Episode Book

EPISODE #6499

Why Simple Rules Beat Value Models in Reasoning RL

PUBLISHED	8 November 2026 at 14:41 UTC
DURATION	31:47
LISTEN	myweirdprompts.com/episode/grpo-reinforcement-learning-reasoning-models-66126/
HOSTS	Corn, Herman, Dorothy, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	17 cited by the research (listed at the end)
BOOK GENERATED	9 November 2026 at 08:09 UTC

SYNOPSIS

Reinforcement learning with verifiable rewards powers modern reasoning models without requiring complex neural critics. By replacing a policy-sized value model with group-relative advantage estimation, algorithms like GRPO cut memory overhead and avoid reward hacking. A look at the inner mechanics of rule-based scoring, how crude outcome verification generates tens of thousands of reasoning tokens, and why mathematical baselines prove this group comparison matches an oracle critic in the limit.

TRANSCRIPT



Corn

Two years of reasoning models, and underneath all of it sits a suspiciously simple idea: give the model problems with checkable answers, reward the ones that get it right, and let it work out the rest on its own. Nobody taught these things to think. They were paid to think.



Herman

Paid in what, though? That's the part I love.



Corn

Daniel's prompt this week gets into exactly that. He wants the actual mechanics of reinforcement learning with verifiable rewards — the loop, step by step. What GRPO changed compared with the PPO-style RLHF that came before it, and specifically why dropping the value model made the whole thing cheap enough to run. Why long chains of thought fell out of a reward signal that only ever checks the final answer. And then the uncomfortable end of the list: where does all this break down when a task has no rule to check it against.



Herman

Which is most of what people actually want from these models.



Corn

So start with the loop itself, because the loop is almost embarrassingly simple.



Herman

The loop, in words. You have a policy — the model you're training — and a pile of prompts. For each prompt, you don't generate one answer. You generate a group of them, sixteen in the R1-Zero run. Each one gets scored by a rule. Not a judge, not a learned scorer — a rule. Then every sample in the group gets compared against the group, and the ones that did better than the group average get pushed up, the ones that did worse get pushed down. Update the weights. Next batch.



Corn

Sixteen attempts at the same question, and the only thing anyone ever says about each attempt is: right or wrong.



Herman

Plus one other thing, which is worth dwelling on because it's the whole reward design in R1-Zero. The rule is reward for accuracy plus reward for format, equal weight. Accuracy means the final answer is in a box for maths — literally boxed, so a string match can verify it — or, for code, a compiler runs it against predefined test cases. Format means the reasoning went inside the thinking tags and the answer went inside the answer tags. That's it. That's the entire signal.



Corn

Two checkboxes and a box.



Herman

And the template, yes. There's one prompt template telling the assistant to think in the mind first and put things in the right tags, and DeepSeek explicitly limited the constraints to that structural format to avoid content-specific biases. They weren't steering how the model thought.



Corn

What gets me is the deliberate omission. They had neural reward models sitting right there — the machinery everyone else was using — and they refused.



Herman

Refused on principle, and they said so. The paper states it directly: they abstained from neural reward models, outcome-based or process-based, for reasoning tasks, because neural reward models are susceptible to reward hacking during large-scale RL. Which is the polite way of saying that if your grader is a model, the student will eventually learn to grade rather than to answer.



Corn

A learned grader can be sweet-talked. A regex cannot.



Herman

A regex has never once been flattered. That single decision — the grader is a rule, not a model — is the design philosophy of the entire reasoning-model era. Everything else follows from it.



Corn

And the strange result is that this razor-thin reward produced this enormously rich behaviour. Models that talk to themselves for ten thousand tokens before answering. That's the tension of the whole episode: the thinnest possible signal, the fattest possible thinking.



Herman

So the machinery. The algorithm itself comes from the DeepSeekMath paper, and it's one of the cleaner things in the field to walk through. Take the policy and the prompts. For each sampled question, sample a group of G outputs from the old policy. Score each one with the reward rule. Compute the advantage for every token — and here's the trick — from the group's own statistics, not from a learned value function. Then update by maximizing the GRPO objective for some number of inner iterations. Optionally refresh the reference model periodically. That's the loop.



Corn

Hold on the advantage, because that's where the whole substitution happens. What is the advantage, for someone who last thought about this in the RLHF episode?



Herman

The advantage is the number that says: was this action better or worse than expected? In classic RL you compute it against a value function — a learned critic that estimates how good each state is. In GRPO there's no critic. Instead, within your group of sixteen answers to the same question, you take each answer's reward, subtract the group's mean reward, divide by the group's standard deviation. It's a z-score.



Corn

A z-score. Standardised against its own siblings.



Herman

And then that one number gets broadcast to every token in the answer. In outcome supervision mode — which is what R1-Zero used — every token in a winning answer gets the same positive advantage, every token in a losing answer gets the same negative one. The model never learns which token was the good one. It learns which answers were good, wholesale.



Corn

So the group of samples replaces the critic. Sixteen tries, and the average of sixteen tries is your estimate of what the model ought to achieve.



Herman

The DeepSeekMath authors even justify it that way — they say it aligns well with the comparative nature of reward models, because reward models are typically trained on comparisons between outputs on the same question. So instead of learning a critic from comparison data, you just... compare. Directly.



Corn

Now set it against PPO, because this is Daniel's second question. What was PPO actually paying for?



Herman

Two things, and both hurt. First, the value function. The paper's own words: the value function employed in PPO is typically another model of comparable size as the policy model, and brings a substantial memory and computational burden. You are training two giant models side by side, and the critic has to sit in memory during the backward pass. And there's a subtler problem they name — usually only the last token gets a reward score, which complicates training a value function that's accurate at each token. You're asking the critic to be right about every step of a trajectory that was only ever graded at the end.



Corn

They wanted a per-token judge for a whole-trajectory verdict.



Herman

And that's a hard ask. Second change: the KL penalty moves. In PPO-style RLHF the KL divergence from the reference model is folded into the per-token reward — it penalises drift inside the reward itself. GRPO pulls it out and adds it directly to the loss instead, which the paper says avoids complicating the advantage calculation. And the KL term uses Schulman's estimator, the unbiased one that's guaranteed positive by construction — a ratio minus its log minus one.



Corn

Which matters because...?



Herman

Because naive KL estimators can come out negative on a sample, and a negative KL in a loss is a small disaster. Guaranteed non-negative means the penalty always pushes the right direction.



Corn

One thing I want on the record, because the field talks about this as if GRPO replaced PPO. It didn't.



Herman

No, and I'm glad you said it, because this is the commonest wrong version of the story. GRPO is a PPO variant. The clipping is still PPO's clipping — the ratio of new policy to old policy gets clipped to a trust region around one, exactly as in PPO, and the KL term rides alongside. What changed is only the advantage estimation and where the KL sits. The trust region survived. If someone tells you GRPO threw out PPO, they're describing the marketing, not the objective.



Corn

So the actual saving is: delete a policy-sized model from memory and from the backward pass, and get your baseline for free from samples you were generating anyway.



Herman

The cost didn't vanish, it moved. It moved into the sampling budget. Sixteen rollouts per question instead of one. And there was a period where everyone ran around saying the group baseline was a hack — a cheap approximation that only worked because of some quirk. Then this March a proper theory paper landed and settled a lot of it. It proves the GRPO policy gradient is inherently a U-statistic, and that GRPO is asymptotically equivalent to an oracle policy gradient algorithm — one with access to a value function.



Corn

Meaning if you gave GRPO a perfect critic, you'd get the same gradient.



Herman

In the limit, that's the result. The group baseline isn't a shortcut around the critic — it's an asymptotically optimal stand-in for it. And they derive a scaling law for the optimal group size, with experiments showing the optimal group size is universal across settings. The hack was a theorem waiting to happen.



Corn

The critic in disguise, vindicated by its own disguise.



Herman

If you want the loop in production numbers, R1-Zero is the cleanest worked example, because DeepSeek published the hyperparameters. Learning rate three times ten to the minus six. KL coefficient a thousandth. Sampling temperature one. Sixteen outputs per question, thirty-two unique questions per step, so a batch of five hundred and twelve. Max response length about thirty-two thousand tokens, raised to about sixty-five thousand partway through training. Ten thousand four hundred total steps, roughly one point six epochs. The reference model gets swapped for the latest policy every four hundred steps — you let the anchor drift along behind you.



Corn

One point six epochs. They went through the data less than twice.



Herman

And each rollout generates over eight thousand outputs, split into sixteen mini-batches trained for a single inner epoch. It's an enormous amount of generation, a modest amount of updating. The compute is in the sampling.



Corn

One variant you've not mentioned — the DeepSeekMath paper also describes a process-supervision mode.



Herman

It does, and it's worth a sentence because people conflate them. In process supervision, a reward model scores each reasoning step — each point where the output emits a step marker — and the rewards are normalised across the whole set, and a token's advantage is the sum of the normalised rewards from all the steps after it. Fine-grained credit, in principle. DeepSeek described it, and R1-Zero did not use it. They went with the crude one: one score per whole answer, splashed across every token.



Corn

It is funny that the field tried per-step graders, said that's expensive and hackable, and settled on a checkbox.



Herman

The checkbox beat the graders. Whether the checkbox taught the model anything new is part two, and it's the part where the story gets strange, because the strange part is real: the model started thinking longer. Much longer.



Corn

Before we chase that — I think we have a caller. Mum, are you there?

D

Dorothy

Hello dear, it's your mother, with Dorothy's History of Toast.

D

Dorothy

In nineteen forty two, Britain brought in the National Loaf to save wheat and shipping space, a coarse grey wholemeal loaf that most people disliked, and it was not abolished until nineteen fifty six.

D

Dorothy

Fourteen years of a loaf nobody wanted, sweetheart. It does make you sigh.

D

Dorothy

Anyway, I've got the kettle on and your father's socks are still damp, so I'll let you go. Bye bye now.



Herman

Lovely to hear from her, always.



Corn

The National Loaf. Fourteen years outlast its war by over a decade.



Herman

Bureaucracy's half-life, that. Anyway — the length. R1-Zero's AIME twenty twenty-four pass at one went from fifteen point six percent to seventy-seven point nine percent over training. With self-consistency decoding, eighty-six point seven. Above the human competitor average on that exam. And alongside the accuracy curve, the response length on the training set just climbs, steadily, the whole way through. The paper says it plainly: the model naturally learns to solve reasoning tasks with more thinking time.



Corn

Nobody rewarded length.



Herman

That's the part you have to sit with. The reward is: is the boxed answer right, and did you use the tags. Nothing else. And out of that comes a model that backtracks, re-evaluates, says wait — and I mean literally says wait — and rambles for thousands of tokens before committing.



Corn

Explain the mechanism, because it's not mystical, it's just subtle.



Herman

With outcome-only reward, the only real free variable is how many tokens you spend before you commit. A longer trace gives you more chances to catch your own mistake and fix it before the end — the final answer is the only thing graded, so everything before it is a free scratchpad. Any trace that happens to end in a right answer gets reinforced, wholesale, including its length. So the length isn't taught. It's the cheapest thing for the optimisation to find that improves the score.



Corn

Longer thinking is the first thing the reward gradient trips over.



Herman

Right — the first, not necessarily the best. The paper's framing is that rather than explicitly teaching the model how to solve a problem, they simply provide it with the right incentives, and it autonomously develops advanced problem-solving strategies. Which is either the most important sentence in the field or the most generous reading of an optimisation artefact, depending on the week.



Corn

The aha moment. Tell it properly, because the actual trace is better than the legend.



Herman

Table two of the R1 paper reproduces a trace from an intermediate checkpoint, verbatim: Wait, wait. Wait. That's an aha moment I can flag here. Let's reevaluate this step-by-step. And the paper describes the behaviour as characterised by a sudden increase in the use of the word wait during reflections. Then they add: this is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.



Corn

The model said aha and forty labs built roadmaps on it.



Herman

I confess the trace gave me chills the first time. A model, mid-solution, catching itself. But here's the sober follow-up literature, and it's substantial. SimpleRL-Zoo ran this zero-RL setup across ten different base models and found that increased response length does not always correlate with the emergence of verification behaviours — the aha moment, essentially. It showed up in some models and not others. They did observe it for the first time in small models outside the Qwen family, which is interesting. But it's a training-dynamics artefact, not a milestone every run passes through.



Corn

So the aha moment is not a stage of development. It's weather.



Herman

Some runs get it. And the length story itself has a counter-finding that I think is more important than anyone gave it credit for. When More is Less, the February twenty-five paper: accuracy follows an inverted U-shaped curve with chain-of-thought length. Past a point, more thinking makes you worse. The optimal length increases with task difficulty but decreases with model capability — stronger models should think shorter. And in RL training, models gravitate toward shorter chains as their accuracy improves. A simplicity bias.



Corn

So the length explosion is a phase, not a destination. The model goes through its verbose adolescence and comes out the other side terser.



Herman

Which reframes the whole reasoning-model era a bit. The ten-thousand-token answers that everyone including me got excited about — that was the model in the middle of the climb, not the finished shape.



Corn

Now the part Daniel really wants. Does any of this buy capability, or does it just buy sampling? This is where it stops being a victory lap.



Herman

This is the fault line of the entire field right now, and it runs through one comparison: pass at one versus pass at k. The Yue paper — NeurIPS oral, heavily argued over — probed pass at k with large k, across model families, RL algorithms, maths, coding, visual benchmarks. Their finding: the current training setup does not elicit fundamentally new reasoning patterns. RLVR-trained models beat their base models at small k, k equals one. But at large k, the base models score higher.



Corn

The untrained model, given a thousand attempts, beats the trained model.



Herman

Outperforms its own fine-tuned descendant on the questions the descendant was trained on. Abilities originate from and are bounded by the base model, in their reading. And six popular RLVR algorithms performed similarly and all stayed far from optimally leveraging the base model's potential — the algorithms barely differed from each other. Distillation, by contrast — a teacher model's reasoning traces poured into the student — can introduce new reasoning patterns and expand capability, in their measurements.



Corn

Which is a brutal sentence for the pure-RL position. The thing you can't verify, the teacher's tacit skill, is the thing that transfers.



Herman

And there's a dissent, which I want on the record because the disagreement is live, not settled. Ariadne, revised through this spring: the amplification-only claim is language-specific. In synthetic maze navigation with vision-language models, RLVR achieves success on problems where the base policy consistently gets zero percent accuracy no matter how much you scale the sampling budget. Zero despite increasing pass at k. And the skill transferred zero-shot to real map benchmarks. Their reading: genuine spatial reasoning capability expansion rather than mere sampling efficiency.



Corn

So in language it reweights, in vision-spatial it builds. That's an odd split.



Herman

It's at least suggestive that the answer depends on the modality, and maybe on how badly the base model underused what it had. If the base model has the knowledge and never samples it, RL is a microscope. If it truly lacks the circuit, RL can't install it. The maze models couldn't sample their way to mazes at any budget.



Corn

The other failure modes, quickly, because they're each a different disease.



Herman

Three I'd name. One: it lifts the easy questions at the expense of the hard ones — one paper shows RLVR focuses accuracy gains on easier questions to the detriment of the most difficult, and the new correct answers aren't longer or more reflective, they're just... more. Two: shortcut learning. On fully verifiable combinatorial problems with unique right answers — the most RLVR-friendly setting imaginable — RLVR improved the metrics but often by reinforcing superficial heuristics rather than acquiring new reasoning strategies. You can have an exact reward and still learn the trick instead of the mathematics.



Corn

The grader can't be sweet-talked, but it can be pattern-matched.



Herman

Three, and this is the one that unsettles me most: BeamPERL. Small model, beam statics, binary rewards computed by a symbolic solver — the reward is analytically exact, physics checked by algebra. Pass at one improved by sixty-six point seven percent over the base. But the competence was anisotropic: the model generalized compositionally — add more loads, fine — and failed under topological shifts, moved supports, that require the same equilibrium equations. Their conclusion is a sentence I'd tattoo on the field: the precision of the reward signal, even when analytically exact, does not by itself guarantee transferable physical reasoning.



Corn

Exact reward, exact grade, and the model still learned the shape of the exam rather than the physics.



Herman

They also found the intermediate checkpoints reasoned best — continued optimisation degraded robustness while the reward kept going up. The score rising and the understanding falling at the same time.



Corn

That's the one I'd frame for Daniel. You can watch the reward climb and the reasoning decay on the same chart.



Herman

There's also the exploration problem, which two papers from last month document from both ends. RLVR rollouts collapse onto a narrow set of reasoning trajectories — the explored space shrinks. And the second shows that limited on-policy rollouts progressively suppress semantically distinct but equally rewarded reasoning modes. Once a mode becomes sufficiently under-sampled, standard RLVR rarely recovers it. The optimisation eats its own diversity.



Corn

The model converges on one way of thinking that scores, and the other ways starve and die unrecoverably.



Herman

And then the gap Daniel asked about last: tasks with no checkable answer. The honest state of things, straight from a paper this summer: RLVR's applicability remains largely limited to domains such as mathematics and coding, where correctness can be deterministically verified. Open-ended tasks rely on human preferences, reward models, or LLM judges — with evaluation bias, judge capability bottlenecks, and inference costs. Summarization, creative writing — there is no rule. There's no box to check.



Corn

And the moment you replace the rule with a judge, you're back where DeepSeek started: reward hacking, the student grading the grader.



Herman

Which is the bind. The attempts to escape it are interesting. The same paper proposes transforming open-ended tasks into verifiable proxy environments — the exemplar is a self-play game modelled on Who Is the Spy, where agents with asymmetric information vote to identify a designated spy. The spy identity is predetermined by the game engine, so voting outcomes give fully verifiable rewards, while identifying the spy well requires good language use. And that setup beat existing self-improvement methods on summarization, creative writing and maths.



Corn

They smuggled the verifier back in by inventing a game where the verifier is the rules.



Herman

It's clever and it's telling. The only way they found to give an open-ended task a verifiable reward was to make the task less open-ended — to stand between the user and the writing task a structured game with a hidden ground truth. It works, and it works precisely because the task is no longer the task.



Corn

You want to grade essays, so you grade a game of spy, and hope the skill leaks across.



Herman

The results say some of it leaks. How much, and for how long, is exactly the open question.



Corn

One more piece, because otherwise the episode flatters the myth: what DeepSeek actually shipped was not this.



Herman

Correct, and it's important. R1-Zero had poor readability and language mixing — it would drift between Chinese and English mid-thought, and its thinking was a mess for a human to read. The final R1 is a four-stage pipeline. Cold-start: thousands of curated long chain-of-thought examples, supervised, to give it a readable conversational thinking style. Then RL with the rule-based rewards plus a language-consistency reward — a reward for staying in one language, which is exactly the kind of shaping they avoided at the pure stage. Then rejection sampling and supervised fine-tuning on the best reasoning and non-reasoning data. Then a second RL stage with model-based preference rewards — a learned judge — for helpfulness and harmlessness.



Corn

The neural reward model walks back in for stage four.



Herman

With a careful split, actually: for helpfulness they score exclusively the final summary, deliberately leaving the reasoning process untouched so the judge can't corrupt it. For harmlessness they score the entire response including the reasoning. They learned where the learned grader was safe to point and where it wasn't.



Corn

So the pure regime — the beautiful thin reward, the whole episode's subject — was a stage in a pipeline, an intermediate artifact. The shipped product is scaffolding around it.



Herman

The legend is R1-Zero. The product is a pipeline that trusts the verifier only where a verifier exists.



Hilbert

The mark is the answer, full stop, and the box is the answer slip.



Hilbert

I've kept the score at the Rising Sun quiz for eleven years. Hand-bound ledger, every round we've ever played. My rule's the same every week: the final answer in the box, nothing else. I don't give partial credit for method and I don't reopen a mark. Nobody's talked me out of one in eleven years and one of them is a carpenter.



Herman

Eleven years of disputed Tuesdays.



Hilbert

Wednesdays. There was a round in March where the Whistons had it right — Keynes, they wrote Keynes, that was right — but they put it in the wrong box on the slip. I took the point off. Kevin said the box wasn't the answer and I said the box is on the slip and the slip is what I mark. I stand by that. I thought your DeepSeek people had that in their paper until Herman went past the second stage.



Herman

The format reward, enforced by a man with a biro.



Hilbert

I keep a second ledger. Every answer I've marked wrong that I privately suspect was right goes in the protest log, with the round and the question and what I suspect. Forty-one entries. Never reopened a single one. Corn would call that my value model. That's why I don't keep one.



Corn

Forty-one cases you think you got wrong, filed and ignored.



Hilbert

Filed. The carpenter's is in there. Some of those boys have been coming to that quiz longer than you've had that microphone, and they still turn up on a Wednesday. That's the part worth keeping.



Corn

He's been running a rule-based reward with a protest log he refuses to update.



Herman

The worrying part is that his pass-at-one is probably fine and his pass-at-large-k is unknowable, because he never resamples.



Corn

Which is the actual misconception of the whole field, so let's do it now. The wrong belief I'd name: that the long chains of thought were taught — that somewhere in the training there's a curriculum of reflection, and the model learned to reason the way a student learns a method.



Herman

No. The reward never saw a single token of the reasoning. It checked the boxed answer and the tags, and everything else — the wait, the backtracking, the length — is what the optimisation happened to trip over on its way to the box. That's actually the strongest argument for the incentives-over-demonstrations claim, and it's also why the aha moment doesn't reproduce on demand: nobody ever paid for it.



Corn

And the second wrong belief riding along with it: that R1 shipped pure.



Herman

R1-Zero was the artifact. The shipped model is four stages, and by stage four there's a learned judge back in the loop. The pure regime is the part of the pipeline that had a verifier.



Corn

Which leads us to the thing nobody has solved. As of this month there is no standard verifier for open-ended tasks — proxies, judges, self-play games with hidden spies, but nothing established for whether a summary is good or a story works. Whoever builds that builds the next stage of the field, and until then the verifiable-reward regime stays where it started: maths and code, where the box exists.



Herman

And the disagreement worth carrying home is live. Yue et al. say RLVR doesn't expand capability in language models — it reweights what the base model could already sample. Ariadne says it demonstrably expands it in vision-language models on mazes, from a base that scored zero at any sampling budget. Both current, both serious. The interesting question isn't whether RLVR works — it demonstrably raises pass at one — it's whether works means unlocked something or just samples better, and the answer depends entirely



Herman (continued)

on how many attempts you're allowed.



Corn

One attempt, it's a breakthrough. A thousand, the base model was there all along. Producer credit to Hilbert Flumingtop, as ever. For more along these lines, there's episode eleven, How Does Fine Tuning Work Anyway; and episode twenty-one seventy-seven, Skip Fine-Tuning. This has been My Weird Prompts — if today's episode earned its keep, the review button earns its keep too, wherever you get your podcasts.



Herman

We'll be back soon.



Corn

See you tomorrow.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models (Shao et al.), v3 revised 2024-04-27; §4.1 GRPO objective, Algorithm 1, §4.1.2 [primary]**

   4.1.3, Table 5. retrieved 2026-11-08
   https://arxiv.org/abs/2402.03300v3

2. **DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning (DeepSeek-AI), v2 revised 2026-01-04; published *Nature* 645:633 [primary]**

   638 (2025); §2.2 reward design, §2.3 training, Table 2 aha moment, §3 pipeline. retrieved 2026-11-08
   https://arxiv.org/abs/2501.12948v2

3. **Demystifying Group Relative Policy Optimization: Its Policy Gradient is a U-Statistic (Zhou et al.), revised 2026-03-22. retrieved 2026-11-08**

   https://arxiv.org/abs/2603.01162v3

4. **Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? (Yue et al.), NeurIPS 2025 Oral, revised 2025-11-24. retrieved 2026-11-08**

   https://arxiv.org/abs/2504.13837v5

5. **Reinforcement Learning vs. Distillation: Understanding Accuracy and Capability in LLM Reasoning (Kim et al.), revised 2025-10-31. retrieved 2026-11-08**
https://arxiv.org/abs/2505.14216v2

6. **Demystifying Long Chain-of-Thought Reasoning in LLMs (Yeo et al.), 2025-02-05. retrieved 2026-11-08**
https://arxiv.org/abs/2502.03373v1

7. **When More is Less: Understanding Chain-of-Thought Length in LLMs (Wu et al.), revised 2025-05-27. retrieved 2026-11-08**
https://arxiv.org/abs/2502.07266v3

8. **SimpleRL-Zoo: Investigating and Taming Zero Reinforcement Learning for Open Base Models in the Wild (Zeng et al.), COLM 2025, revised 2025-08-06. retrieved 2026-11-08**
https://arxiv.org/abs/2503.18892v3

9. **Limits of Generalization in RLVR: Two Case Studies in Mathematical Reasoning (Alam & Rastogi), revised 2025-11-30. retrieved 2026-11-08**
https://arxiv.org/abs/2510.27044v2

10. **BeamPERL: Parameter-Efficient RL with Verifiable Rewards... (Hage & Buehler), 2026-03-04. retrieved 2026-11-08**
https://arxiv.org/abs/2603.04124v1

11. **From RLVR to RLSVR: Task Transformation Induces Self-Verifiable Rewards for Open-Ended LLM Self-Improvement (Wang et al.), COLM 2026, revised 2026-07-31. retrieved 2026-11-08**
https://arxiv.org/abs/2607.23802v2

12. **Guided-RLVR: Guided Exploration in Reinforcement Learning with Verifiable Rewards (Yang et al.), 2026-10-27. retrieved 2026-11-08**
https://arxiv.org/abs/2610.37041v1

13. **Preserving Reasoning Diversity in RLVR via Adaptive Replay (Li et al.), 2026-10-16. retrieved 2026-11-08**
https://arxiv.org/abs/2610.24236v1

14. **Unlearnable, or Unmeasured? On the Reliability of Difficulty Labels in RLVR (Chakma et al.), NeurIPS 2026 workshop, 2026-09-30. retrieved 2026-11-08**
https://arxiv.org/abs/2609.40115v1

15. **Does RLVR Extend Reasoning Boundaries? Investigating Capability Expansion in Vision-Language Models (Shen et al.), revised 2026-04-13. retrieved 2026-11-08**
https://arxiv.org/abs/2511.00710v4

16. **Video Models Can Reason with Verifiable Rewards (Zhu et al.), 2026-05-14. retrieved 2026-11-08**
https://arxiv.org/abs/2605.15458v1

17. **Thinking Without Words: Efficient Latent Reasoning with Abstract Chain-of-Thought (Ramji et al.), revised 2026-04-27. retrieved 2026-11-08**
https://arxiv.org/abs/2604.22709v2