How Reinforcement Learning After Next-Token Prediction Facilitates Learning

A paper discussion: learning from rare long demonstrations

Paper by N. Tsilivis, E. Malach, K. Ullrich, and J. Kempe

arXiv:2510.11495v2 25 October 2026

Research group reading presentation

A model can sample a good long answer yet fail greedily

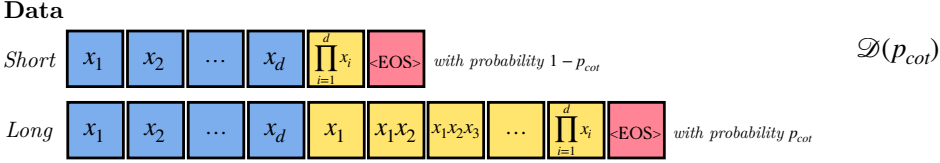
Research question

Why can reinforcement learning after next-token prediction rapidly improve accuracy and increase response length when correct long demonstrations are rare?

- Pre-training can learn the long path while greedy decoding chooses a short path.
- Rewarding correct generations enriches long answers in the next training round.
- The paper proves this for a linear autoregressive parity model and tests related behavior in transformers.

Tsilivis et al., arXiv:2510.11495v2, Sec. 1, PDF pp. 1–3.

The data contain two correct formats for one parity task



- Input: $x \in \{-1, +1\}^d$; answer: $s_d = \prod_{i=1}^{d} x_i$.
- Long fraction p_{cot}: output every partial parity $s_j = \prod_{i=1}^{j} x_i$.
- Short fraction $1 - p_{\text{cot}}$: output only s_d.

Paper, Fig. 1 (data panel), PDF p. 2; Sec. 2, PDF pp. 3–4. Figure cropped from the author source.

Long traces turn global parity into local updates

Short route

Predict $\prod_{i=1}^{d} x_i$ in one output step from all input bits.

Long route

Start with $s_1 = x_1$, then repeat $s_j = s_{j-1} x_j$ for $j = 2, \ldots, d$.

$$x = (+1, -1, +1, -1, +1) \quad \Rightarrow \quad (s_1, \ldots, s_5) = (+1, -1, -1, +1, +1)$$

- The parity task is representable by a shallow transformer; the key parity experiment probes learning from limited samples.

Paper, Secs. 2, 3.1, 5, PDF pp. 3–6 and 9–10. Example is pedagogical.

Training first fits demonstrations, then filters generations

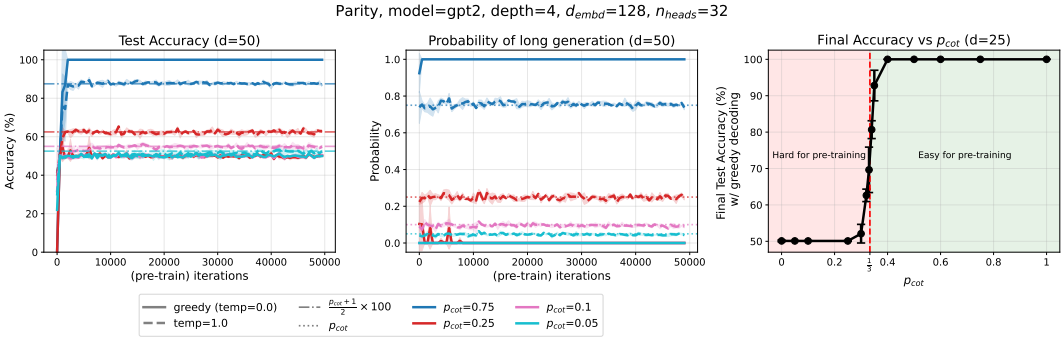
Next-token prediction on the mixed dataset

$$\mathcal{L}_{\mathrm{NTP}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}(p_{\mathrm{cot}})}\left[-\sum_{j=1}^{|y|}\log \pi_\theta{}_j(y_j \mid x, y_{<j})\right].$$

1. Sample outputs from the current model for new inputs.
2. Verify the final answer (r_{e2e}) or the entire trace (r_{cot}).
3. STaR refits accepted outputs; experiments also use REINFORCE and GRPO.

Paper, Sec. 2 and Appendix A, PDF pp. 4, 18–19.

Pre-training learns a calibrated mixture, not a greedy solver



- Dashed curves: temperature-1 sampling; solid curves: greedy decoding.
- At $p_{\text{cot}} = 0.25$, sampled accuracy is about 62.5% while greedy accuracy stays near 50%.

Paper, Fig. 2, PDF p. 5. Left/center: $d = 50$; right: $d = 25$; 3 seeds.

The $1/3$ threshold is a second-token decision

After the first output is x_1

$$P(\text{continue} \mid y_1 = x_1) = \frac{2p_{\text{cot}}}{1 + p_{\text{cot}}}, \qquad P(\langle \text{EOS} \rangle \mid y_1 = x_1) = \frac{1 - p_{\text{cot}}}{1 + p_{\text{cot}}}.$$

- If $p_{\text{cot}} < 1/3$, stopping has higher probability: greedy output is $(x_1, \langle \text{EOS} \rangle)$.
- x_1 equals the full parity on only half of random inputs.
- The derivation assumes the first-step predictor has not learned a full-parity feature.

Paper, Sec. 4, PDF p. 8; Appendix D.2, PDF p. 60.

Correctness filtering doubles the long-to-short odds

Before filtering

Correct long: p_n

Correct short: $(1 - p_n)/2$

After fitting accepted outputs

$$p_{n+1} = \frac{2p_n}{1 + p_n}, \qquad \frac{p_{n+1}}{1 - p_{n+1}} = 2\frac{p_n}{1 - p_n}.$$

- Idealized example: $p_0 = 0.25 \Rightarrow p_1 = 0.40 > 1/3$.
- More long responses become greedy choices, increasing test-time computation.

Paper, Sec. 3.2 and Sec. 4, PDF pp. 7–9. Example uses the noiseless recurrence.

Theorem 1 explains the pre-training state

Simplified model: per-position linear autoregressive heads

With ℓ_2-regularized logistic loss and polynomially many SGD iterations:

$$P(\text{correct long} \mid x) \approx p_{\text{cot}}, \qquad P(\text{correct short} \mid x) \approx \frac{1 - p_{\text{cot}}}{2}.$$

- For $p_{\text{cot}} < 1/3$, greedy decoding returns $(x_1, \langle \text{EOS} \rangle)$ and chance-level parity accuracy.
- Local updates are learnable via degree-two features; no direct full-parity feature is supplied.
- The guarantee needs sufficient accuracy away from the threshold.

Paper, Theorem 1, PDF p. 8; formal version in Appendix D.3, PDF pp. 73–76.

Theorem 2 reaches perfect greedy parity via STaR

Conditions and result

Start with $p_{\mathrm{cot}} < 1/3$, small calibration error, full-trace reward r_{cot}, and sufficient SGD work per round. Then

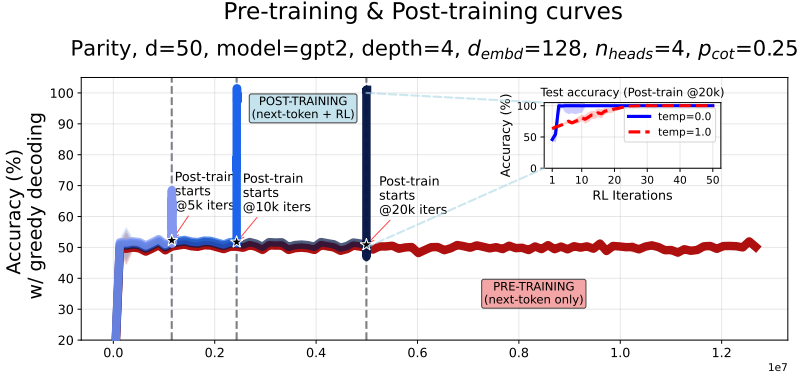
$$n^\star = O\left(\log \frac{1 - p_{\mathrm{cot}}}{p_{\mathrm{cot}}}\right)$$

STaR rounds suffice for the greedy model to output the entire correct trace.

- Actual effective weights q_n track the ideal recurrence: $|q_n - p_n| \lesssim 2^n \varepsilon$.
- If $p_{\mathrm{cot}} \in \Omega(d^{-\kappa})$, total SGD steps are polynomial in d.

Paper, Theorem 2, PDF p. 9; proof in Appendix D.4, PDF pp. 77–81.

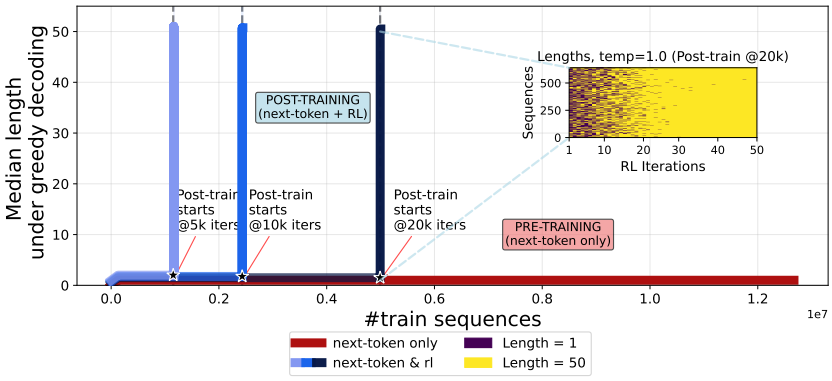
Parity RL rapidly unlocks greedy accuracy



- $d = 50, p_{\text{cot}} = 0.25$: NTP alone stays around chance even near 10^7 sequences.
- GRPO from sufficiently mature checkpoints quickly approaches full accuracy.

Paper, Fig. 1 right, PDF p. 2; upper panel cropped from author source; median of 3 seeds.

The parity accuracy jump accompanies longer responses



- Greedy NTP responses remain length 1; successful GRPO runs shift toward length $d = 50$.
- Starting RL too early can lengthen outputs without reliable generalization.

Paper, Fig. 1 right, PDF p. 2; lower panel cropped from author source.

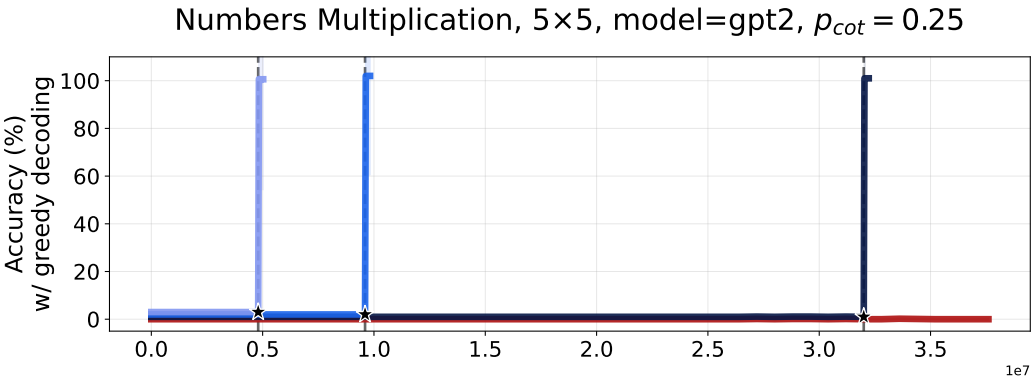
The effect depends on checkpoint and reward protocol

Setting	Observation
$p_{\text{cot}} = 0.25$, parity	STaR, REINFORCE, and GRPO usually improve; high-temperature STaR with final-answer reward is unstable.
$p_{\text{cot}} = 0.05$, parity	At 20k NTP iterations RL is unreliable; at 50k iterations success becomes more likely.
Partial traces	A later checkpoint can benefit from a length penalty; an earlier one may fail.

- Fig. 3 uses 3,200 generated samples per STaR round versus 64 per GRPO/REINFORCE iteration.

Paper, Sec. 3 and Fig. 3, PDF pp. 5–7; Appendix C.1–C.2, PDF pp. 23–30.

Five-digit multiplication shows a deeper-task analogue



- Scratch GPT-2; long examples contain grade-school multiplication traces.
- At $p_{\text{cot}} = 0.25$, greedy NTP stays near 0%; GRPO from useful checkpoints approaches 100%.

Paper, Sec. 5.1 and Fig. 5 left, PDF pp. 10–11; top panel cropped from author source.

The multiplication failure identifies a necessary seed

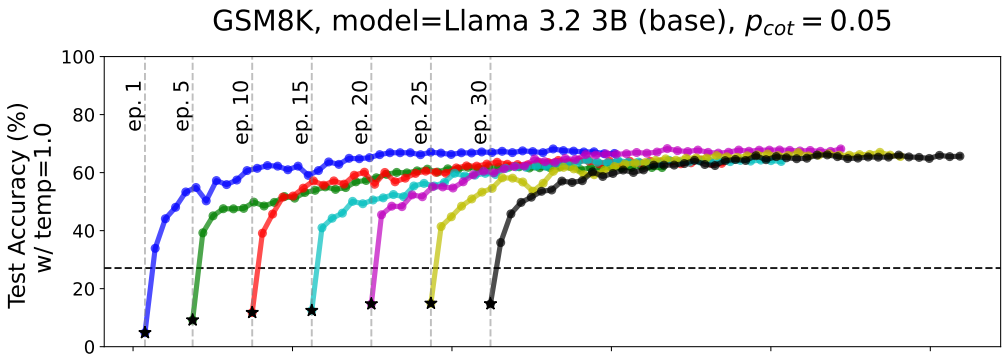
A limit of reward amplification

For 7×7 digit multiplication at $p_{\text{cot}} = 0.1$, none of the three seeds learned useful long solutions during pre-training, even after the reported 38 GPU hours; subsequent RL did not succeed.

- The model must generate some valid long outputs for correctness filtering to enrich them.
- The training set has 808k unique examples per task, reused over multiple epochs.
- Reward for multiplication checks complete agreement of the final number.

Paper, Sec. 5.1, PDF p. 10; Appendix B.2/C.5, PDF pp. 20–21 and 36–39.

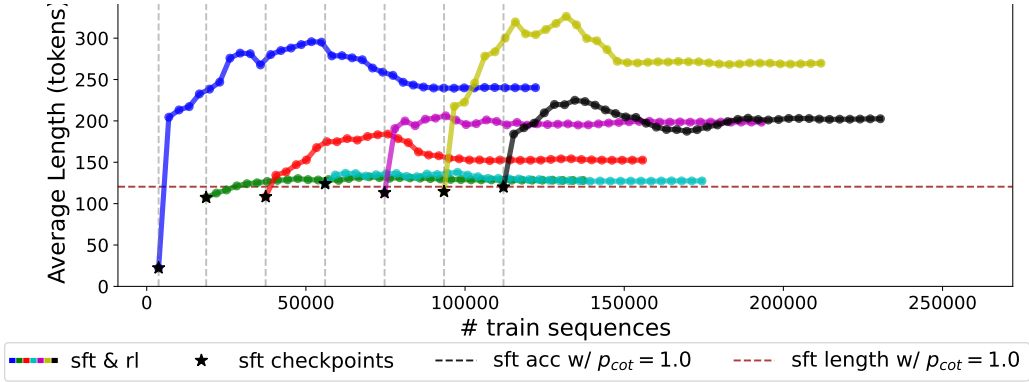
GSM8K: sparse long SFT traces precede rapid GRPO gains



- Llama 3.2 3B base, $p_{\text{cot}} = 0.05$: stars are SFT checkpoints; colored lines are GRPO continuations.
- Runs rapidly reach the dashed full-CoT SFT baseline, then continue above it.

Paper, Sec. 5.2 and Fig. 5 right, PDF p. 11; top panel cropped from author source.

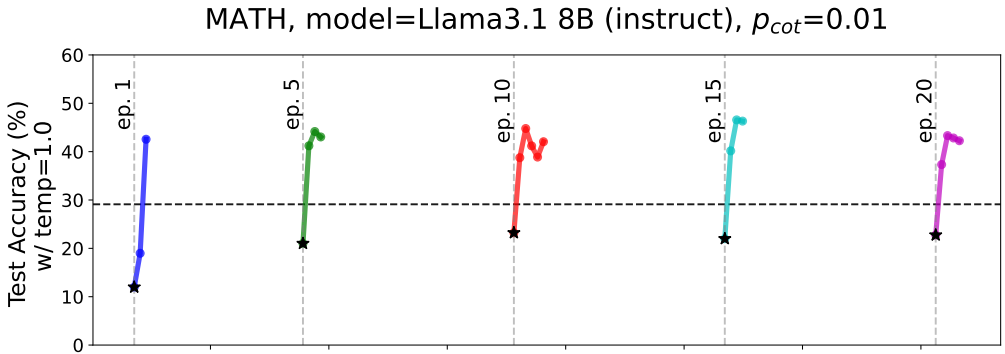
GSM8K length grows while answer quality improves



- Early completions are brief and often wrong; later ones resemble long SFT demonstrations.
- Later responses can depart from SFT style, suggesting an additional mechanism.

Paper, Fig. 5 right, PDF p. 11; Appendix C.6 examples, PDF pp. 43–46. Lower panel cropped from author source.

MATH improves too, with stronger attribution concerns



- Llama 3.1 8B instruct at $p_{\text{cot}} = 0.01$: GRPO raises test accuracy from SFT checkpoints.
- Some responses depart from SFT traces; the authors suspect prior exposure to MATH-like data.

Paper, Sec. 5.2, PDF pp. 11–12; Appendix Fig. 29, PDF p. 42. Panel cropped from author source.

Separate the proved mechanism from the empirical analogy

Evidence	Supported conclusion
Formal theory	A linear autoregressive parity learner with full-trace STaR reward crosses the greedy threshold in logarithmically many rounds.
Controlled simulations	Scratch transformers show length-calibrated sampling, poor greedy NTP output, and RL gains under tested configurations.
Llama benchmarks	Similar accuracy/length trajectories occur; late improvements may also use knowledge acquired before SFT.

- In Sec. 5.2 / Appendix B, "drop CoT with probability p_{cot}" conflicts with the paper's definition and $p_{\mathrm{cot}} = 1$ full-CoT plots.

Paper, Secs. 2–6 and Appendix B, PDF pp. 3–12, 21–22.

Takeaways for discussion

1. **Latent sampled skill:** rare long demonstrations can teach a useful path without making it the greedy choice.
2. **Reward-weighted data shift:** correct long outputs become more common in accepted batches, raising both response length and accuracy.
3. **Practical dependence:** a mature checkpoint, valid reward signal, and available long solutions matter; extra test-time tokens are part of the tradeoff.

Open question

How does this mechanism change with noisy, partial, or task-mismatched reasoning traces?

Paper, Sec. 6 and Appendix C, PDF pp. 12, 23–35.

Paper and evidence

- Tsilivis, Malach, Ullrich, Kempe. *How Reinforcement Learning After Next-Token Prediction Facilitates Learning*. arXiv:2510.11495v2.
- Full 81-page PDF, including appendices: arxiv.org/pdf/2510.11495v2.
- Key locators: parity Figs. 1–3 (PDF pp. 2, 5–6); Theorems 1–2 (pp. 8–9); multiplication/GSM8K Fig. 5 (p. 11); MATH appendix (pp. 41–42).

Questions?