

Learning reasoning strategies with RL

Policy optimization and group-relative advantages.

Reasoning agents, feedback, and scientific evidence

Useful experience need not change the next attempt

A model occasionally finds a correct reasoning trajectory, yet the next attempt repeats an avoidable mistake. Useful experience has not changed its future behavior.

Keeping a successful example could support imitation. A correctness checker could instead score new attempts, even when no complete demonstration is supplied.

This opens a choice about what experience should update: a stored example, a prompt, or the probability of future trajectories.

Constructing a signal for policy improvement

To change the policy, reward must distinguish behaviors worth making more likely. A final answer gives sparse feedback about the decisions that produced it.

Comparing raw rewards across easy and hard prompts can obscure that signal. Samples from one prompt provide a local reference, while a learned value model offers another baseline.

Group Relative Policy Optimization (GRPO) uses the sampled group to form relative advantages. This avoids a separate critic, but identical rewards provide no within-group contrast.

Pedagogical reconstruction

[Shao et al. \(2024\)](#)

[Zelikman et al. \(2022\)](#)

Choosing how experience changes behavior

Candidate	Reason to choose it	Cost or limitation
Imitate successful traces	Good demonstrations are available	Success filtering can retain flawed rationales
Learn a value baseline	State-dependent credit information is useful	Training and maintaining a critic costs resources
Compare a sampled group	Several outcomes share the same prompt	Sampling costs and reward ties limit the signal

GRPO becomes attractive when a reliable reward and contrasting samples are available. Its full objective still needs clipping and regularization choices.

Pedagogical reconstruction

[Shao et al. \(2024\)](#)

[Zelikman et al. \(2022\)](#)

Outcome rewards provide a sparse learning signal

A final correctness signal can be inexpensive when an executable checker exists. It does not directly identify which intermediate decisions caused success or failure.

Process feedback can provide additional information but may also reinforce a misleading decomposition.

Scientific outcomes are often delayed and noisy. A null finding may be useful when it discriminates mechanisms, while a positive result from a flawed procedure should not receive scientific credit.

Group-relative advantages compare sampled outcomes

$$A_i = \frac{r_i - \text{mean}(r)}{\text{std}(r) + \varepsilon}$$

DeepSeekMath's Group Relative Policy Optimization (GRPO) normalises each prompt's sampled outcome rewards by their group mean and standard deviation.

The resulting quantity is relative to the sampled group, not an absolute measure of reasoning quality. A group in which all rewards are identical provides little contrast.

Practical algorithms include clipping, token aggregation, sampling, and regularisation choices; the simple normalisation alone does not specify the training system.

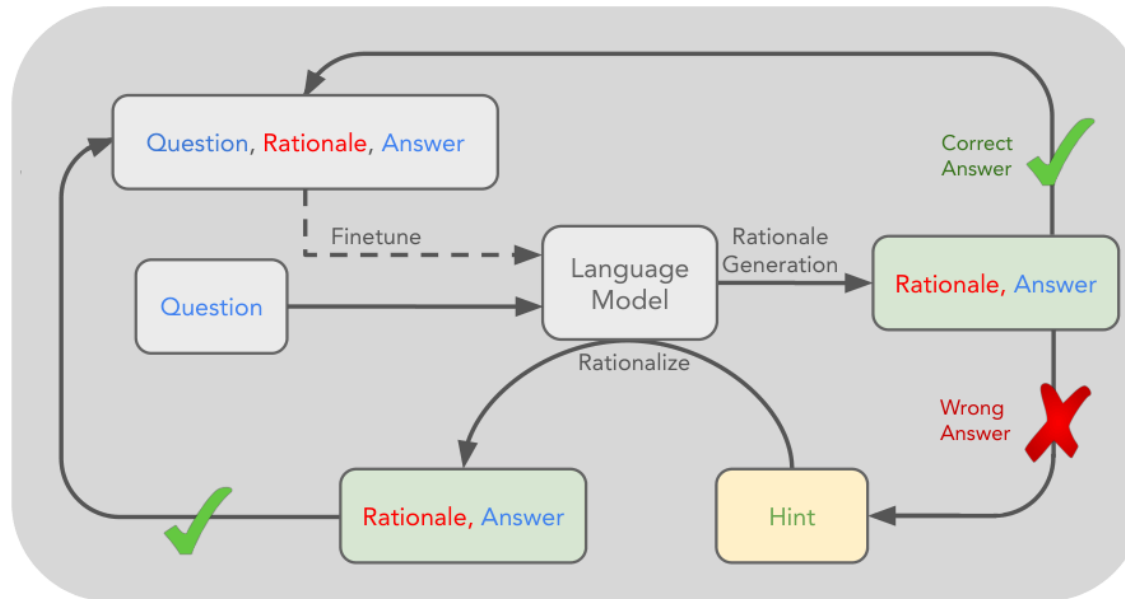
STaR: bootstrapping with supervised updates

STaR bootstraps reasoning by generating rationales, retaining successful solutions, and fine-tuning on them. When a model initially fails, an answer-conditioned rationalisation can supply an additional training example.

Iteration changes the distribution of available rationales.

Correct final answers provide a useful filter, but the retained rationale may still include spurious reasoning or depend on information unavailable at deployment.

STaR: rationale generation and filtering



Q: What can be used
to carry a small dog?

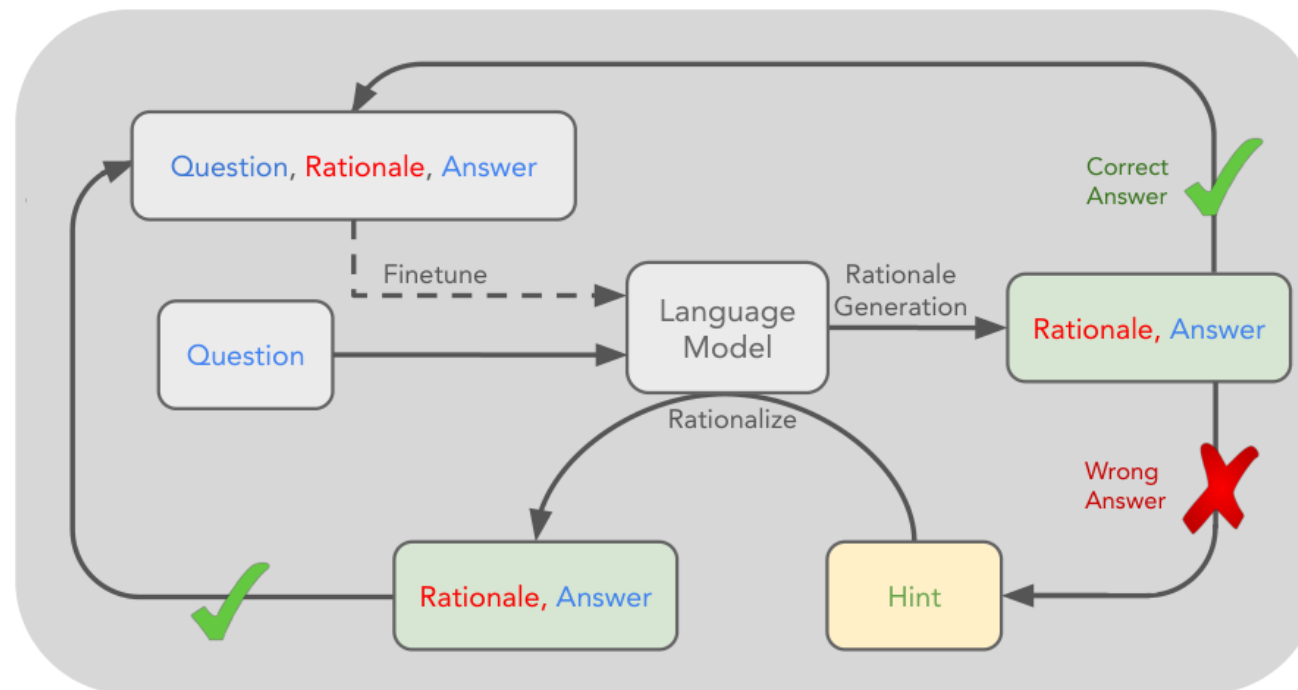
Answer Choices:

- (a) swimming pool
- (b) basket
- (c) dog show
- (d) backyard
- (e) own home

A: The answer must be
something that can be
used to carry a small
dog. Baskets are
designed to hold things.
Therefore, the answer
is basket (b).

Correct generated answers enter the training set directly, while incorrect attempts can enter an answer-conditioned rationalisation path. Information used to construct training data differs from information available at inference time.

STaR: mechanism detail



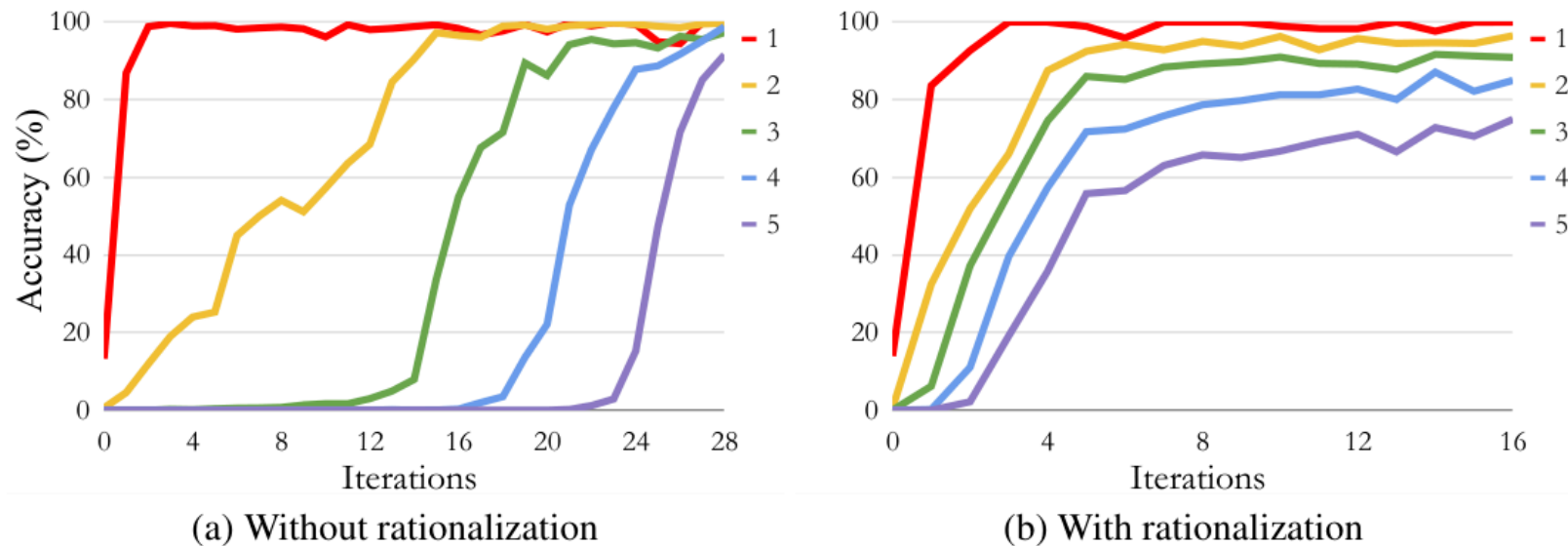
The rationalization branch supplies extra information after failure and applies a retention condition. Its interpretation depends on separating useful reasoning supervision from copying the supplied answer.

Evidence: training changes and rationalisation

	GSM8K Test Accuracy (%)	Train Data Used (%)
Few-shot Direct GPT-J	3.0	~0
Few-shot CoT GPT-J	3.1	~0
GPT-J Direct Finetuned	5.8	100
STaR without rationalization	10.1	25.0
STaR with rationalization	10.7	28.7

On GSM8K with GPT-J, direct fine-tuning reaches 5.8%, STaR without rationalisation 10.1%, and STaR with rationalisation 10.7%. The training-data shares also differ. The paper discusses iteration and compute limits. A comparison of final accuracy does not by itself isolate the causal contribution of rationalisation.

STaR: interpretation and limits



A correctness filter can retain lucky solutions and reinforce a narrow set of successful patterns. Rationalisation can introduce training–inference differences, so held-out evaluation and process analysis remain necessary.

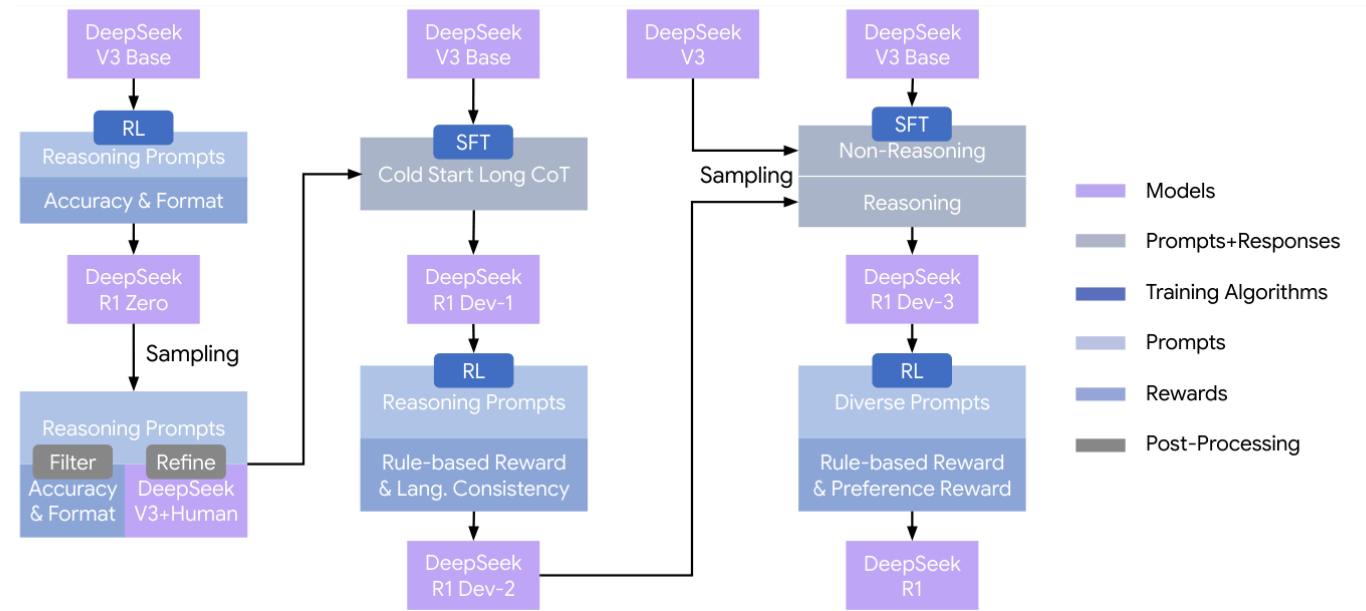
DeepSeek-R1: the method

The source distinguishes reinforcement-learning-driven reasoning development from the additional supervised and post-training stages used in the complete R1 system.

The pipeline combines several data and optimisation choices, so comparing checkpoints can illuminate the contribution of individual stages.

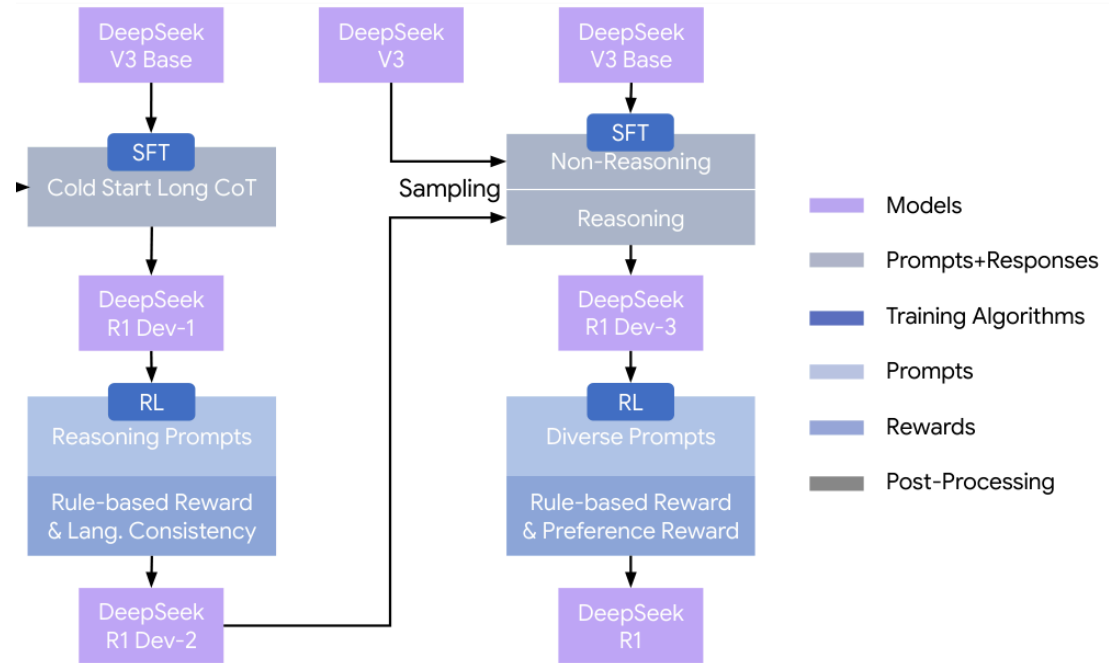
The preserved source version documents the updated pipeline and evaluation. Conclusions concern that setup, including its distinct training stages.

DeepSeek-R1: training stages



The diagram connects the base model, reinforcement learning, cold-start supervision, filtering and later training stages. Comparisons of individual stages support narrower attribution than comparisons between complete systems.

DeepSeek-R1: mechanism detail



Supervised intermediate checkpoints and later stages change cold-start data, example mixtures and reward criteria. R1-Zero and the multi-stage R1 pipeline therefore support different attribution claims.

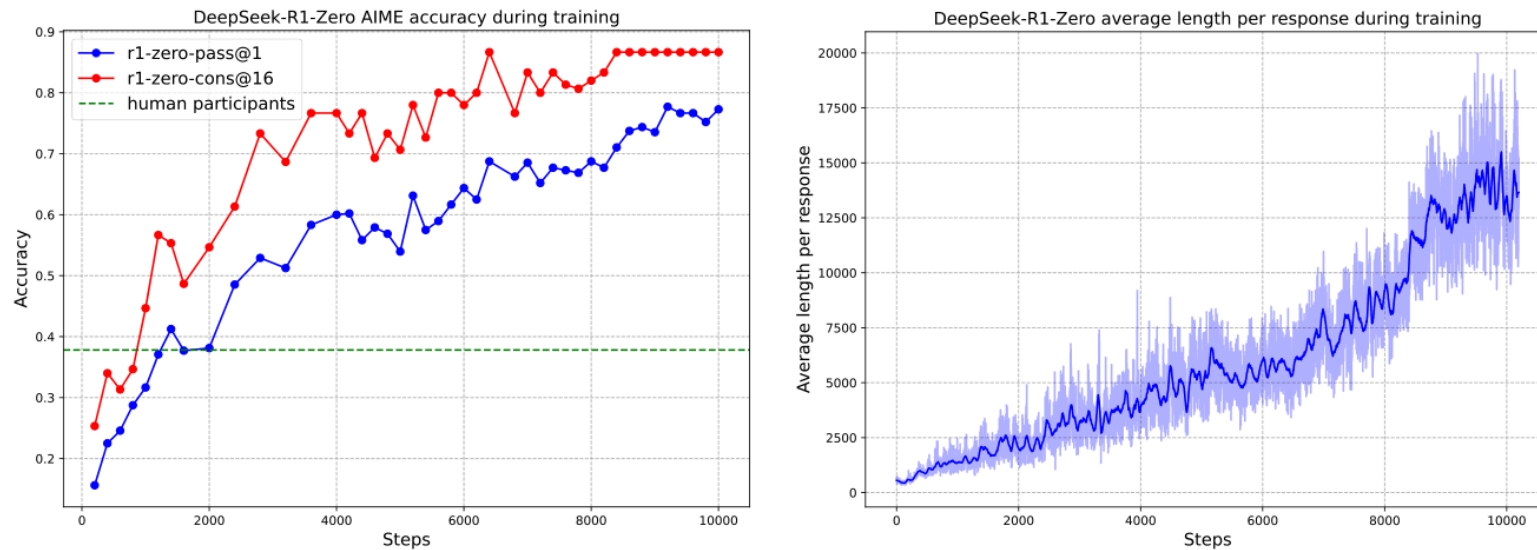
Evaluation scope

The original training figure reports mathematical evaluation performance alongside response length during reinforcement learning.

Increasing response length and increasing task performance are correlated observations in this setup; the plot alone does not prove that longer text causes better reasoning.

Sampling and aggregation conditions affect the reported results. R1-Zero follows a different training trajectory from the multi-stage R1 pipeline.

DeepSeek-R1: interpretation and limits



Training data, filtering, reward design, and evaluation procedures all affect the result. Selected mathematical benchmarks do not establish scientific discovery competence or universal benefits from longer reasoning traces.

Pass-at-one and coverage can move differently

Training can concentrate probability on known successful patterns and improve the chance that one sample is correct. It may also change the diversity of solutions found at larger sampling budgets.

Compare pass-at-k curves using consistent sampling and evaluation. Finite samples do not prove that an unobserved solution is impossible for a model.

Claims about new reasoning capacity require carefully defined tasks, budgets, and alternative explanations.

Reward optimisation can exploit an imperfect checker

As optimisation progresses, generated trajectories can move outside the distribution on which the reward was validated. The policy may exploit omissions or biases in the checker.

Test adversarial and degenerate outputs, keep a held-out evaluation, and inspect the relationship between reward and the target capability.

A separate verifier model is helpful only to the extent that its errors differ meaningfully and its evaluations remain valid under optimisation.

Credit for scientific learning needs a target belief

If the reward values information gain, define the target uncertainty, prior, likelihood, and observation process. Otherwise the policy can earn novelty or surprise scores without producing useful evidence.

Reward measurement validity separately from the sign of the result. Preserve negative observations and inconclusive outcomes.

A simulator with known mechanisms makes it possible to compare reward quality with actual identification performance.

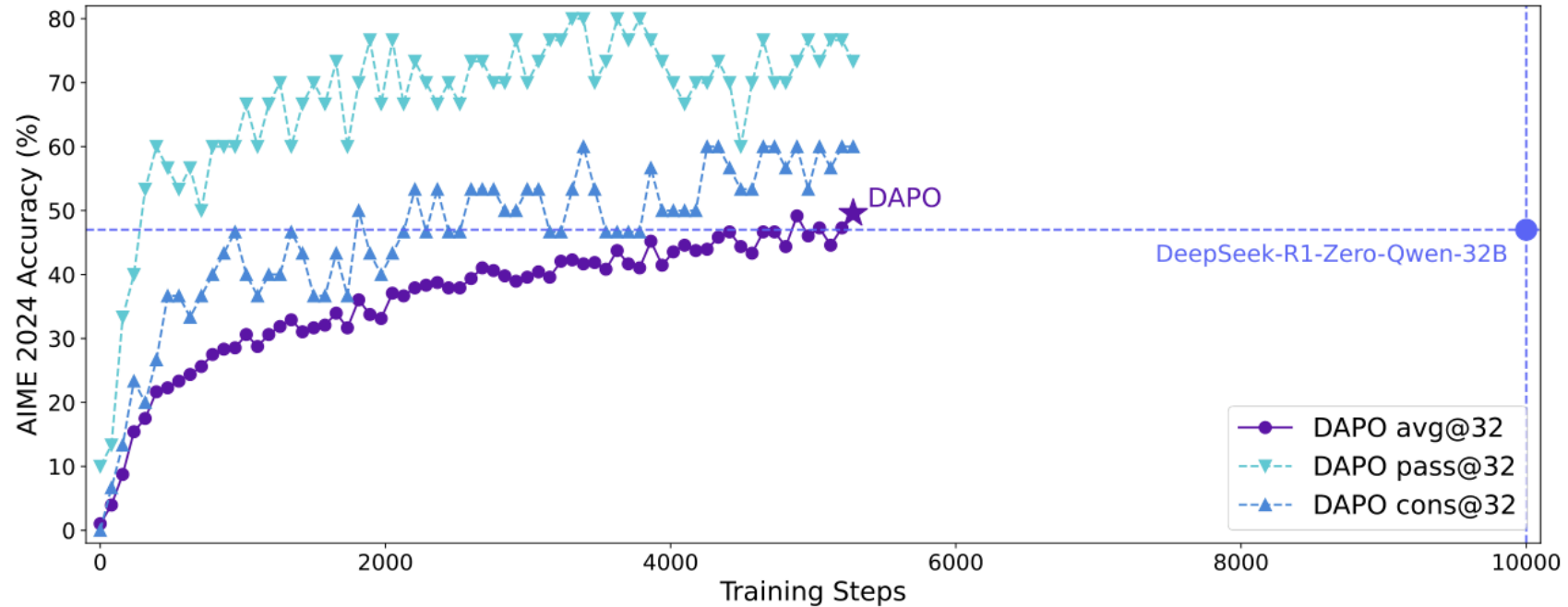
DAPO: the method

DAPO—Decoupled Clip and Dynamic sAmpling Policy Optimization—is a reinforcement-learning recipe combining asymmetric clipping, dynamic sampling, token-level loss aggregation and reward shaping.

These decisions affect exploration, learning signals, and optimisation stability.

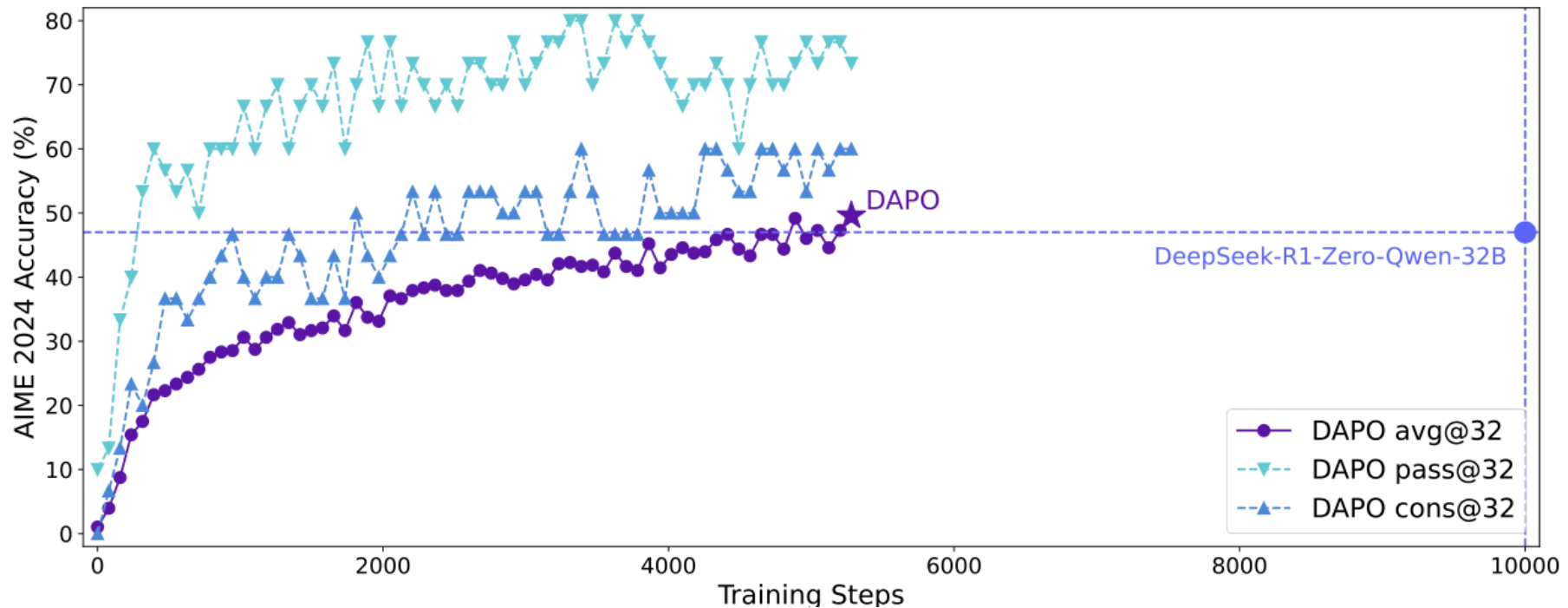
The contribution should be analysed as a documented training system, with ablations distinguishing individual changes from the overall result.

DAPO: accuracy under different selection rules



The curves distinguish average accuracy, pass-based aggregation and consensus under the labeled conditions. Their interpretation depends on the base model, task, sampling and training setup.

DAPO: aggregation and training curves



Average, pass-based and consensus curves use different answer-selection rules over the same training steps. Their separation partly reflects aggregation; gains under one rule do not establish the same gain in single-sample accuracy.

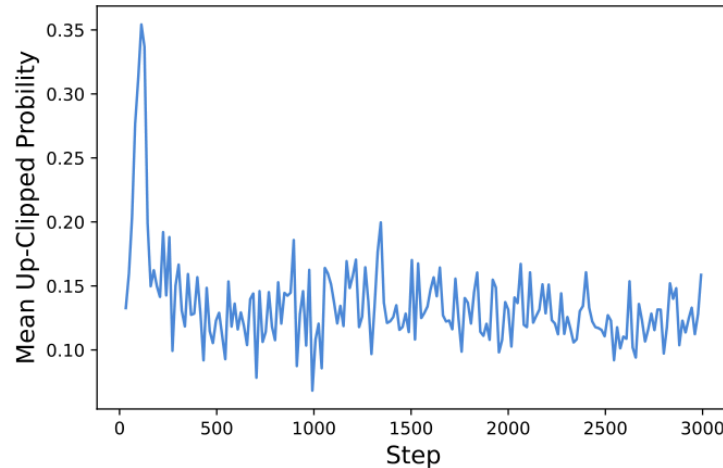
Evaluation scope

The source examines training dynamics and the prevalence of prompts with uniform rewards, alongside the main reasoning evaluation.

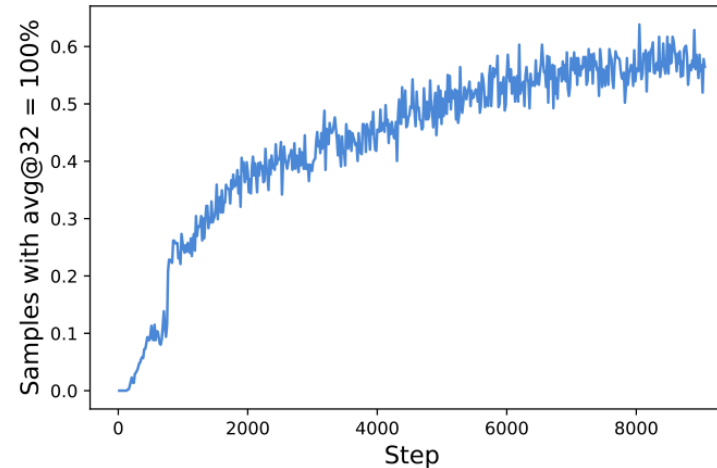
The extracted diagnostic plots show how optimisation conditions evolve during training. They are diagnostics of the procedure, not independent measures of scientific validity.

Ablations and controlled comparisons connect each engineering change to the failure it is intended to address.

DAPO: interpretation and limits



(a) Mean up-clipped probability.



(b) The proportion of samples with an accuracy of 1.

Filtering and sampling can change the effective training distribution and computational cost. Gains on the evaluated reasoning benchmark do not establish a universally best reward or training recipe.

DR Tulu: the reward criteria also evolve

Training stage	Information used	Object updated
Sample research trajectories	Search results and contrasting answers	Evidence for useful feedback criteria
Maintain rubrics	Persistent criteria plus an evolving buffer	Which factual distinctions are rewarded
Score the current group	Rubrics and auxiliary reward terms	Rewards and relative advantages
Update the policy	Group Relative Policy Optimization (GRPO)	The model's trajectory distribution

Reinforcement Learning with Evolving Rubrics (RLER): first released in 2025; inspected May 2026 revision. Rubrics here are model rewards.

What does experience update?

Mechanism	Updated object	Feedback
Supervised fine-tuning	Model weights	Demonstration tokens
Preference learning	Model weights	Preferred / rejected pairs
Reinforcement learning	Model weights	Trajectory rewards
Prompt optimization	Instructions or program	Development evaluations
Memory adaptation	Persistent records	Experience and review

The table organizes update mechanisms. It does not assert equivalent data efficiency or generalization.

One policy update, with the numbers exposed

Quantity	Value	Meaning
Policy	$P(a = 1) = p = 0.4$	One Bernoulli action; reward $r = a$
Parameter	$z = \log[p/(1 - p)] = -0.4055$	The logit controls the action probability
Expected-reward gradient	$dE[r]/dz = p(1 - p) = 0.24$	Average reward rises in this direction
Step of size 0.1	$z' = -0.3815; p' \approx 0.4058$	The rewarding action becomes more likely

Exact one-parameter teaching example. This is an expected policy-gradient step, before clipping, KL control or a learned reward model.

DPO uses a preference margin

Direct Preference Optimization (DPO) uses preferred output y^+ and rejected output y^- . Define $\Delta = \log[\pi(y^+)/\pi_{\text{ref}}(y^+)] - \log[\pi(y^-)/\pi_{\text{ref}}(y^-)]$; the loss is $-\log \sigma(\beta\Delta)$.

The objective uses comparative preferences. Its reward interpretation depends on the preference model and reference-policy formulation.

The policy-gradient identity has a concrete trace

Operation	Calculation or state	Consequence
Two outputs	$p=0.4$; rewards 1 and 0	Expected reward $J=p$
Logit derivative	$dp/d\theta=p(1-p)$	Exact gradient is 0.24
Score gradients	Good: 0.6; bad: -0.4	Gradient of log probability
Baseline $b=0.4$	$0.4 \times 0.36 + 0.6 \times 0.16 = 0.24$	Same expectation

Synthetic Bernoulli policy $p=\sigma(\theta)$. Treat the baseline as constant within the score-function estimator. The baseline changes variance while preserving the expected gradient here.

Worked policy update: four sampled trajectories

Trajectory	Reward	Reward – mean	Normalised advantage
1	1	+0.5	+1
2	1	+0.5	+1
3	0	-0.5	-1
4	0	-0.5	-1

The reward mean and population standard deviation are both 0.5. Divide centred rewards by this standard deviation, ignoring epsilon in this illustration. The contrast directs learning within the group; it does not validate every intermediate step. Clipping, regularisation, and token-level credit remain parts of the actual training objective.

GRPO clipping modifies the surrogate objective

Advantage	New / old probability ratio	Clipped surrogate term
+1	$0.22 / 0.20 = 1.10$	$\min(1.10, 1.10) = 1.10$
+1	$0.25 / 0.20 = 1.25$	$\min(1.25, 1.20) = 1.20$
-1	$0.18 / 0.20 = 0.90$	$\min(-0.90, -0.90) = -0.90$
-1	$0.15 / 0.20 = 0.75$	$\min(-0.75, -0.80) = -0.80$

$\epsilon = 0.2$; group advantages come from the previous slide. The mean is 0.15 before KL. Ratios are trial values in different token contexts.

An all-success group can carry no relative signal

If all four rewards are one, every centred reward is zero. After stabilised normalisation, the relative advantages remain zero.

This does not mean the trajectories are poor. It means that this particular group-relative contrast cannot distinguish them.

Sampling strategy and task difficulty therefore affect the learning signal. Methods that filter or resample groups change the data distribution and must report the resulting computational cost.

An evaluation that separates competing explanations

Matched inference budgets and a common checker make base-versus-trained comparisons interpretable. Pass-at-one, coverage across k values and solution diversity describe different aspects of the change.

Held-out problem families and format changes distinguish newly observed successful trajectories from transferable capability.

Variation across prompts and seeds limits conclusions from a single curve. Improved sampling efficiency alone does not settle the origin of a model's reasoning ability.

Clipping constrains a policy update surrogate

$$L_{\text{clip}} = \mathbb{E}[\min(\rho_{\theta}\hat{A}, \text{clip}(\rho_{\theta}, 1 - \varepsilon, 1 + \varepsilon)\hat{A})]$$

ρ_{θ} is the probability ratio of an action under the new and old policies; $\hat{A}$ is an estimated advantage. The objective is maximised.

For positive advantage, clipping limits the incentive to increase the ratio beyond $1+\varepsilon$. For negative advantage, the opposite side limits the incentive to decrease it excessively.

This surrogate does not guarantee a hard bound on every policy change; implementation and training dynamics still matter.

Group-relative reward needs a scientific objective

The worked group standardises binary rewards, but a scientific task can assign positive utility to a valid experiment that rejects a hypothesis.

Define the reward before training: measurement validity, decision improvement, resource use, and final reporting quality may matter.

Independent evaluation separates reward optimisation from scientific usefulness. A confirmation-favouring reward can encourage selective reporting even when numerical optimisation is stable.

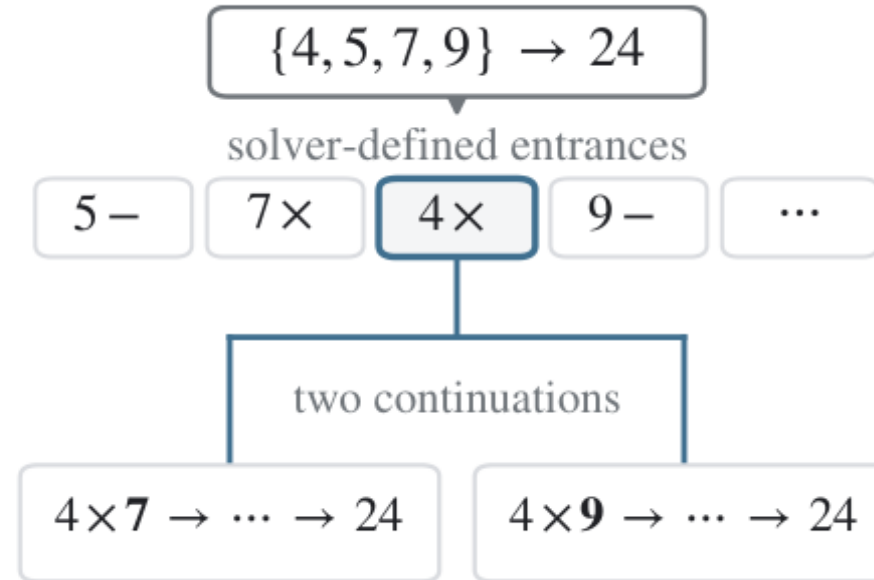
Recent research: Where RLVR narrows solution access

Reinforcement learning with verifiable rewards (RLVR) uses checkable outcome rewards. This study separates initiating a valid solution family from executing it, using a task with enumerable solution families.

It probes early trajectory choices and tests supplied prefixes and targeted interventions.

The literature snapshot is dated 11 September 2026. Findings remain conditional on the source version and evaluation conditions. Recent preprints contribute reported results with distinct validation requirements.

RLVR: supplied prefixes and solution access



A supplied solution-family entrance changes the information available to the model. Completion under this intervention diagnoses conditional execution capacity rather than unaided access to the solution family.

Recent methods in context

DAPO (NeurIPS 2025). DAPO exposes how clipping, sampling and reward handling affect large-scale policy optimization. Its ablations connect the mathematical objective to the engineering conditions needed for the reported result.

DeepSeek-R1 (Nature 2025). DeepSeek-R1 distinguishes reasoning learned through reinforcement learning from the additional stages used in the final model. Cold-start data, reward design and distillation make different contributions.

AlphaProof (Nature 2025/26). Uses reinforcement learning in a formal proof environment, including problem-specific test-time training. Adaptation cost and human formalization affect the result; a verified proposition alone does not establish broader mathematical novelty.

References (1/2)

[Eric Zelikman, Yuhuai Wu, Jesse Mu et al. \(2022\). STaR: Bootstrapping Reasoning With Reasoning](#)

[DeepSeek-AI, Daya Guo, Dejian Yang et al. \(2025\). DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning](#)

[Qiyong Yu, Zheng Zhang, Ruofei Zhu et al. \(2025\). DAPO: An Open-Source LLM Reinforcement Learning System at Scale](#)

[Yang Yue, Zhiqi Chen, Rui Lu et al. \(2025\). Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?](#)

[Zhihong Shao, Peiyi Wang, Qihao Zhu et al. \(2024\). DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models](#)

[Rulin Shao, Akari Asai, Shannon Zejiang Shen et al. \(2025\). DR Tulu: Reinforcement Learning with Evolving Rubrics for Deep Research](#)

References (2/2)

[John Schulman, Filip Wolski, Prafulla Dhariwal et al. \(2017\). Proximal Policy Optimization Algorithms](#)

[Qiancheng Zhou, Ruizhe Li \(2026\). Locked at the Entrance, Open Inside: Where RLVR Narrows the Solution Space](#)

[Rafael Rafailov, Archit Sharma, Eric Mitchell et al. \(2023\). Direct Preference Optimization: Your Language Model is Secretly a Reward Model](#)

[Thomas Hubert, Rishi Mehta, Laurent Sartran et al. \(2025\). Olympiad-level formal mathematical reasoning with reinforcement learning](#)

[Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun et al. \(2025\). Establishing Best Practices for Building Rigorous Agentic Benchmarks](#)