Reasoning RL Primer

Deep Dive into DeepSeek — Interlude

Yin Yang

Foundation Books Project

The one rule of this interlude

Training can reinforce only what the reward *exposes*.

- A pretrained policy receives a prompt, samples several completions, and receives scores for properties a *declared* grader can observe.
- Training then makes higher-scoring samples more likely.
- A final-integer checker can reward a correct integer. It does not know whether every reasoning step was faithful, concise, or readable.

The one rule of this interlude

Training can reinforce only what the reward *exposes*.

- A pretrained policy receives a prompt, samples several completions, and receives scores for properties a *declared* grader can observe.
- Training then makes higher-scoring samples more likely.
- A final-integer checker can reward a correct integer. It does not know whether every reasoning step was faithful, concise, or readable.

Goal: establish that one loop — sample, grade, compare, update, limit the change — and the vocabulary around it, before the V3 post-training and R1 chapters add recipe, equations, and implementation.

Roadmap

1. Why reinforcement learning here

Roadmap

1. Why reinforcement learning here
2. One loop before the terminology

Roadmap

1. Why reinforcement learning here
2. One loop before the terminology
3. Three layers that must stay separate

Roadmap

1. Why reinforcement learning here
2. One loop before the terminology
3. Three layers that must stay separate
4. Translations that keep their boundaries

Roadmap

1. Why reinforcement learning here
2. One loop before the terminology
3. Three layers that must stay separate
4. Translations that keep their boundaries
5. Misconceptions and the handoff

Why reinforcement learning here

Outcomes are often cheaper to check than traces

Writing one ideal trace per prompt

- Expensive.
- Can be unnecessarily restrictive — several derivations may be valid.

Outcomes are often cheaper to check than traces

Writing one ideal trace per prompt

- Expensive.
- Can be unnecessarily restrictive — several derivations may be valid.

Checking an outcome

- Integers against an answer key.
- Programs against tests.
- Proofs against a proof assistant.

Outcomes are often cheaper to check than traces

Writing one ideal trace per prompt

- Expensive.
- Can be unnecessarily restrictive — several derivations may be valid.

Checking an outcome

- Integers against an answer key.
- Programs against tests.
- Proofs against a proof assistant.

So reinforcement learning can *sample* alternatives, *score* the visible outcomes, and move probability toward the better samples.

Note the asymmetry that makes this work: the grader has to recognize a good answer, not produce one.

What automatic scoring does not remove

Warning (Task design does not disappear)
Automatic scoring replaces some target-writing work with prompts, extractors, verifiers, sampling rules, reward coverage, and movement guards.

- This is not training from random text in the R1-Zero account. The starting policy is a *pretrained* model that already produces candidate behavior — the handoff from the V3 pretraining chapter.

What automatic scoring does not remove

Warning (Task design does not disappear)
Automatic scoring replaces some target-writing work with prompts, extractors, verifiers, sampling rules, reward coverage, and movement guards.

- This is not training from random text in the R1-Zero account. The starting policy is a *pretrained* model that already produces candidate behavior — the handoff from the V3 pretraining chapter.
- The public mechanisms come from standard RL, REINFORCE, and PPO accounts, plus the group-relative specialization in DeepSeekMath and DeepSeek-R1.

What automatic scoring does not remove

Warning (Task design does not disappear)
Automatic scoring replaces some target-writing work with prompts, extractors, verifiers, sampling rules, reward coverage, and movement guards.

- This is not training from random text in the R1-Zero account. The starting policy is a *pretrained* model that already produces candidate behavior — the handoff from the V3 pretraining chapter.
- The public mechanisms come from standard RL, REINFORCE, and PPO accounts, plus the group-relative specialization in DeepSeekMath and DeepSeek-R1.
- Those sources do not disclose DeepSeek's private prompt mixture, reward weights, optimizer state, or training infrastructure.

What automatic scoring does not remove

Warning (Task design does not disappear)
Automatic scoring replaces some target-writing work with prompts, extractors, verifiers, sampling rules, reward coverage, and movement guards.

- This is not training from random text in the R1-Zero account. The starting policy is a *pretrained* model that already produces candidate behavior — the handoff from the V3 pretraining chapter.
- The public mechanisms come from standard RL, REINFORCE, and PPO accounts, plus the group-relative specialization in DeepSeekMath and DeepSeek-R1.
- Those sources do not disclose DeepSeek's private prompt mixture, reward weights, optimizer state, or training infrastructure.

Every number in this interlude's worked example is a teaching fixture. [OPEN: private prompt mixture and reward weights]

One loop before the terminology

The congruence fixture, priced by a local grader

A prompt asks for the unique integer $n \in \{0, \ldots, 34\}$ with $n \bmod 5 = 2$ and $n \bmod 7 = 3$, with the final answer in tags. The answer is 17. The reward program assigns 1 for a correct extracted integer and 0.1 for the required format.

Sample	What the reward program observes	Reward	Relative pressure
o_1	Correct tagged answer 17	1.1	Above group; reinforce

The congruence fixture, priced by a local grader

A prompt asks for the unique integer $n \in \{0, \ldots, 34\}$ with $n \bmod 5 = 2$ and $n \bmod 7 = 3$, with the final answer in tags. The answer is 17. The reward program assigns 1 for a correct extracted integer and 0.1 for the required format.

Sample	What the reward program observes	Reward	Relative pressure
o_1	Correct tagged answer 17	1.1	Above group; reinforce
o_2	Wrong tagged answer 32	0.1	Below group; discourage

The congruence fixture, priced by a local grader

A prompt asks for the unique integer $n \in \{0, \ldots, 34\}$ with $n \bmod 5 = 2$ and $n \bmod 7 = 3$, with the final answer in tags. The answer is 17. The reward program assigns 1 for a correct extracted integer and 0.1 for the required format.

Sample	What the reward program observes	Reward	Relative pressure
o_1	Correct tagged answer 17	1.1	Above group; reinforce
o_2	Wrong tagged answer 32	0.1	Below group; discourage
o_3	Prose the extractor cannot grade	0.0	Below group; discourage

The congruence fixture, priced by a local grader

A prompt asks for the unique integer $n \in \{0, \ldots, 34\}$ with $n \bmod 5 = 2$ and $n \bmod 7 = 3$, with the final answer in tags. The answer is 17. The reward program assigns 1 for a correct extracted integer and 0.1 for the required format.

Sample	What the reward program observes	Reward	Relative pressure
o_1	Correct tagged answer 17	1.1	Above group; reinforce
o_2	Wrong tagged answer 32	0.1	Below group; discourage
o_3	Prose the extractor cannot grade	0.0	Below group; discourage
o_4	Correct bare answer 17	1.0	Above group; reinforce

The congruence fixture, priced by a local grader

A prompt asks for the unique integer $n \in \{0, \ldots, 34\}$ with $n \bmod 5 = 2$ and $n \bmod 7 = 3$, with the final answer in tags. The answer is 17. The reward program assigns 1 for a correct extracted integer and 0.1 for the required format.

Sample	What the reward program observes	Reward	Relative pressure
o_1	Correct tagged answer 17	1.1	Above group; reinforce
o_2	Wrong tagged answer 32	0.1	Below group; discourage
o_3	Prose the extractor cannot grade	0.0	Below group; discourage
o_4	Correct bare answer 17	1.0	Above group; reinforce

The mean reward is 0.55: a comparison baseline, not a passing score. [DERIVED: this primer's teaching fixture]

One iteration of the loop

prompt + current policy

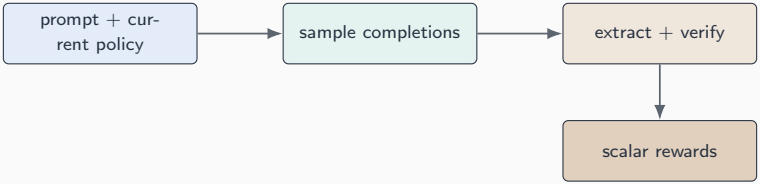
One iteration of the loop



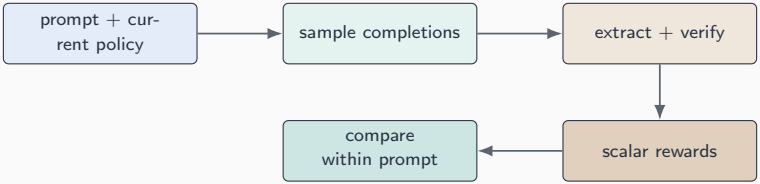
One iteration of the loop



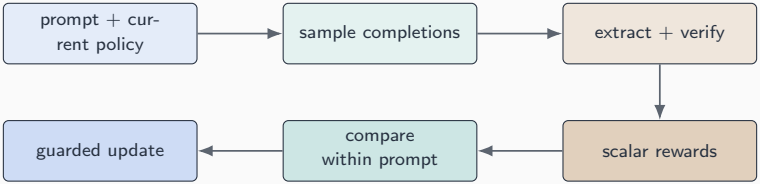
One iteration of the loop



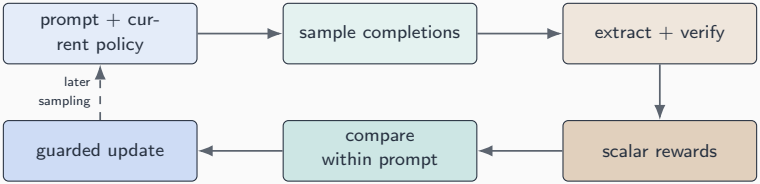
One iteration of the loop



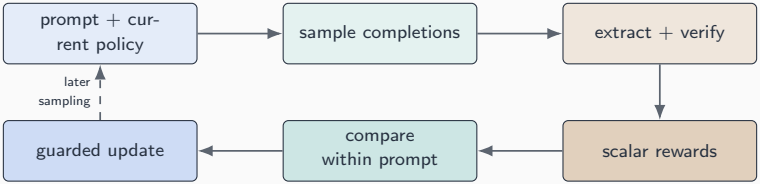
One iteration of the loop



One iteration of the loop



One iteration of the loop



The grader sees *declared output properties*, not every property of the reasoning. The update changes future sampling probabilities; it does not paste a response into the model.

If every sampled reward is equal, the comparison stage has no relative direction to pass to the update — whether the answers were all right or all wrong.

Three layers that must stay separate

Reward, comparison, movement control

Layer	What it does	What it cannot do
Task and reward	Extractor and verifier assign credit for the properties they test	Calling a reward an objective does not complete its specification

Reward, comparison, movement control

Layer	What it does	What it cannot do
Task and reward	Extractor and verifier assign credit for the properties they test	Calling a reward an objective does not complete its specification
Comparison and learning	Above-baseline rewards get positive advantage, below-baseline negative	Advantage interprets an existing reward; it is not another grader

Reward, comparison, movement control

Layer	What it does	What it cannot do
Task and reward	Extractor and verifier assign credit for the properties they test	Calling a reward an objective does not complete its specification
Comparison and learning	Above-baseline rewards get positive advantage, below-baseline negative	Advantage interprets an existing reward; it is not another grader
Movement control	An old-policy ratio and a reference-policy KL penalty limit change	Preserve behavior; cannot repair an incomplete checker or prove reasoning faithful

Reward, comparison, movement control

Layer	What it does	What it cannot do
Task and reward	Extractor and verifier assign credit for the properties they test	Calling a reward an objective does not complete its specification
Comparison and learning	Above-baseline rewards get positive advantage, below-baseline negative	Advantage interprets an existing reward; it is not another grader
Movement control	An old-policy ratio and a reference-policy KL penalty limit change	Preserve behavior; cannot repair an incomplete checker or prove reasoning faithful

Kullback–Leibler divergence measures distribution drift on a declared probability surface. It does not check mathematical correctness.

The PPO bridge: clipping credits, it does not constrain

PPO separates two questions: *which* sampled completions receive update pressure, and *how much objective credit* the current-to-old policy ratio receives.

- Positive advantage, probability $0.10 \to 0.13$: ratio 1.30. With an upper clip at 1.20, the objective credits only 1.20 times the advantage.

The PPO bridge: clipping credits, it does not constrain

PPO separates two questions: *which* sampled completions receive update pressure, and *how much objective credit* the current-to-old policy ratio receives.

- Positive advantage, probability $0.10 \rightarrow 0.13$: ratio 1.30. With an upper clip at 1.20, the objective credits only 1.20 times the advantage.
- Negative advantage, probability $0.10 \rightarrow 0.07$: ratio 0.70. With a lower clip at 0.80, the objective stops granting additional credit for pushing that sample farther down.

The PPO bridge: clipping credits, it does not constrain

PPO separates two questions: *which* sampled completions receive update pressure, and *how much objective credit* the current-to-old policy ratio receives.

- Positive advantage, probability $0.10 \rightarrow 0.13$: ratio 1.30. With an upper clip at 1.20, the objective credits only 1.20 times the advantage.
- Negative advantage, probability $0.10 \rightarrow 0.07$: ratio 0.70. With a lower clip at 0.80, the objective stops granting additional credit for pushing that sample farther down.

Clipping is a soft objective guard, not a hard constraint on the realized ratio. [REPORTED: PPO, arXiv:1707.06347]

The PPO bridge: clipping credits, it does not constrain

PPO separates two questions: *which* sampled completions receive update pressure, and *how much objective credit* the current-to-old policy ratio receives.

- Positive advantage, probability $0.10 \rightarrow 0.13$: ratio 1.30. With an upper clip at 1.20, the objective credits only 1.20 times the advantage.
- Negative advantage, probability $0.10 \rightarrow 0.07$: ratio 0.70. With a lower clip at 0.80, the objective stops granting additional credit for pushing that sample farther down.

Clipping is a soft objective guard, not a hard constraint on the realized ratio. [REPORTED: PPO, arXiv:1707.06347]

Actor–critic PPO commonly learns advantages with a critic and Generalized Advantage Estimation. GRPO replaces that *source* with within-prompt comparison while keeping a PPO-shaped clipped-ratio guard. [REPORTED: DeepSeekMath, arXiv:2402.03300]

Translations that keep their boundaries

Classical RL, bounded translation

Classical term	Language-model reading	Main-path use
State or observation	Prompt plus generated prefix	Conditions the next-token distribution

Classical RL, bounded translation

Classical term	Language-model reading	Main-path use
State or observation	Prompt plus generated prefix	Conditions the next-token distribution
Action	Next token	One step in a completion

Classical RL, bounded translation

Classical term	Language-model reading	Main-path use
State or observation	Prompt plus generated prefix	Conditions the next-token distribution
Action	Next token	One step in a completion
Trajectory or episode	Response through its stop	Rollout to extract and score

Classical RL, bounded translation

Classical term	Language-model reading	Main-path use
State or observation	Prompt plus generated prefix	Conditions the next-token distribution
Action	Next token	One step in a completion
Trajectory or episode	Response through its stop	Rollout to extract and score
Reward or return	Immediate or accumulated feedback	Usually a completion-level score here

Classical RL, bounded translation

Classical term	Language-model reading	Main-path use
State or observation	Prompt plus generated prefix	Conditions the next-token distribution
Action	Next token	One step in a completion
Trajectory or episode	Response through its stop	Rollout to extract and score
Reward or return	Immediate or accumulated feedback	Usually a completion-level score here
Value or critic	Estimate of future return	Common in RL, but not needed for the primer's group baseline

Classical RL, bounded translation

Classical term	Language-model reading	Main-path use
State or observation	Prompt plus generated prefix	Conditions the next-token distribution
Action	Next token	One step in a completion
Trajectory or episode	Response through its stop	Rollout to extract and score
Reward or return	Immediate or accumulated feedback	Usually a completion-level score here
Value or critic	Estimate of future return	Common in RL, but not needed for the primer's group baseline
Environment	Prompt source, generation, tools, and grader	Declared task protocol

Classical RL, bounded translation

Classical term	Language-model reading	Main-path use
State or observation	Prompt plus generated prefix	Conditions the next-token distribution
Action	Next token	One step in a completion
Trajectory or episode	Response through its stop	Rollout to extract and score
Reward or return	Immediate or accumulated feedback	Usually a completion-level score here
Value or critic	Estimate of future return	Common in RL, but not needed for the primer's group baseline
Environment	Prompt source, generation, tools, and grader	Declared task protocol

Tool-using or multi-turn agents can add several interactions and rewards. The reasoning chapters begin with terminal completion rewards because that is what their examples require.

Three post-training signals, three evidence paths

Method	Training evidence	Direct pressure
SFT	Prompt paired with a trusted response	Increase likelihood of its target tokens

Three post-training signals, three evidence paths

Method	Training evidence	Direct pressure
SFT	Prompt paired with a trusted response	Increase likelihood of its target tokens
RL	Sampled responses paired with rewards	Favor higher-reward samples under a guarded update

Three post-training signals, three evidence paths

Method	Training evidence	Direct pressure
SFT	Prompt paired with a trusted response	Increase likelihood of its target tokens
RL	Sampled responses paired with rewards	Favor higher-reward samples under a guarded update
Distillation	Teacher outputs or distributions	Transfer selected teacher behavior

Three post-training signals, three evidence paths

Method	Training evidence	Direct pressure
SFT	Prompt paired with a trusted response	Increase likelihood of its target tokens
RL	Sampled responses paired with rewards	Favor higher-reward samples under a guarded update
Distillation	Teacher outputs or distributions	Transfer selected teacher behavior

For the congruence prompt: SFT would imitate one worked answer, RL would compare the four sampled completions, and distillation would learn from a stronger teacher's response or distribution.

Pipelines can combine all three. Combining them does not make them interchangeable, and does not erase their distinct evidence paths.

Misconceptions and the handoff

Shortcuts to drop before the equations

Shortcut	Correction
Reward proves truth.	Reward certifies only the grader's tested properties.
Advantage is answer quality.	Advantage compares reward with a baseline.
The reference policy supplies the answer.	It anchors drift; the reward supplies task feedback.
Old and reference policies have one job.	The old policy anchors a ratio; the reference policy anchors a drift penalty.
Pure RL starts from random weights.	R1-Zero starts from a pretrained base; *pure* marks no preliminary cold-start SFT in that account.
A rollout must be physical.	Here, a sampled text completion is a rollout.
RL creates missing capability on demand.	Results depend on the starting policy, prompts, exploration, and reward coverage.

From concept to code: what “pure” actually marks

The release itself states the starting point and the ordering, which is what makes the “random weights” shortcut checkable rather than arguable.

from `DeepSeek-R1/README.md`

```
DeepSeek-R1-Zero, a model trained via large-scale reinforcement
learning (RL) without supervised fine-tuning (SFT) as a
preliminary step, ...
... we introduce DeepSeek-R1, which incorporates cold-start data
before RL.
```

From concept to code: what “pure” actually marks

The release itself states the starting point and the ordering, which is what makes the “random weights” shortcut checkable rather than arguable.

from **DeepSeek-R1/README.md**

```
DeepSeek-R1-Zero, a model trained via large-scale reinforcement
learning (RL) without supervised fine-tuning (SFT) as a
preliminary step, ...
... we introduce DeepSeek-R1, which incorporates cold-start data
before RL.
```

“Without SFT as a preliminary step” is a statement about *ordering*, not about initialization: the base model is pretrained in both accounts. R1 then adds the cold-start stage that R1-Zero omits. [REPORTED: DeepSeek-R1 README.md]

Where this primer stops, and what picks it up

- **release-facing pure-RL, pipeline, and distillation claims** → `DeepSeek-R1/README.md`

Where this primer stops, and what picks it up

- **release-facing pure-RL, pipeline, and distillation claims** → `DeepSeek-R1/README.md`
- **the R1 report behind those claims** → `DeepSeek-R1/DeepSeek_R1.pdf`

Where this primer stops, and what picks it up

- **release-facing pure-RL, pipeline, and distillation claims** → `DeepSeek-R1/README.md`
- **the R1 report behind those claims** → `DeepSeek-R1/DeepSeek_R1.pdf`
- **clipped-ratio objective guard** → `arXiv:1707.06347 (PPO)`

Where this primer stops, and what picks it up

- **release-facing pure-RL, pipeline, and distillation claims** → `DeepSeek-R1/README.md`
- **the R1 report behind those claims** → `DeepSeek-R1/DeepSeek_R1.pdf`
- **clipped-ratio objective guard** → `arXiv:1707.06347 (PPO)`
- **group-relative advantage source** → `arXiv:2402.03300 (DeepSeekMath)`

Where this primer stops, and what picks it up

- **release-facing pure-RL, pipeline, and distillation claims** → `DeepSeek-R1/README.md`
- **the R1 report behind those claims** → `DeepSeek-R1/DeepSeek_R1.pdf`
- **clipped-ratio objective guard** → `arXiv:1707.06347 (PPO)`
- **group-relative advantage source** → `arXiv:2402.03300 (DeepSeekMath)`

Sample answers, score what the grader observes, compare samples for the same prompt, update toward the better ones, and limit the change.

This primer stops at conceptual vocabulary and qualitative update direction; it does not specify an executable optimizer. Chapter 7 separates SFT, preference evidence, RL, and distillation. Chapter 8 begins the mathematical and implementation track. Chapter 9 adds the cold-start, refinement, distillation, and release-facing stages.