Tool Verification makes your test time **training** **wiser.**

Ruotong Liao, et al. Tool Verification for Test Time Reinforcement Learning. arxiv.org/html/2603.02203v1