READ MODE · ANSWER FIRST

# How pretraining changes reinforcement learning reasoning returns

See how pretraining loss and exposure shape RL returns, compute allocation, policy evolution, easy and hard tasks, and transfer limits.

CHEAT SHEET · 6 ESSENTIAL IDEAS

## The whole story in 6 lines

Pretraining shapes the starting policy and RL's local value; task difficulty and domain limits decide how far the lesson travels.

1. Chess makes exposure controllable and rewards exact enough to inspect the full pretraining to RL pipeline.
2. Loss tracks learned fit while token exposure records how long the checkpoint trained, so they provide different signals.
3. Lower loss predicts a higher local reward level and more tokens predict a steeper local RL slope in the studied range.
4. Low budgets favor stronger initialization, while fitted frontiers give RL a larger share as total compute grows.
5. Easy states mostly amplify correct modes, while hard states mix useful tail discovery with wrong mode amplification.
6. Math showed the same pattern, but chess exponents and policy splits are not universal.

STAGE 1 OF 8

# Setup


Two models receive the same reinforcement learning budget. One climbs quickly while the other stalls, so the important difference must have existed before that RL run began.

Pretraining means learning patterns by predicting the next token. Loss summarizes prediction error on held out data, so a lower value means the model fits those patterns better.

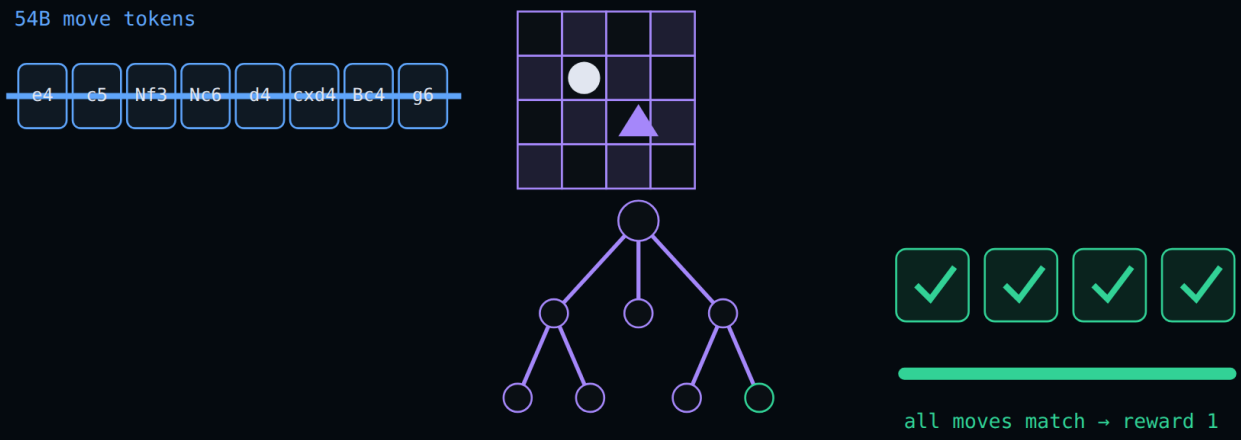
Reinforcement learning, or RL, updates behavior from outcome rewards. A policy is the probability distribution over possible next actions, such as the legal moves from one chess position.

We will build a controlled chess laboratory, separate two pretraining signals, follow later RL curves, divide compute, inspect policy changes, and finish at the transfer boundary.

The puzzle is now framed as one connected training story. Measured findings will stay distinct from illustrative teaching values. Now let us start with the controlled laboratory.

STAGE 2 OF 8

# A controlled chess laboratory




02 A CONTROLLED CHESS LABORATORY

Setup gave us a mystery about unequal RL returns. The study needs a domain where earlier experience can be counted and each later decision can be checked without a learned judge.

The authors serialize 2022 Lichess Blitz and Rapid games into move tokens. Any valid prefix determines one board state, while the compact action space lists the legal next moves.

A puzzle supplies one designated best continuation and ends after a wrong solver move. Which property removes ambiguity from the RL reward?

PAUSE AND PREDICT

**Which property removes ambiguity from the RL reward?**

- **Every solver move has an exact check**
- **Millions of people play chess**
- **A board is easy to draw**

Skip this check

Between pretraining and RL, supervised fine tuning teaches a reasoning format. Sampled continuations merge into a prefix tree, then serialize as candidate paths before the model commits to one move.

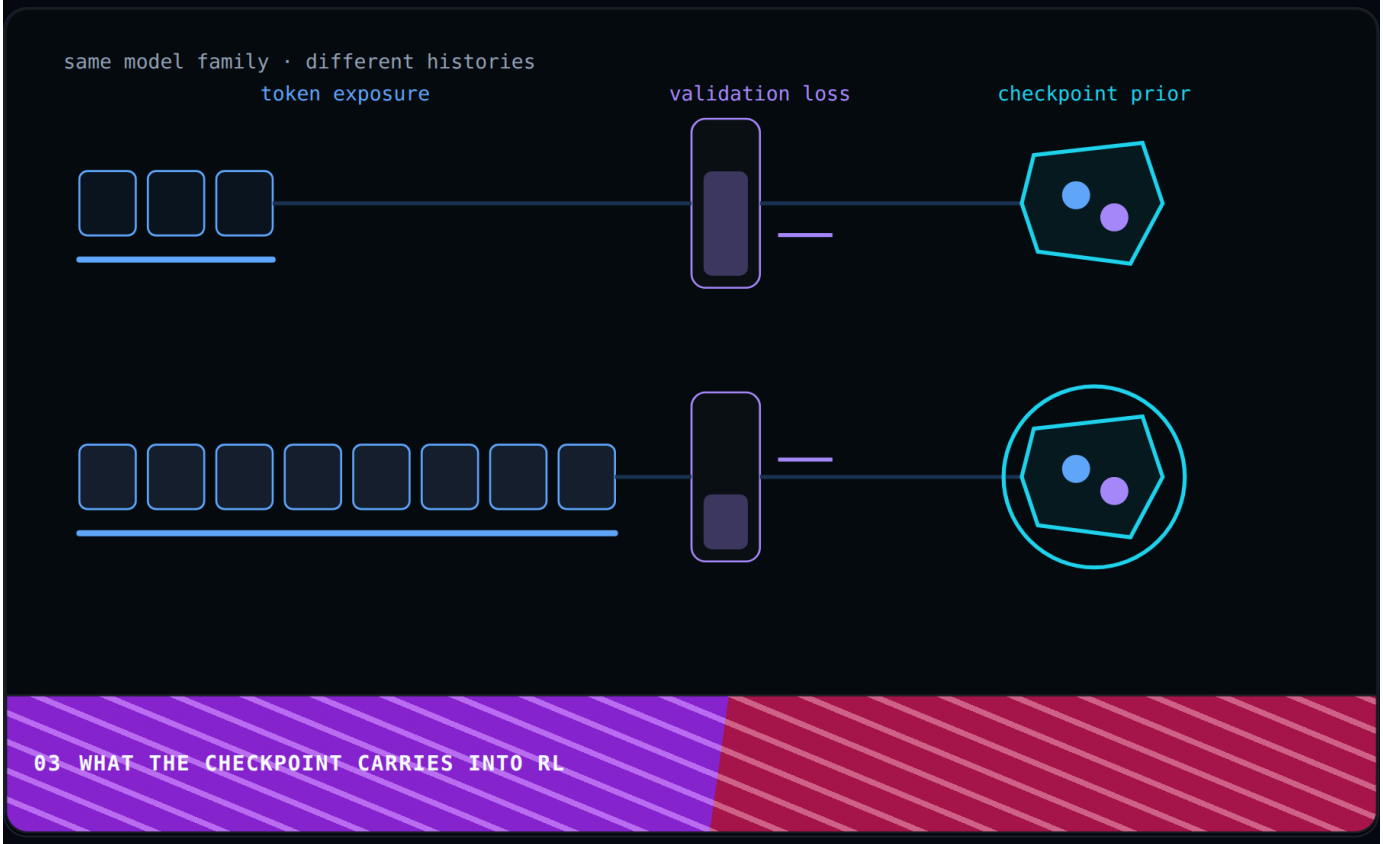
The full experiment now connects human game pretraining, synthetic reasoning traces, and binary outcome RL. One wrong required move makes reward zero, while a complete matching solution earns

one.

The paper sweeps 36 pretraining and RL combinations across ten models from 5 million to 1 billion parameters. Next we will separate the signals recorded before RL.

STAGE 3 OF 8

# What the checkpoint carries into RL




03 WHAT THE CHECKPOINT CARRIES INTO RL

The chess laboratory gave every training phase a clear object. Before RL starts, each checkpoint carries both a training history and a measured ability to predict held out human moves.

Token exposure records how much pretraining data passed through the model. A longer tape means more examples were processed, but it does not directly say how low prediction error became.

Validation loss asks a different question. It measures next move prediction error on held out games, so lower loss marks a better fit to the human play distribution.

Both signals now enter the same checkpoint without collapsing into one label. The paper later links loss to the post RL reward level and exposure to the local improvement slope.

Here is the useful distinction. Loss describes predictive fit, while token exposure describes training duration and data scale across the experiment. Next we will see how each signal maps onto an RL curve.

# Pretraining predicts the local RL curve

**★ If you remember one thing · Longer, lower loss pretraining can raise the later RL level and steepen its local improvement curve.**


04 PRETRAINING PREDICTS THE LOCAL RL CURVE

We now have two pretraining signals, so the paper fits later pass at one against RL compute. The studied runs cover the early, unsaturated region rather than an unlimited training horizon.

Lower pretraining loss predicts a higher fitted performance level at a fixed RL compute point. This relationship becomes more monotone as the chosen reference compute grows.

More token exposure could only lift the starting point, or it could change how efficiently later RL compute pays off. Which visual outcome matches the reported local trend?

PAUSE AND PREDICT

**Which visual outcome matches the reported local trend?**

- **The longer trained curve also becomes steeper**
- **Both curves remain parallel**
- **The shorter trained curve overtakes it**

Skip this check

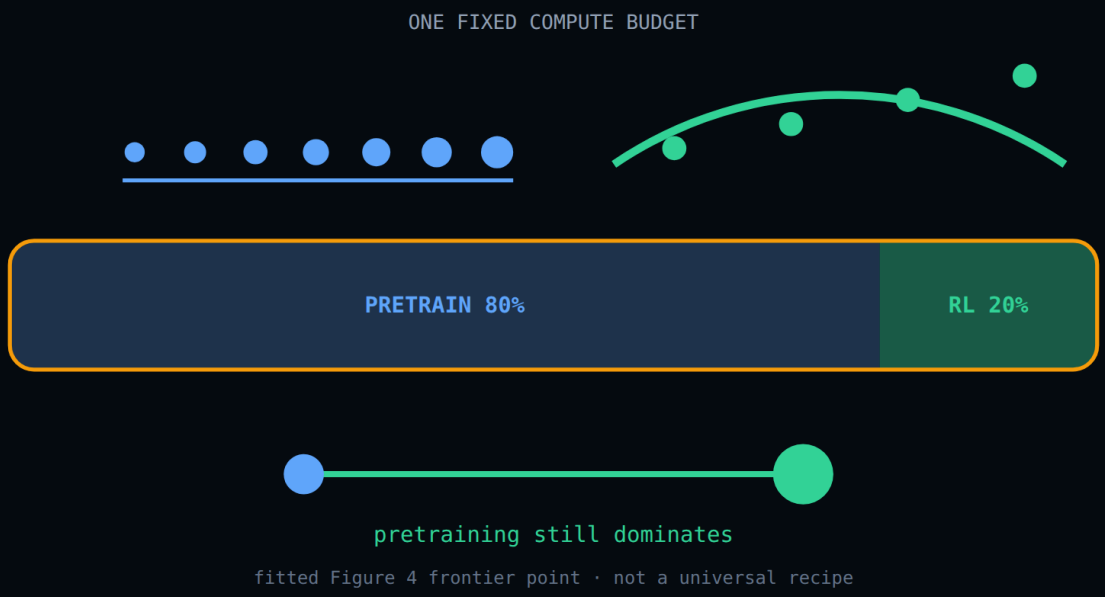
The contrast now shows both relationships together. Stronger pretraining enters at a higher fitted level and gains faster with additional RL compute, so the separation widens across the local range.

This is a local empirical law, not a promise of endless linear improvement across every training regime. Easy tasks can saturate and compress slope estimates. Next we will turn the relationship into a compute allocation choice.

STAGE 5 OF 8

# How the best compute split moves

**Try it in Watch mode.** Find the fitted frontier point where RL receives more than one quarter of compute.




## 05 HOW THE BEST COMPUTE SPLIT MOVES

The local scaling link makes pretraining and RL interdependent. Under one fixed budget, every extra unit spent building the checkpoint leaves one fewer unit for outcome learning.

At low total compute, the model is initialization limited. Starting RL from a much weaker checkpoint provides more updates, but those updates do not recover the skipped pretraining value.

As total compute grows, additional pretraining brings smaller marginal returns while the stronger policy can use feedback more effectively. Where should a larger fraction move?

PAUSE AND PREDICT

### Where should a larger fraction move?

- Toward reinforcement learning
- Entirely toward pretraining
- Entirely toward supervised fine tuning

Skip this check

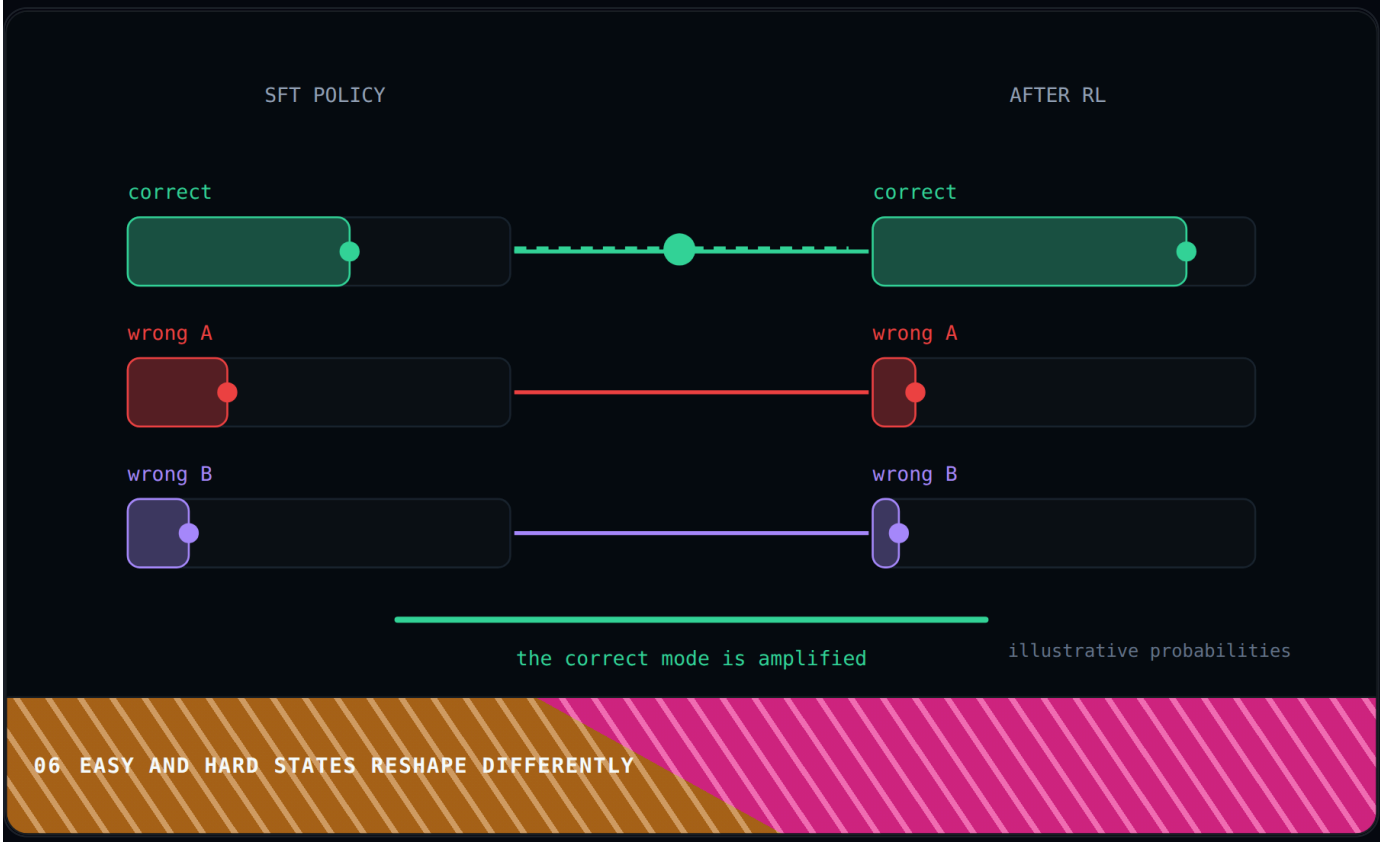
The paper combines its fitted laws to extrapolate frontier points. The reported optimal RL share rises from about 20 percent at 50 million parameters to about 28 percent at 680 million.

Switch the Total budget control across both fitted points. Compare the conserved bar and the downstream verdict, while remembering that these values come from model based extrapolation.

The lesson is directional rather than universal. Strong initialization dominates early, then RL earns a larger proportional share as compute grows. Next we will inspect what those RL updates change inside the policy.

STAGE 6 OF 8

# Easy and hard states reshape differently




06 EASY AND HARD STATES RESHAPE DIFFERENTLY

The compute split tells us when RL becomes valuable, but not what it changes. The researchers reconstruct a probability distribution over legal moves for each puzzle state before and during RL.

On easier puzzles, the correct move already sits near the top under supervised fine tuning. RL usually concentrates more probability on that existing correct mode.

Hard states often begin with the correct move buried in the tail. Does RL only sharpen the already preferred wrong move?

PAUSE AND PREDICT

**Does RL only sharpen the already preferred wrong move?**

- **No, it can discover the correct tail move or amplify a wrong mode**
- **Yes, it only sharpens the top move**

Skip this check

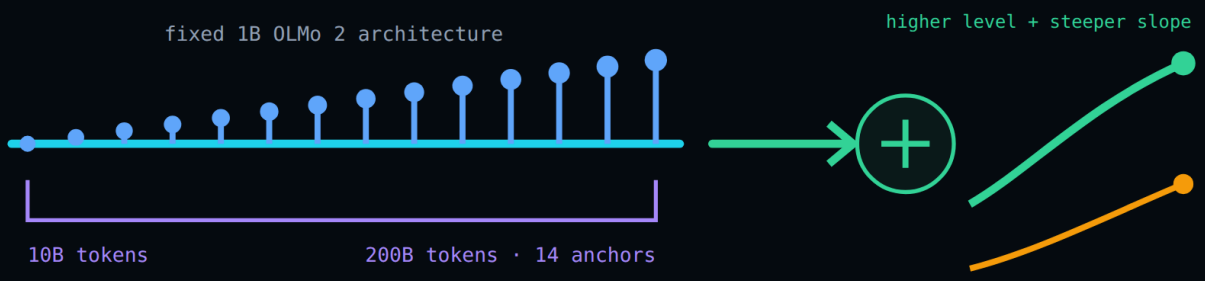
The before and after bars now diverge by difficulty. Easy states mostly amplify a correct leader, while hard states can promote a buried correct move or push even more mass into a wrong mode.

Switch the Puzzle difficulty control between Easy and Hard. Compare where the correct move begins, where it ends, and which wrong move still absorbs probability after training.

RL is therefore state dependent, not a uniform sharpening operation. Its harder task gains arrive with wrong mode risk, which helps explain limited pass at many improvement. Next we will test transfer and

limits.

# What transferred beyond chess




07 WHAT TRANSFERRED BEYOND CHESS

We just saw that difficulty changes policy evolution inside chess. The final question is whether the scaling pattern survives when moves become natural language solutions and the model becomes much larger.

The authors fork fourteen checkpoints from one fixed OLMo 2 run between 10 billion and 200 billion pretraining tokens. Each anchor receives math fine tuning and GRPO.

The architecture stays fixed while exposure grows, which isolates checkpoint age more cleanly. Which part of the chess result should count as meaningful transfer?

PAUSE AND PREDICT

**Which part of the chess result should count as meaningful transfer?**

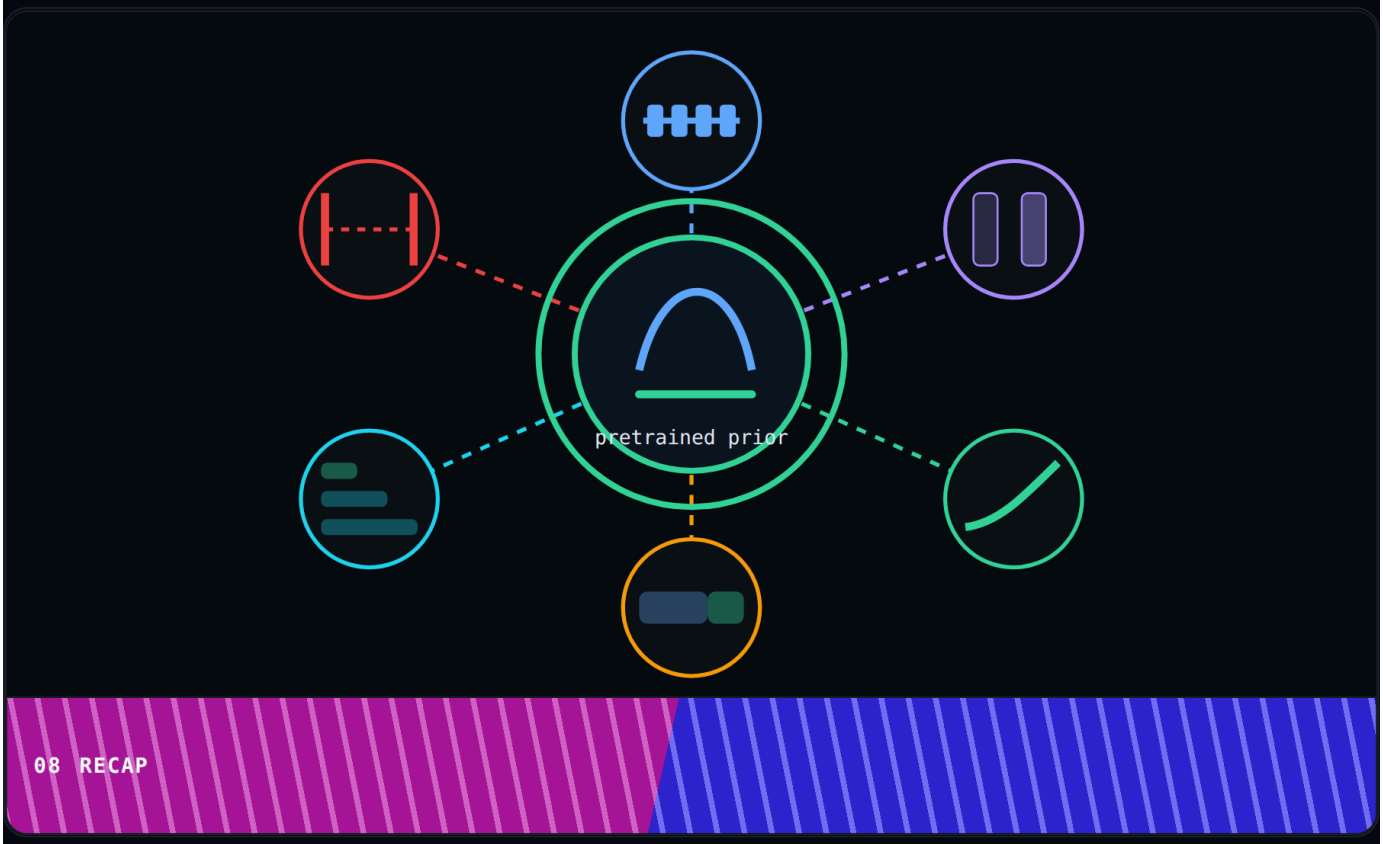
- **The direction of the loss, level, and slope relationships**
- **The exact chess exponents and policy fractions**

Skip this check

The math anchors show the same direction. Longer trained checkpoints reach higher post RL performance and improve faster across the studied local range. The paper calls this early transfer evidence.

The boundary matters. Chess uses 81 tokens, exact binary rewards, unique solutions, and models no larger than 1 billion parameters. Natural language also mixes world knowledge with fluent expression. Now let us step back and connect the whole picture.

# Recap




We started with a chess laboratory where data exposure is countable, legal actions are explicit, and every required puzzle move has an exact check.

Then we separated pretraining loss from token exposure. One measures held out predictive fit, while the other records how much data the checkpoint processed.

Next we connected those signals to later RL. Lower loss predicted a higher local level, and more exposure predicted a steeper local improvement rate.

The compute frontier turned that relationship into a budget choice. Strong initialization dominated early, while fitted larger scale points assigned RL a growing share.

Policy inspection showed why average sharpening is incomplete. Easy states amplify correct modes, while hard states mix tail discovery with wrong mode amplification.

Finally, a fixed size math run echoed the qualitative relationship. Exact chess coefficients stayed behind because reward structure, vocabulary, knowledge, and scale all changed.

The complete lesson is now connected. Pretraining shapes where RL starts and how it improves, while task difficulty and domain boundaries determine what RL can safely amplify.