How RL Reshapes LLM Reasoning: Transferability, Coverage, and Scaling Laws

Anonymous authors
Paper under double-blind review

Abstract

Recent studies on reinforcement learning (RL) report seemingly conflicting evidence about large language model (LLM) reasoning. Training on mathematics can improve performance in other domains, yet gains in Pass@1 can coincide with lower Pass@N than the base model. This raises a fundamental question: does RL expand an LLM's reasoning boundary, or merely reweight its existing reasoning space? We revisit these phenomena across Qwen and Gemma model families, revealing that cross-domain transfer and forgetting both exist, while coverage at large sampling budgets increases on some tasks and decreases on others. Detailed analysis of solution traces before and after RL indicates a shift in the reasoning strategies the model employs, motivating a two-stage autoregressive policy model that separates *strategy selection* from problem-specific execution. Within this framework, we prove how RL's implicit bias reshapes strategy preferences, allowing gains on some tasks while suppressing strategies required by others. This mechanism can also broaden or narrow finite-budget coverage even without expanding strategy support. We further provide theoretical justifications for log-sigmoid and log-linear scaling laws in RL compute, and evaluate their predictive power. Together, these results connect changes in strategy selection to cross-domain transfer, reasoning coverage, and compute scaling.

1 Introduction

Reinforcement learning (RL) has improved the performance of large language models (LLMs) on reasoning tasks (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025). RL updates a model using reward feedback on sampled responses, rather than training it to imitate demonstrations. Its success on mathematics and coding raises a broader question about how RL reshapes LLM reasoning.

What accounts for these gains remains debated. RL may teach a model new ways to reason, or make it more likely to use reasoning it can already produce. For example, RL training on mathematical problems can improve performance in other domains (Huan et al., 2025), but such gains are compatible with either interpretation. Sampling provides another perspective. Gains in one-shot accuracy (Pass@1) can accompany lower Pass@N, the fraction of problems solved within N sampled attempts (Yue et al., 2025), while prolonged training has been reported to broaden coverage (Liu et al., 2025). These findings motivate the central question. Does RL expand an LLM's reasoning boundary, or merely reweight its existing reasoning space?

Our experiments show that training can improve performance in its own domain while reducing performance on other tasks or lowering Pass@1024 (Figures 2 and 3). Such outcomes leave open what has changed in the model's reasoning. A higher score may reflect new solution strategies, better execution of familiar strategies, or a shift in which strategies the model selects. Distinguishing these possibilities requires examining the solution trajectories, rather than inferring capability growth from accuracy alone.

Comparing Qwen trajectories before and after RL indicates a preference shift in reasoning strategies the model employs (Section 3.1). This motivates a two-stage policy model that separates *strategy selection*, from problem-specific execution. Within this framework, we characterize how the implicit bias of RL reshapes strategy preferences, improving performance on some tasks while suppressing strategies required by others. We also contrast this mechanism against supervised fine-tuning (SFT).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

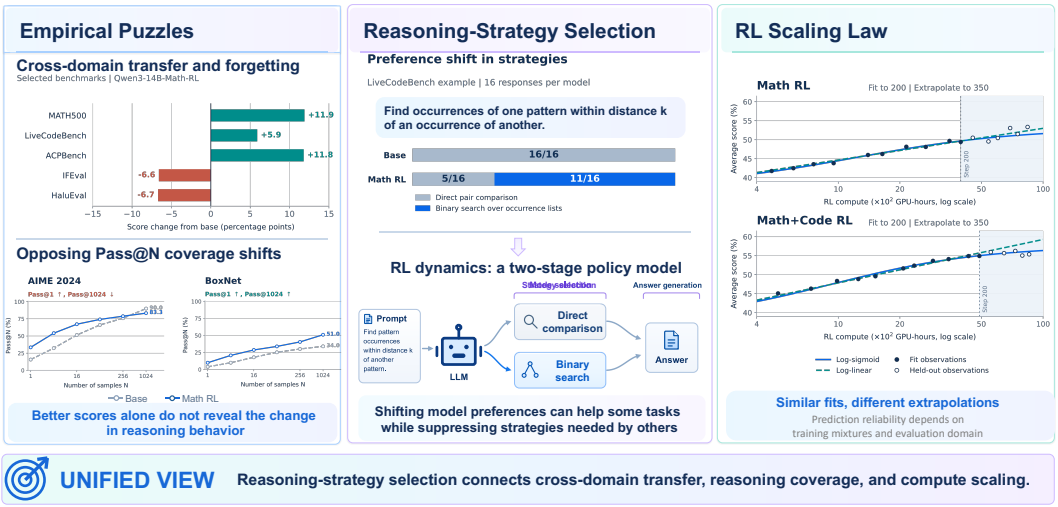


Figure 1: Left: RL gains coexist with forgetting and opposing Pass@N shifts. Center: strategy shifts motivate a two-stage model. Right: similar scaling fits extrapolate differently.

Our focus on how strategy preferences generalize across tasks complements the analyses of training-prompt sharpening and coverage (Chen et al., 2026b; 2025). Crucially, this reweighting mechanism explains coverage shifts without introducing new strategies. Pass@N at large N is especially sensitive to problems with low success probabilities. Amplifying or suppressing rare valid strategies can broaden or narrow benchmark coverage, even as Pass@1 improves.

Beyond transferability and reasoning boundary, recent studies (Khatri et al., 2025; Meta AI, 2026) highlight another critical frontier: how does performance scale with RL compute? Our framework also connects strategy selection to scaling laws, showing selection dynamics admit log-sigmoid and log-linear approximations.

Our contributions are summarized as follows:

- We document task-specific RL transfer and forgetting alongside divergent coverage shifts, with outcomes dictated by the training data, evaluation benchmark, and base model (Section 2).
- We analyze Qwen solution trajectories and find changes in which strategies are sampled for a question, motivating a separation of strategy selection from execution (Sections 3.1 and 3.2).
- Theoretically, we characterize RL's implicit bias in strategy selection and derive conditions for transfer, forgetting, and expansion or shrinkage of finite-budget coverage (Sections 4 and 5).
- We further bound the approximation error within a local regime for log-sigmoid and log-linear scaling laws, and evaluate their extrapolation performance on future training, showing that predictive reliability varies with the training mixture and evaluation domain (Section 6).

2 Empirical Evidence on Transfer and Reasoning Coverage

The extent to which RL transfers beyond its training domain and expands the reasoning boundary remains actively debated. In this section, we revisit these phenomena across various model families and task domains to show that these seemingly contradictory findings can be reconciled.

2.1 Experimental Setup

Models and datasets. Our experiments use Qwen3-14B-Base (Yang et al., 2025) and Gemma-3-12B-IT (Gemma Team, 2025) as base models. Both models are trained on 48K mathematics problems from UniReason (Huan et al., 2025). For Qwen, we additionally consider a variant by augmenting this dataset with 5K curated coding problems from Eurus-2-RL (Cui et al., 2025). For comparison, we also conduct SFT on both models using 40K mathematics demonstrations obtained by rejection sampling on Qwen3-32B responses (Huan et al., 2025).

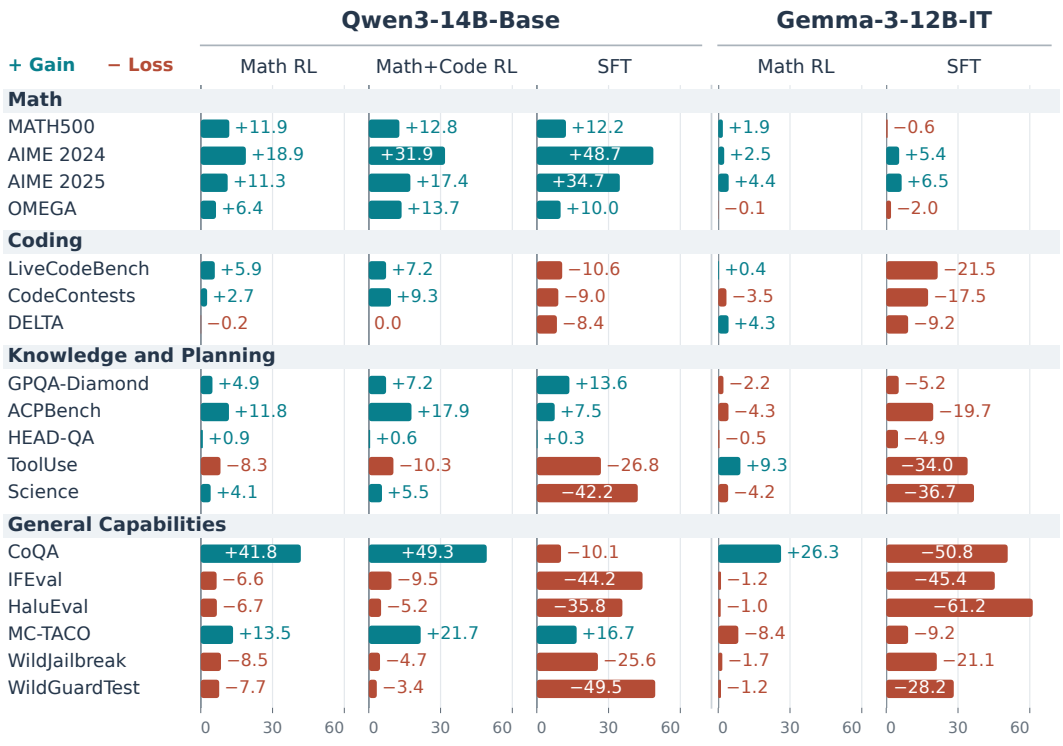


Figure 2: **Transfer and forgetting depend on the benchmark, training data, and base model.** Each column reports performance shifts from the base model after training. RL consistently improves in-domain performance, but transfer behavior beyond math and coding domain varies significantly. In contrast, SFT exhibits catastrophic forgetting on broad domains.

Training setup. For all RL experiments, we train with GRPO (Shao et al., 2024) using verl (Sheng et al., 2024) for 200 steps at a learning rate of 10^{-6}. Each step samples 512 prompts and generates 16 responses per prompt (up to 16,384 tokens), followed by one actor update epoch with a minibatch size of 128. For SFT, we train 1.5 epochs using a batch size of 512 and a learning rate of 5×10^{-5}. Further training details are given in Appendix E.

Evaluation protocol. To comprehensively evaluate transfer behavior, we select benchmarks across different domains: Math, Coding, Knowledge and Planning, General Capabilities. Our default inference settings use temperature 0.6, top-p 0.95, and a maximum generation length of 32,768, though specific benchmarks retain their original protocols such as greedy decoding. Appendix E.2 gives the benchmark list and complete task-specific sampling and scoring details.

2.2 Cross-domain transfer and forgetting

Figure 2 illustrates the positive transfer and negative forgetting results. While RL consistently improve in-domain math performance, out-of-domain (OOD) behavior diverges significantly. Incorporating coding data into training (Qwen3-Math-Code-RL) improves coding tasks but fails to yield universal transfer. Moreover, despite identical training data, Qwen3-Math-RL and Gemma-3-Math-RL shift in opposite directions on many OOD tasks. SFT results exhibit similar model dependence: Qwen3-SFT shows selective transfer on knowledge & planning despite broad forgetting elsewhere, whereas Gemma-3-SFT suffers universal OOD degradation.

Together, these results indicate that training-domain success does not reliably predict OOD outcomes. Instead, **transferability emerges from the complex interaction among training data, evaluation benchmark, and base model**. Complete scores are detailed in Table 1. Although prior work suggests that RL can generalize better to OOD tasks than SFT (Huan et al., 2025; Chu et al., 2025), our findings highlight that RL remains susceptible to forgetting, with the direction and

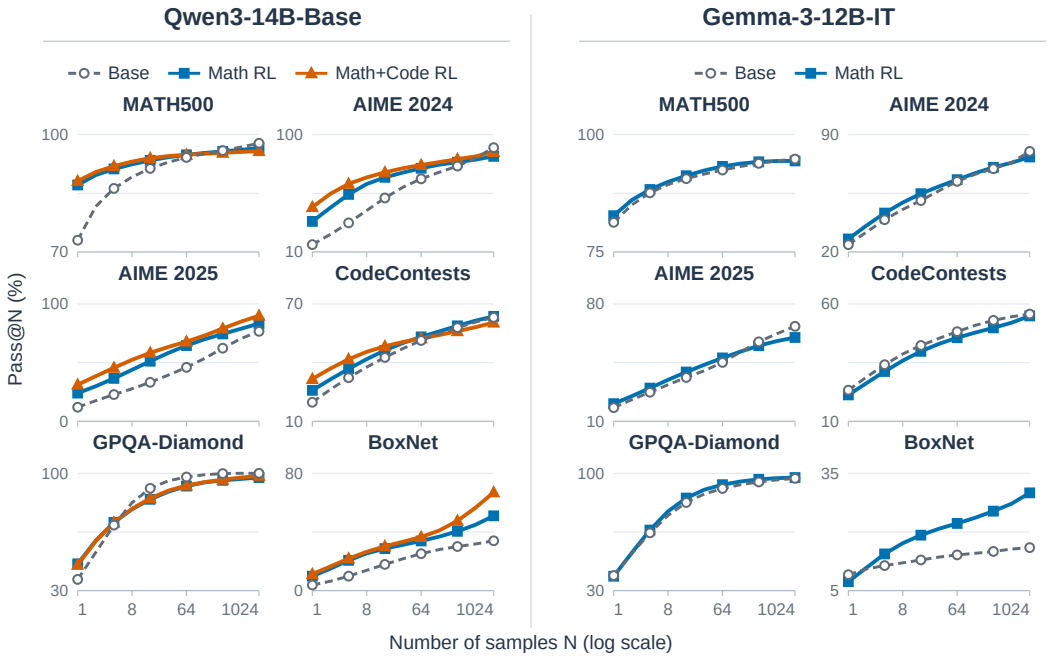


Figure 3: **RL can either expand or contract reasoning coverage at large sampling budgets.** Higher Pass@1 can accompany either lower or higher Pass@1024 relative to the base model. The direction of the coverage change varies across training mixtures, benchmarks, and base models.

magnitude of OOD performance changes varying substantially across settings (Chen et al., 2026a; Shenfeld et al., 2026b).

2.3 Expansion and Shrinkage of Reasoning Boundary

Task-level gains and losses show how training changes performance across domains. To examine how it changes the range of problems solved within each task, we compare finite-budget coverage using Pass@N (Figure 3). For a fixed prompt, Pass@N increases with Pass@1 rate; however, benchmark-level comparisons do not necessarily preserve the Pass@1 ordering at large N.

Reduced coverage at large sampling budgets. The most prevalent pattern we observe is higher Pass@1 alongside lower Pass@1024 relative to the base model, especially in mathematics. This agrees with the coverage contrast reported by Yue et al. (2025).

Broader coverage when base success is rare. On Reasoning Gym task BoxNet (Stojanovski et al., 2026) where base model success is exceptionally rare, Qwen3-Math-RL raises Pass@1 from 3.8% to 9.9% and Pass@1024 from 34.0% to 51.0%, with Gemma-3 exhibiting similar gains. These gains show that higher one-shot accuracy can accompany higher Pass@1024, consistent with Liu et al. (2025); Wen et al. (2026).

Our observations are broadly consistent with prior work (Liu et al., 2025): shifts in Pass@N are largely dictated by the benchmark distribution and the base model's initial ability, with substantial gains generally emerging where initial coverage is low. We further find that the training data itself can alter this regime: Qwen3-Math-RL and Qwen3-Math-Code-RL move CodeContests coverage in opposite directions. Consequently, whether RL expands or shrinks the reasoning boundary is context-dependent, rather than a universal RL effect or a trend that can be inferred from Pass@1.

These benchmark-level contrasts leave open how the solution process changes, motivating the trajectory analysis in Section 3.1.

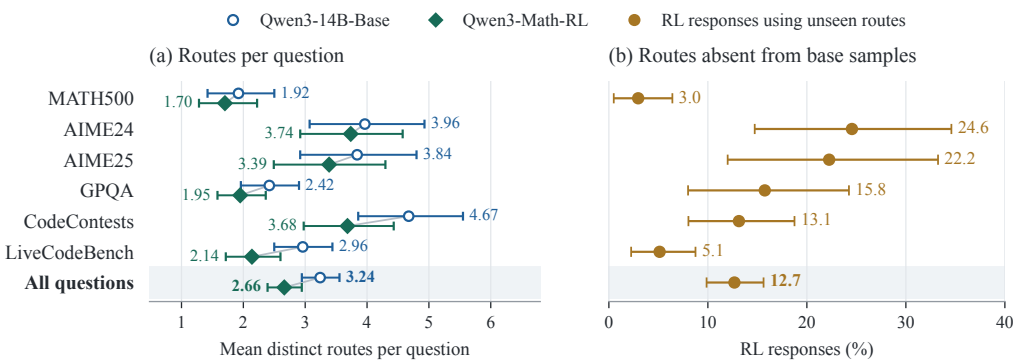


Figure 4: **RL uses fewer routes, but also routes absent from base samples.** (a) Mean distinct-route counts with matched numbers of identifiable responses per question. (b) Percentage of RL responses whose route is absent from the same question's 16 base responses.

3 From Empirical Observations to Reasoning Strategy Selection

To understand the performance and coverage changes in Sections 2.2 and 2.3, we compare solution trajectories before and after RL. The comparison indicates a shift in the strategies used to solve a question. We describe these strategies as *reasoning routes* and examine how their use differs between the base and RL models. The findings motivate a model that separates strategy selection from conditional answer generation (Section 3.2).

3.1 Reasoning-route analysis

3.1.1 Analysis setup

We compare Qwen3-14B-Base with Qwen3-Math-RL on 240 questions from six benchmarks, with 16 responses per model and question. A *reasoning route* is the strategy used to reach a response's adopted conclusion, when identifiable. Route identity reflects the problem representation and key mechanism, not notation, arithmetic, or final answers. For example, analytic reduction and exhaustive search are different routes. We pool and group both models' LLM-extracted strategy descriptions from reasoning traces within each question. We average distinct-route counts over equal-size samples of identifiable responses from both models, using the smaller available count for each question, and then equally over questions. Separately, we check which RL routes appear among the question's original 16 base responses. Appendix F.1 gives classification and statistical details.

3.1.2 Empirical findings

RL concentrates on fewer routes per question. Figure 4(a) shows lower mean route counts for Qwen3-Math-RL on all six benchmarks, most markedly in coding. Across questions, the mean falls from 3.24 for base to 2.66 for RL, with a paired RL minus base difference of -0.58 (95% confidence interval $[-0.78, -0.38]$). For example, a LiveCodeBench question asks for the minimum absolute difference between array elements at least x indices apart. Its 16 base responses use three routes, including binary-search neighbor comparison, which all 16 RL responses use.

RL also uses routes absent from the base samples. Of 3,840 Qwen3-Math-RL responses, 487 (12.7%) follow a route absent from the base responses to the same question (Figure 4(b)). For example, on a LiveCodeBench question finding pattern occurrences within distance k of another pattern's occurrence, all 16 base responses compare occurrence pairs directly, whereas 11 RL responses use binary search over occurrence lists.

3.2 A two-stage autoregressive model of chain-of-thought reasoning

For analysis, we assume a shared library of reasoning strategies and represent the CoT trajectory through a strategy z, followed by an answer y. Both z and y may correspond to sequences of tokens rather than single tokens, yielding an autoregressive factorization at the block level. This abstraction

separates strategy selection from execution, while the shared library provides a common space for studying transfer, coverage, and compute scaling in Sections 4 to 6.

Let $[M] = \{1, \dots, M\}$ be a finite library of reasoning strategies and let $\mathcal{Y}$ be a finite answer space. For any prompt $x \in \mathcal{X}$, we consider a two-stage policy that first generates a reasoning strategy $z \in [M]$ and then an answer $y \in \mathcal{Y}$ conditional on z. The joint policy is

$$\pi_{\theta,w}(z, y|x) = \pi_\theta(z|x)\pi_w(y|x, z), \tag{3.1}$$

where both stages are linear softmax policies,

$$\pi_\theta(z|x) = \frac{\exp(\theta_z^\top \phi(x))}{\sum_{k=1}^M \exp(\theta_k^\top \phi(x))}, \qquad \pi_w(y|x, z) = \frac{\exp(w_y^\top \phi(x, z))}{\sum_{v\in\mathcal{Y}} \exp(w_v^\top \phi(x, z))}. \tag{3.2}$$

Here $\theta_z, w_y \in \mathbb{R}^d$ are trainable parameters for strategy selection and answer generation, respectively. The feature maps $\phi(x)$ and $\phi(x, z)$ are frozen. For a given prompt x, let $\mathcal{M}(x) \subseteq [M]$ be its non-empty set of valid reasoning strategies and let $\mathcal{A}(x) \subseteq \mathcal{Y}$ be its non-empty correct-answer set. We define the conditional and end-to-end scores as

$$R_w(x, z) = \sum_{y\in\mathcal{A}(x)} \pi_w(y|x, z), \qquad R_{\theta,w}(x) = \sum_{z\in\mathcal{M}(x)} \pi_\theta(z|x)R_w(x, z). \tag{3.3}$$

3.3 Optimization dynamics

Let $\{x_1, \dots, x_n\}$ be the finite training set. For $\beta > 0$, define the KL-regularized training objective

$$J_{\text{strategy}}(\theta, w) = \frac{1}{n}\sum_{i=1}^n \left[R_{\theta,w}(x_i) - \beta D_{\text{KL}}(\pi_\theta(\cdot|x_i) \,\|\, \pi_{\theta_0}(\cdot|x_i))\right], \tag{3.4}$$

$$J_{\text{ans}}(\theta, w) = \frac{1}{n}\sum_{i=1}^n \left[R_{\theta,w}(x_i) - \beta\mathbb{E}_{\pi_\theta(\cdot|x_i)}\left[\mathbb{1}_{z\in\mathcal{M}(x_i)} D_{\text{KL}}\big(\pi_w(\cdot|x_i, z) \,\|\, \pi_{w_0}(\cdot|x_i, z)\big)\right]\right]. \tag{3.5}$$

Motivated by the small learning rates used in prominent RL pipelines (Yu et al., 2025; DeepSeek-AI, 2025), we analyze the policy gradient flow given by

$$\dot\theta_t = \nabla_\theta J_{\text{strategy}}(\theta_t, w_t), \qquad \dot w_t = \nabla_w J_{\text{ans}}(\theta_t, w_t), \qquad (\theta_t, w_t)\big|_{t=0} = (\theta_0, w_0). \tag{3.6}$$

Along the RL dynamics, define $R_t(x, z) := R_{w_t}(x, z), R_t(x) := R_{\theta_t,w_t}(x)$. Throughout our theoretical analysis, we consider a vanishing but strictly positive KL coefficient, $\beta \downarrow 0$. Without KL regularization, the policy collapses to the valid strategies with the largest initial probability.

4 Reasoning Transfer and Forgetting

We now use the two-stage model to analyze how RL-induced shifts in strategy selection transfer across tasks and influence target performance.

Assumption 1 (Realizability on training data)**.** *The training-prompt features are orthogonal:*

$$\langle\phi(x_i), \phi(x_j)\rangle = 0, \qquad \forall\, i, j \in [n],\ i \neq j. \tag{4.1}$$

Moreover, for every $y \in \mathcal{Y}$, there exists u_y such that

$$\phi(x_i, z)^\top u_y = \mathbb{1}_{y\in\mathcal{A}(x_i)} - |\mathcal{A}(x_i)|/|\mathcal{Y}|, \qquad \forall\, i \in [n], \quad z \in \mathcal{M}(x_i). \tag{4.2}$$

This is essentially an over-parameterized regime. We adopt this feature orthogonality Eq. (4.1) primarily for ease of exposition; our analysis can be readily extended linearly independent features. Furthermore, Eq. (4.2) ensures proper linear separation of training data within the answer space.

For a target prompt x', define its feature overlap with each training prompt by

$$c(x', x_i) = \langle\phi(x'), \phi(x_i)\rangle / \|\phi(x_i)\|^2. \tag{4.3}$$

Define the strategies with largest aggregate training support by

$$\mathcal{S}(x') = \arg\max_{z\in[M]} \sum_{i=1}^n c(x', x_i)\mathbb{1}_{z\in\mathcal{M}(x_i)}. \tag{4.4}$$

Let π_∞^β denote the convergent policy at fixed β and define vanishing KL limit $\pi_{\text{RL}} := \lim_{\beta\downarrow 0} \pi_\infty^\beta$.

Theorem 4.1 (Transferability of RL). *Under Assumption 1, for any finite t and $z \in [M]$,*

$$\pi_{\theta_t}(z|x') \propto \pi_{\theta_0}(z|x') \prod_{i=1}^{n} \left[\frac{\pi_{\theta_t}(z|x_i)}{\pi_{\theta_0}(z|x_i)} \right]^{c(x',x_i)}. \tag{4.5}$$

In the vanishing KL limit, the RL strategy policy is the base policy conditioned on $\mathcal{S}(x')$, and the probability of selecting a valid strategy is

$$\pi_{\mathrm{RL}}(\mathcal{M}(x')|x') = \frac{\pi_0(\mathcal{S}(x') \cap \mathcal{M}(x')|x')}{\pi_0(\mathcal{S}(x')|x')}. \tag{4.6}$$

Assumption 2 (Target answer-feature orthogonality). *The answer features of x' are orthogonal to source-valid answer features: $\langle \phi(x', z'), \phi(x_i, z) \rangle = 0, \forall\, z \in \mathcal{M}(x_i), z' \in [M]$.*

Corollary 4.2. *Under Assumptions 1 and 2 and vanishing KL limit, the reward on target prompt is*

$$R_{\pi_{\mathrm{RL}}}(x') = \frac{\sum_{z \in \mathcal{S}(x') \cap \mathcal{M}(x')} \pi_0(z|x') R_0(x', z)}{\pi_0(\mathcal{S}(x')|x')}. \tag{4.7}$$

Detailed proofs are deferred to Appendix B. Theorem 4.1 shows how changes in strategy preferences on training prompts transfer to target prompts through feature overlap. Under answer-feature orthogonality, Corollary 4.2 translates this strategy selection into end-to-end reward.

Our results show that the inductive bias of RL preserves the base model's preference among the surviving strategies (Shenfeld et al., 2026b). More crucially, it decouples this preservation from performance shifts. **Feature overlap governs which strategy preferences transfer, but cannot alone predict whether the target task benefits**. If $\mathcal{S}(x') \cap \mathcal{M}(x') = \emptyset$, all valid strategies are suppressed thereby inducing forgetting. Therefore, RL transferability is jointly determined by the reasoning strategies favored in the training data, the valid strategies required by the target task, and the base model's bias. The heterogeneous shifts in Figure 2 are compatible with this selective landscape and match the qualitative predictions. Further prompt-level analyses are given in Appendix F.

In Appendix B.3, we leverage the same two-stage model to investigate why SFT is more susceptible to forgetting than RL, and how its behavior depends on the characteristics of the teacher policy.

5 Expansion and Shrinkage of the Reasoning Boundary

We now examine how these prompt-level shifts in strategy preference aggregate across an evaluation benchmark $\mathcal{B}$ to alter overall coverage. Define the Pass@N metric as

$$\mathrm{Pass@}\,N(\pi) = \mathbb{E}_{x' \sim \mathcal{B}} \left[1 - \left(1 - R_\pi(x')\right)^N \right]. \tag{5.1}$$

To isolate strategy selection, we evaluate each strategy policy with an ideal answer decoder: a response succeeds exactly when its strategy is valid. Consequently, Theorem 4.1 gives:

$$R_{\pi_0}(x') = \pi_0(\mathcal{M}(x')|x'), \qquad R_{\pi_{\mathrm{RL}}}(x') = \frac{\pi_0(\mathcal{S}(x') \cap \mathcal{M}(x')|x')}{\pi_0(\mathcal{S}(x')|x')}. \tag{5.2}$$

Theorem 5.1. *Under Assumption 1, fix $\alpha \in (0, 1]$ and an integer $N > \log(4/\alpha)$.*

*(I) **Strategy loss (Base wins):** If*

$$\Pr_{x' \sim \mathcal{B}}[\mathcal{S}(x') \cap \mathcal{M}(x') = \emptyset] \geq \alpha, \quad \Pr_{x' \sim \mathcal{B}}\left[\pi_0(\mathcal{M}(x')|x') \leq \frac{\log(4/\alpha)}{N} \right] \leq \frac{\alpha}{4},$$

then

$$\mathrm{Pass@}\,N(\pi_0) - \mathrm{Pass@}\,N(\pi_{\mathrm{RL}}) > \frac{\alpha}{2}.$$

*(II) **Rare-strategy amplification (RL wins):** If*

$$\Pr_{x' \sim \mathcal{B}}\left[\pi_0(\mathcal{M}(x')|x') \leq \frac{1}{4N} \right] \geq \alpha, \quad \Pr_{x' \sim \mathcal{B}}\left[\frac{\pi_0(\mathcal{S}(x') \cap \mathcal{M}(x')|x')}{\pi_0(\mathcal{S}(x')|x')} \leq \frac{\log(4/\alpha)}{N} \right] \leq \frac{\alpha}{4},$$

then

$$\mathrm{Pass@}\,N(\pi_{\mathrm{RL}}) - \mathrm{Pass@}\,N(\pi_0) > \frac{\alpha}{4}.$$

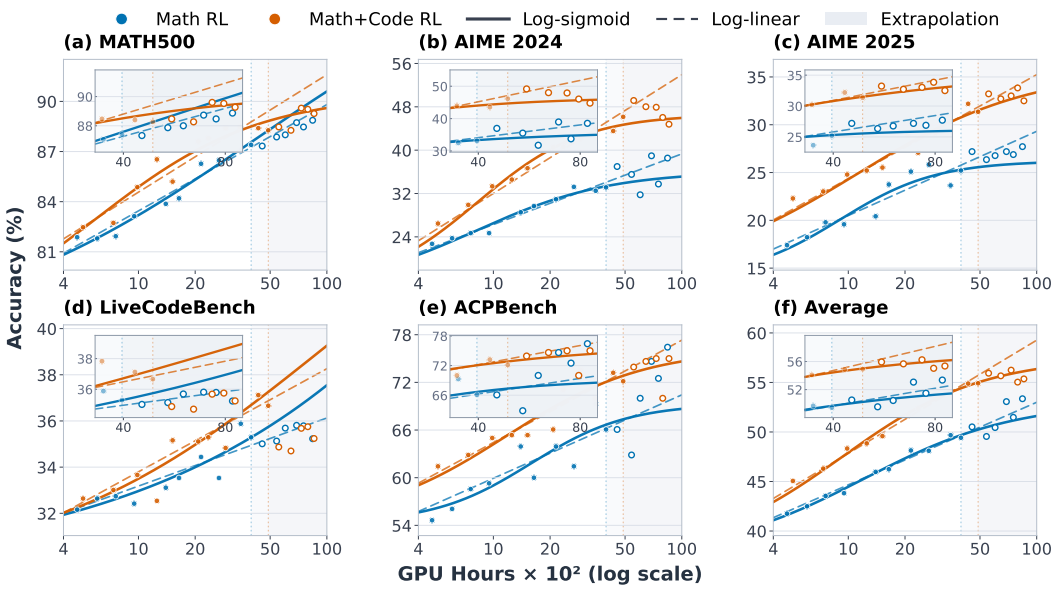


Figure 5: **Scaling behavior of RL compute.** Both laws capture the fitting regime effectively with $R^2 > 0.99$. Extrapolative predictions are highly reliable in the math domain and average scores, yet exhibit greater variation on coding and planning tasks due to noisy RL training dynamics.

Detailed proofs are provided in Appendix C. The theorem shows that Pass@N is governed by the low-success tail, particularly prompts whose one-shot success rate is of order $1/N$ (Chen et al., 2025), rather than by average Pass@1. Coverage decreases when RL suppresses strategies on such prompts and increases when it amplifies rare successes above this scale. Both directions arise from reweighting an existing strategy library, so broader sampled coverage need not imply new strategies.

Consistent with our transferability results, the reasoning boundary shifts depend jointly on the **training data**, **benchmark distribution** and **base model**. Figure 3 illustrates this interaction empirically: Qwen3-Math-RL expands coverage on BoxNet but shrinks it on AIME24 despite Pass@1 gains. Altering the training mixture with coding data completely inverts Qwen's performance on CodeContests, while the same Math-RL yields opposite AIME25 coverage shifts for Qwen and Gemma. More prompt-level analysis and evaluation caveats are in Appendix F.

6 Scaling Laws of RL Compute

Beyond transferability and reasoning boundary, recent studies highlight another critical frontier: how does performance scale with RL compute? Khatri et al. (2025) introduce a **log-sigmoid** law:

$$R(C) = R_0 + \frac{A - R_0}{1 + (C_{\text{mid}}/C)^B}. \tag{6.1}$$

However, the sigmoidal plateau typically emerges only during a late saturation stage, far beyond the reach of most compute-constrained RL training. Alternatively, several works (Meta AI, 2026; Tan et al., 2026; Shen et al., 2026) apply a first-order approximation to derive a **log-linear** law:

$$R(C) = R_0 + B_{\text{lin}} \log(C/C_{\text{ref}}). \tag{6.2}$$

In this section, we examine these empirical laws and demonstrate that our theoretical framework provides a rigorous foundation for deriving and explaining RL scaling behavior mechanistically.

6.1 Empirical scaling laws in RL

We fit Eqs. (6.1) and (6.2) to Qwen3-Math-RL and Qwen3-Math-Code-RL evaluations through step 200, using cumulative GPU-hours as compute. To test the prediction power of scaling laws, we extend both RL training runs to step 350, reaching near 2× compute budget (step 350). Figure 5

shows the observed scores and fitted curves in both regimes. Detailed fitting procedures, coefficient estimates, and further results using other compute metrics are given in Appendix E.4.

As shown in Figure 5, the log-sigmoid and log-linear fits are often nearly indistinguishable within the fitting regime. This is expected since the eq. (6.2) is a local approximation of eq. (6.1). The two laws diverge only at the extrapolation regime: the log-linear law maintains a constant improvement and is therefore unbounded, whereas the log-sigmoid law bends toward a finite plateau.

Over the evaluated extrapolation horizon, Qwen3-Math-RL adheres more closely to the log-linear prediction, yet Qwen3-Math-Code-RL exhibits saturation: the log-sigmoid curve better tracks the late-stage performance on MATH500, AIME and ACPBench, as well as the average scores. Meanwhile, the log-linear model increasingly overpredicts performance at large compute level.

The reliability of scaling law prediction also varies by evaluation domain. In-domain mathematical benchmarks consistently follow the prediction consistently, whereas extrapolation to coding and planning benchmarks is more brittle. Individual benchmark trajectories are also less stable than their average, reflecting the noisy dynamics of practical RL training. Notably, while Khatri et al. (2025); Tan et al. (2026) also report extrapolative scaling laws, their evaluations are restricted to held-out data drawn from the training distribution. In contrast, our analysis incorporates out-of-distribution and cross-domain evaluations. Our setting therefore tests not only whether performance scales predictably with RL compute, but also whether this scaling behavior transfers to new domains.

6.2 Theoretical justification for scaling laws

We now investigate why these simple laws capture RL dynamics via reasoning-strategy selection. For ease of exposition, we take $n = 1$, i.e., there is only one training prompt denoted by x. For a given evaluation benchmark $\mathcal{B}$, denote its reward during training by $R(t) := \mathbb{E}_{x' \sim \mathcal{B}} R_t(x')$.

Assumption 3 (Strategy-dominated regime)**.** *There exists T_0 such that $\pi_0(\cdot|x)$ and $\pi_{T_0}(\cdot|x)$ are uniform within $\mathcal{M}(x)$ and within $[M]\backslash\mathcal{M}(x)$. From T_0 onward, the answer policy saturates and remains frozen: $R_t(x, z) = R^{\beta}_{\infty}(x, z) = \rho_\beta$ for all $t \geq T_0, z \in \mathcal{M}(x)$.*

After T_0, training is essentially reduced to a tractable evolution of the probability of valid strategies. We use elapsed time $C = t - T_0$ as a proxy for compute and fit the scaling laws over an interval (a, b) of this accuracy. For any $u \in [a, b]$, let C_u be the first time such that $\pi_{T_0+C_u}(\mathcal{M}(x)|x) = u$. For any $x' \sim \mathcal{B}$, we assume feature overlap $c(x', x) > 0$ and positive transfer.

Theorem 6.1 (Informal)**.** *Under Assumptions 1 to 3, there exist $R_0, A, B, C_{\text{mid}}$ such that*

$$\sup_{C \in [C_a, C_b]} \left| R(T_0 + C) - R_0 - \frac{A - R_0}{1 + (C_{\text{mid}}/C)^B} \right| \leq \mathcal{O}((b - a)^2). \tag{6.3}$$

Theorem 6.2 (Informal)**.** *Under the same assumptions, there exist $R_0, B_{\text{lin}}, C_{\text{ref}}$ such that*

$$\sup_{C \in [C_a, C_b]} |R(T_0 + C) - R_0 - B_{\text{lin}} \log(C/C_{\text{ref}})| \leq \mathcal{O}((b - a)^2). \tag{6.4}$$

The results show that both log-sigmoid and log-linear laws closely approximate RL dynamics over a local compute range. Formal statements and error bounds are deferred to Theorems D.6 and D.7 in the appendix. Direct computation verifies that our error bounds remain below 1% for fitting windows spanning up to 30%, as visualized in Appendix D.5.

7 Conclusion and Discussion

In this paper, we investigate how RL reshapes LLM reasoning behaviors. Starting from empirical observations across various model families and task domains, we analyze solution traces before and after RL, revealing that the underlying mechanism is a shift in reasoning strategy. We propose a two-stage policy model to formalize this phenomenon and prove the conditions under which RL can transfer or forget capabilities, and expand or contract the reasoning boundary. We further leverage this model to theoretically justify empirical scaling laws for RL compute. These results connect changes in strategy selection to cross-domain transfer, reasoning coverage, and compute scaling.

AI USE STATEMENT

LLM-based agents assisted with experiment execution, data analysis, literature search, and drafting of this manuscript. The authors take full responsibility for all claims and artifacts.

ETHICS STATEMENT

We analyze open-weight models on synthetic and public text; there are no human subjects and no personal data. Dual-use risk is minimal.

REPRODUCIBILITY STATEMENT

The appendices describe the evaluation protocols, training configurations and pipeline for reasoning strategy identification.

REFERENCES

Fan Chen, Audrey Huang, Noah Golowich, Sadhika Malladi, Adam Block, Jordan T. Ash, Akshay Krishnamurthy, and Dylan J. Foster. The coverage principle: How pre-training enables post-training. *arXiv preprint arXiv:2510.15020*, 2025. URL https://arxiv.org/abs/2510.15020.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. In *International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2510.18874.

Xingwu Chen, Tianle Li, and Difan Zou. Reshaping reasoning in LLMs: A theoretical analysis of RL training dynamics through pattern selection. In *International Conference on Learning Representations*, volume 2026, pp. 15084–15113, 2026b. URL https://arxiv.org/abs/2506.04695.

Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, et al. Revisiting reinforcement learning for LLM reasoning from a cross-domain perspective. *arXiv preprint arXiv:2506.14965*, 2025. URL https://arxiv.org/abs/2506.14965.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025. URL https://arxiv.org/abs/2501.17161.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025. URL https://arxiv.org/abs/2502.01456.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/

2024/hash/0f69b4b96a46f284b726fbd70f74fb3b-Abstract-Datasets_and_Benchmarks_Track.html.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021. URL https://arxiv.org/abs/2103.03874.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general LLM capabilities? understanding transferability of LLM reasoning. *arXiv preprint arXiv:2507.00432*, 2025. URL https://arxiv.org/abs/2507.00432.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. WildTeaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/54024fca0cef9911be36319e622cde38-Abstract-Conference.html.

Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S. Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for LLMs. *arXiv preprint arXiv:2510.13786*, 2025. URL https://arxiv.org/abs/2510.13786.

Harsha Kokel, Michael Katz, Kavitha Srinivas, and Shirin Sohrabi. ACPBench: Reasoning about action, change, and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. URL https://arxiv.org/abs/2410.05669.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6449–6464, 2023. doi: 10.18653/v1/2023.emnlp-main.397. URL https://aclanthology.org/2023.emnlp-main.397/.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with AlphaCode. *arXiv preprint arXiv:2203.07814*, 2022. URL https://arxiv.org/abs/2203.07814.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025. URL https://arxiv.org/abs/2505.24864.

Meta AI. Introducing muse spark: Scaling towards personal superintelligence. Technical blog, April 2026. URL https://ai.meta.com/blog/introducing-muse-spark-msl/.

Youssef Mroueh. Reinforcement learning with verifiable rewards: GRPO's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025. URL https://arxiv.org/abs/2503.06639.

Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019. doi: 10.1162/tacl_a_00266. URL https://aclanthology.org/Q19-1016/.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Jingyan Shen, Ang Li, Salman Rahman, Yifan Sun, Micah Goldblum, Matus Telgarsky, and Pavel Izmailov. Understanding reasoning from pretraining to post-training. *arXiv preprint arXiv:2607.16097*, 2026. URL https://arxiv.org/abs/2607.16097.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026a. URL https://arxiv.org/abs/2601.19897.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2509.04259.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*, 2024. URL https://arxiv.org/abs/2409.19256.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. *Advances in Neural Information Processing Systems*, 38, 2026.

Joe Suk and Yaqi Duan. On the optimization dynamics of RLVR: Gradient gap and step size thresholds. *arXiv preprint arXiv:2510.08539*, 2025. URL https://arxiv.org/abs/2510.08539.

Yiyou Sun, Yuhan Cao, Pohao Huang, Haoyue Bai, Hannaneh Hajishirzi, Nouha Dziri, and Dawn Song. DELTA-Code: How does RL unlock and transfer new programming algorithms in LLMs? *arXiv preprint arXiv:2509.21016*, 2025a. URL https://arxiv.org/abs/2509.21016.

Yiyou Sun, Shawn Hu, Georgia Zhou, Ken Zheng, Hannaneh Hajishirzi, Nouha Dziri, and Dawn Song. OMEGA: Can LLMs reason outside the box in math? evaluating exploratory, compositional, and transformative generalization. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-1240. URL https://arxiv.org/abs/2506.18880.

Zelin Tan, Hejia Geng, Xiaohang Yu, Mulei Zhang, Guancheng Wan, Yifan Zhou, Qiang He, Xiangyuan Xue, Heng Zhou, Yutao Fan, Zhong-Zhi Li, Zaibin Zhang, Guibin Zhang, Chen Zhang, Zhenfei Yin, Philip Torr, and Lei Bai. Scaling behaviors of LLM reinforcement learning post-training: An empirical study in mathematical reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31300–31319, 2026. doi: 10.18653/v1/2026.acl-long.1444. URL https://aclanthology.org/2026.acl-long.1444/.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*, 2023.

David Vilares and Carlos Gómez-Rodríguez. HEAD-QA: A healthcare dataset for complex reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 960–966, 2019. doi: 10.18653/v1/P19-1092. URL https://aclanthology.org/P19-1092/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. URL https://arxiv.org/abs/2201.11903.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *International Conference on Learning Representations*, volume 2026, pp. 49450–49483, 2026. URL https://arxiv.org/abs/2506.14245.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. URL https://arxiv.org/abs/2504.13837.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3363–3369, 2019. doi: 10.18653/v1/D19-1332. URL https://aclanthology.org/D19-1332/.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

A Related Work

Reinforcement learning for LLM reasoning and chain of thought. CoT prompting and self-consistency establish the value of generating and sampling reasoning trajectories (Wei et al., 2022; Wang et al., 2023). DeepSeekMath introduces GRPO, and DeepSeek-R1 demonstrates large-scale reasoning gains from RL (Shao et al., 2024; DeepSeek-AI, 2025). These results motivate separating a strategy's availability, selection probability, and execution reliability, which our two-stage model treats explicitly.

Transferability, forgetting, and the reasoning boundary. Huan et al. (2025) find broader transfer and better retention than SFT in controlled math-tuning experiments, while Cheng et al. (2025) show that transfer varies with training domain and task difficulty. Chen et al. (2026a) identify on-policy data as a major factor in RL's relative resistance to forgetting. Coverage findings also differ: Yue et al. (2025) observe gains at small sampling budgets but reduced coverage at large ones, whereas Liu et al. (2025) report expansion under prolonged RL training. Wen et al. (2026) further show that evaluating reasoning correctness, rather than final answers alone, can change coverage comparisons. Our framework characterizes conditions for these contrasting outcomes within a shared model of strategy selection, distinguishing finite-budget coverage from the creation or loss of underlying strategies.

Scaling laws in reinforcement learning. Khatri et al. (2025) fit log-sigmoid curves relating held-out performance to RL training compute and validate extrapolation to longer runs. Shen et al. (2026) connect local log-linear RL-compute trends to pretraining. For a one-prompt linear-softmax policy-gradient flow in a weak-KL, strategy-dominated regime, we derive both curve forms as finite-window approximations and show how feature overlap transports the source exponent to target prompts. These results explain possible mechanisms for observed compute trends; they do not establish a global scaling law across model size, data, or test-time sampling.

Theoretical analyses of RL in LLMs. Chen et al. (2026b) analyze reasoning-pattern selection in a two-stage tabular policy with fixed conditional success rates. Complementary analyses study gradient-gap-dependent optimization stability (Suk & Duan, 2025), GRPO success amplification (Mroueh, 2025), and pretrained response coverage (Chen et al., 2025). Our framework learns both strategy and answer policies and uses shared prompt features to derive how source updates reweight target strategies. This transport mechanism connects strategy selection to transfer and forgetting, Pass@N coverage, and approximation laws over finite compute windows.

B Proofs for Transfer and Forgetting

B.1 Source dynamics underlying RL transfer

The following auxiliary result supplies the source limit used to prove Theorem 4.1. We retain its convergence rate for the scaling analysis.

Theorem B.1 (Source RL limit). *Under Assumption 1, fix $\beta > 0$. RL converges on every training prompt to a policy π^{β}_{∞}. For $i \in [n]$ and $z \in \mathcal{M}(x_i)$ and every $y \in \mathcal{Y}$,*

$$\pi^{\beta}_{\infty}(y|x_i, z) = \frac{\pi_0(y|x_i, z)e^{\mathbb{1}_{y\in\mathcal{A}(x_i)}/\beta}}{\sum_{v\in\mathcal{Y}} \pi_0(v|x_i, z)e^{\mathbb{1}_{v\in\mathcal{A}(x_i)}/\beta}}, \tag{B.1}$$

and hence

$$R^{\beta}_{\infty}(x_i, z) = \frac{R_0(x_i, z)}{R_0(x_i, z) + [1 - R_0(x_i, z)]e^{-1/\beta}}. \tag{B.2}$$

The limiting strategy policy is

$$\pi^{\beta}_{\infty}(z|x_i) \propto \pi_0(z|x_i) \begin{cases} e^{R^{\beta}_{\infty}(x_i,z)/\beta}, & z \in \mathcal{M}(x_i), \\ 1, & z \notin \mathcal{M}(x_i). \end{cases} \tag{B.3}$$

Moreover, there are constants $C_\beta, \lambda_\beta > 0$ such that

$$\max_i \left\|\pi_{\theta_t}(\cdot|x_i) - \pi^\beta_\infty(\cdot|x_i)\right\|_{\mathrm{TV}} \le C_\beta e^{-\lambda_\beta t},$$
$$\max_{\substack{i\in[n]\\ z\in\mathcal{M}(x_i)}} \left\|\pi_{w_t}(\cdot|x_i,z) - \pi^\beta_\infty(\cdot|x_i,z)\right\|_{\mathrm{TV}} \le C_\beta e^{-\lambda_\beta t}. \tag{B.4}$$

Proof. Assumption 1 yields parameters $(\theta^\beta_\star, w^\beta_\star)$ whose increments from (θ_0, w_0) lie in the respective source-feature spans and satisfy

$$\phi(x_i)^\top(\theta^\beta_{\star,z} - \theta_{0,z}) = \frac{1}{\beta}\left[\mathbb{1}_{z\in\mathcal{M}(x_i)} R^\beta_\infty(x_i,z) - \frac{1}{M}\sum_{k\in\mathcal{M}(x_i)} R^\beta_\infty(x_i,k)\right], \tag{B.5}$$

$$\phi(x_i,z)^\top(w^\beta_{\star,y} - w_{0,y}) = \frac{1}{\beta}\left[\mathbb{1}_{y\in\mathcal{A}(x_i)} - \frac{|\mathcal{A}(x_i)|}{|\mathcal{Y}|}\right]. \tag{B.6}$$

The first equation holds for all i, z and the second for all i, $z \in \mathcal{M}(x_i)$, and y. In the first equation, the indicator makes the strategy-level return zero when $z \notin \mathcal{M}(x_i)$. Pairwise logit differences in Eqs. (B.5) and (B.6) give exactly Eq. (B.1) and Eq. (B.3). Choose the minimum-norm parameter increments in these spans. Because the right-hand sides are centered across softmax classes, the increments sum to zero across classes, matching the gradient-flow gauge.

We prove convergence to these parameters on the source-feature update spans. Set

$$E_\theta(t) = \frac{1}{2}\sum_{z=1}^{M}\|\theta_{t,z} - \theta^\beta_{\star,z}\|^2, \qquad E_w(t) = \frac{1}{2}\sum_{y\in\mathcal{Y}}\|w_{t,y} - w^\beta_{\star,y}\|^2. \tag{B.7}$$

For a distribution P and vector a, write $\mathrm{Var}_P(a) = \sum_j P(j)(a_j - \sum_k P(k)a_k)^2$. Direct differentiation of Eq. (3.4) and Eq. (3.5), using the target log-odds above, yields

$$\dot{E}_w = -\frac{\beta}{n}\sum_{i=1}^{n}\sum_{z\in\mathcal{M}(x_i)} \pi_{\theta_t}(z|x_i)\,\mathrm{Var}_{\pi_{w_t}(\cdot|x_i,z)}\left(\phi(x_i,z)^\top(w_t - w^\beta_\star)\right), \tag{B.8}$$

$$\dot{E}_\theta = -\frac{\beta}{n}\sum_{i=1}^{n}\mathrm{Var}_{\pi_{\theta_t}(\cdot|x_i)}\left(\phi(x_i)^\top(\theta_t - \theta^\beta_\star)\right) + \frac{1}{n}\sum_{i=1}^{n}\mathrm{Cov}_{\pi_{\theta_t}(\cdot|x_i)}\left(\phi(x_i)^\top(\theta_t - \theta^\beta_\star), \mathbb{1}_{z\in\mathcal{M}(x_i)}[R_t(x_i,z) - R^\beta_\infty(x_i,z)]\right). \tag{B.9}$$

Here a vector-valued expression inside a variance or covariance is indexed by the displayed strategy or answer variable.

Eq. (B.8) makes E_w nonincreasing, so all source answer probabilities have a common lower bound $\underline{q} > 0$. For fixed i and $z \in \mathcal{M}(x_i)$, let $a_y = \phi(x_i,z)^\top(w_{t,y} - w^\beta_{\star,y})$ and interpolate from the current to the limiting answer policy by $q_s(y) \propto \pi_{w_t}(y|x_i,z)e^{-sa_y}$, $0 \le s \le 1$. Then $q_1 = \pi^\beta_\infty(\cdot|x_i,z)$ and $\frac{d}{ds}q_s(\mathcal{A}(x_i)) = -\mathrm{Cov}_{q_s}(\mathbb{1}_{y\in\mathcal{A}(x_i)}, a_y)$. Taking the q_0-mean of a as a candidate center and using $q_s(y) \le 1 \le q_0(y)/\underline{q}$ gives $\mathrm{Var}_{q_s}(a) \le \underline{q}^{-1}\mathrm{Var}_{q_0}(a)$. Since $\mathrm{Var}_{q_s}(\mathbb{1}_{y\in\mathcal{A}(x_i)}) \le 1/4$, integration gives

$$|R_t(x_i,z) - R^\beta_\infty(x_i,z)|^2 \le \frac{1}{4\underline{q}}\mathrm{Var}_{\pi_{w_t}(\cdot|x_i,z)}(a). \tag{B.10}$$

Cauchy–Schwarz and Young's inequality now absorb the covariance in Eq. (B.9) into the two variance terms, giving constants $\gamma, \eta > 0$ such that

$$\frac{d}{dt}(E_\theta + \gamma E_w) \le -\frac{\eta}{n}\sum_{i=1}^{n}\left[\mathrm{Var}_{\pi_{\theta_t}(\cdot|x_i)}\left(\phi(x_i)^\top(\theta_t - \theta^\beta_\star)\right) + \sum_{z\in\mathcal{M}(x_i)}\pi_{\theta_t}(z|x_i)\,\mathrm{Var}_{\pi_{w_t}(\cdot|x_i,z)}\left(\phi(x_i,z)^\top(w_t - w^\beta_\star)\right)\right]. \tag{B.11}$$

Thus both parameter blocks stay bounded, which also gives a uniform positive floor $\underline{p} > 0$ for every source gate probability. The parameter errors remain in the source-feature spans and sum to zero across softmax classes. Consequently, each gate variance in Eq. (B.11) is at least $\underline{p}$ times the sum of squared gate-logit errors over strategies; each weighted answer variance is at least $\underline{pq}$ times the sum of squared answer-logit errors over answers. Orthogonality of the prompt features and positivity of the answer-feature Gram operator on its span make these sums, after summing over source prompts and valid strategies, bound E_θ and E_w, respectively. Thus the right-hand side of Eq. (B.11) is at most $-2\lambda_\beta(E_\theta + \gamma E_w)$ for some $\lambda_\beta > 0$. Gronwall's inequality and smoothness of softmax then give global existence and Eq. (B.4).

Summing Eq. (B.1) over $\mathcal{A}(x_i)$ gives Eq. (B.2). Uniformly over the finite training set, $1 - R^\beta_\infty(x_i, z) = O(e^{-1/\beta}) = o(\beta)$. This uniform estimate is used in the target limit below. □

B.2 Proofs for RL transfer

At fixed $\beta > 0$, the target gate converges to

$$\pi^\beta_\infty(z|x') \propto \pi_0(z|x') \exp\left\{ \frac{1}{\beta} \sum_{i:\, z \in \mathcal{M}(x_i)} c(x', x_i) R^\beta_\infty(x_i, z) \right\}, \tag{B.12}$$

and, after enlarging the constants in Eq. (B.4) if needed,

$$\left\| \pi_{\theta_t}(\cdot|x') - \pi^\beta_\infty(\cdot|x') \right\|_{\mathrm{TV}} \le \left(\sum_{i=1}^n |c(x', x_i)| \right) C_\beta e^{-\lambda_\beta t}. \tag{B.13}$$

If every $c(x', x_i)$ is zero, the target gate remains at initialization for all time.

Proof of Theorem 4.1. For each strategy z, the RL dynamics keep $\theta_{t,z} - \theta_{0,z}$ in $\mathrm{span}\{\phi(x_1), \dots, \phi(x_n)\}$. Orthogonality and Eq. (4.3) give

$$\phi(x')^\top (\theta_{t,z} - \theta_{0,z}) = \sum_{i=1}^n c(x', x_i) \phi(x_i)^\top (\theta_{t,z} - \theta_{0,z}). \tag{B.14}$$

Subtracting Eq. (B.14) for two strategies converts it into an identity between their log-odds changes. Exponentiating and normalizing over z proves Eq. (4.5). Letting $t \to \infty$ and using Eq. (B.3) proves Eq. (B.12).

The probability floors established above and Eq. (B.4) bound every source log-odds error by $C_\beta e^{-\lambda_\beta t}$ after enlarging C_β. Applying Eq. (B.14) to the finite-time and limiting parameters, summing absolute overlaps, and using softmax Lipschitzness gives Eq. (B.13). If every overlap vanishes, Eq. (B.14) is zero for all t, proving invariance.

Finally, Eq. (B.2) implies, uniformly over strategies,

$$\sum_{i:\, z \in \mathcal{M}(x_i)} c(x', x_i) R^\beta_\infty(x_i, z) = \sum_{i=1}^n c(x', x_i) \mathbb{1}_{z \in \mathcal{M}(x_i)} + o(\beta). \tag{B.15}$$

Substitution into Eq. (B.12) makes all mass outside $\mathcal{S}(x')$ vanish and preserves base-policy ratios within $\mathcal{S}(x')$. Thus

$$\pi_{\mathrm{RL}}(z|x') = \frac{\pi_0(z|x') \mathbb{1}_{z \in \mathcal{S}(x')}}{\pi_0(\mathcal{S}(x')|x')}. \tag{B.16}$$

Summing over $\mathcal{M}(x')$ proves Eq. (4.6). □

Proof of Corollary 4.2. Every answer-parameter increment lies in the span of $\phi(x_i, z)$ for $i \in [n]$ and $z \in \mathcal{M}(x_i)$. Assumption 2 therefore fixes every target answer logit. Combining this fact with Eq. (B.13) proves fixed-β reward convergence at the same rate, and substitution of Eq. (B.16) gives Eq. (4.7). □

B.3 When does SFT exhibit catastrophic forgetting

To compare with RL, we analyze SFT under the same theoretical setup. For each training prompt x_i, we sample an expert demonstration $z_{E,i} \sim \pi_E(\cdot|x_i), y_{E,i} \sim \pi_E(\cdot|x_i, z_{E,i})$, conditioning on $z_{E,i} \in \mathcal{M}(x_i)$ and $y_{E,i} \in \mathcal{A}(x_i)$. The training objective is

$$J_{\mathrm{SFT}}(\theta, w) = \frac{1}{n} \sum_{i=1}^{n} \left[\log \pi_\theta(z_{E,i}|x_i) + \log \pi_w(y_{E,i}|x_i, z_{E,i})\right]. \tag{B.17}$$

Consider the gradient flow from (θ_0, w_0) on Eq. (B.17) and let π_{SFT} be the convergent policy. As in the RL analysis, define the strategies with largest support from the realized demonstrations:

$$\mathcal{S}_E(x') = \arg\max_{z \in [M]} \sum_{i=1}^{n} c(x', x_i) \mathbb{1}_{z=z_{E,i}}. \tag{B.18}$$

Theorem B.2 (Transferability of SFT). *Under Assumption 1, for any realized expert traces, the SFT strategy policy converges to*

$$\pi_{\mathrm{SFT}}(z|x') = \frac{\pi_0(z|x')\mathbb{1}_{z \in \mathcal{S}_E(x')} \prod_{i=1}^{n} \left[\frac{\|\phi(x_i)\|^{2\mathbb{1}_{z=z_{E,i}}}}{\pi_0(z|x_i)}\right]^{c(x',x_i)}}{\sum_{k \in \mathcal{S}_E(x')} \pi_0(k|x') \prod_{i=1}^{n} \left[\frac{\|\phi(x_i)\|^{2\mathbb{1}_{k=z_{E,i}}}}{\pi_0(k|x_i)}\right]^{c(x',x_i)}}. \tag{B.19}$$

In particular, its support is exactly $\mathcal{S}_E(x')$:

$$\pi_{\mathrm{SFT}}(z|x') > 0 \quad \Longleftrightarrow \quad z \in \mathcal{S}_E(x'). \tag{B.20}$$

The probability of valid strategies tends to one if $\mathcal{S}_E(x') \subseteq \mathcal{M}(x')$ and to zero if $\mathcal{S}_E(x') \cap \mathcal{M}(x') = \varnothing$.

Proof of Theorem B.2. The strategy and answer gradients are bounded, so the flow exists for all time. The case $M = 1$ is immediate. For $M \geq 2$, write $a_{i,z}(t) = \phi(x_i)^\top \theta_{t,z}^{\mathrm{SFT}}$. Source-feature orthogonality makes the strategy dynamics independent across prompts:

$$\dot{a}_{i,z} = \frac{\|\phi(x_i)\|^2}{n} \left[\mathbb{1}_{z=z_{E,i}} - \pi_{\theta_t^{\mathrm{SFT}}}(z|x_i)\right]. \tag{B.21}$$

For every $k \neq z_{E,i}$,

$$\frac{d}{dt} e^{-a_{i,k}} = \frac{\|\phi(x_i)\|^2/n}{\sum_z e^{a_{i,z}}}. \tag{B.22}$$

Thus $e^{-a_{i,k}(t)} = e^{-a_{i,k}(0)} + u_i(t)$ for the same increasing u_i at every non-demonstrated strategy. The sum $\sum_z a_{i,z}(t)$ is constant, and $u_i(t) \to \infty$; otherwise all logits would stay bounded and $\dot{u}_i$ would stay bounded away from zero. Substituting the preceding expression into the conserved sum gives

$$u_i(t)^M \sim \frac{M\|\phi(x_i)\|^2 t}{n \exp(\sum_z a_{i,z}(0))}. \tag{B.23}$$

Hence, for every z,

$$a_{i,z}(t) - a_{i,z}(0) = \left(\mathbb{1}_{z=z_{E,i}} - \frac{1}{M}\right) \log \frac{M\|\phi(x_i)\|^2 t}{n} + \frac{1}{M} \sum_k a_{i,k}(0) - a_{i,z}(0) + o(1). \tag{B.24}$$

Each strategy-policy parameter increment lies in the source prompt-feature span, so source-feature orthogonality gives Eq. (B.14) with θ_t replaced by θ_t^{SFT}. Substituting Eq. (B.24) into this relation, exponentiating, normalizing, and dropping factors independent of z yields, uniformly over the finite strategy set,

$$\pi_{\theta_t^{\mathrm{SFT}}}(z|x') \propto \pi_0(z|x') \prod_{i=1}^{n} \pi_0(z|x_i)^{-c(x',x_i)} \prod_{i:\, z_{E,i}=z} \left(\frac{M\|\phi(x_i)\|^2 t}{n}\right)^{c(x',x_i)} [1 + o(1)]. \tag{B.25}$$

The power of t is maximized exactly on $\mathcal{S}_E(x')$. On this set, the powers of Mt/n are equal and cancel under normalization. Removing this common factor from Eq. (B.25) proves Eq. (B.19). All remaining factors are positive, so the limiting policy is supported exactly on $\mathcal{S}_E(x')$, proving Eq. (B.20). The two claims about probability mass on target-valid strategies follow immediately from this support characterization. □

The definition of supporting strategies in Eqs. (B.18) and (4.4) exposes a fundamental distinction: RL amplifies all source-valid strategies, whereas SFT reinforces only the demonstrated strategies. Consequently, **RL policy broadly preserves base model's prior, whereas SFT behavior depends heavily on the characteristics of the teacher policy**, and thus is more prone to catastrophic forgetting, as observed in Figure 2. However, this vulnerability is not absolute; Qwen3-SFT, for instance, outperforms the RL models on the OOD GPQA-Diamond.

More broadly, transferability of SFT is highly sensitive to compatibility between the student and the teacher policy. Since the SFT data originates from Qwen3-32B, they naturally align better with Qwen3-14B-Base than with Gemma-3-12B-IT, accounting for the substantially broader forgetting of Gemma-3-SFT in Figure 2. Further examples and analysis are provided in Appendix F.

C Proof for the Pass@N Reasoning Boundary

Proof of Theorem 5.1. Under the strategy-success evaluation, $R_\pi(x') = \pi(\mathcal{M}(x')|x')$, and Theorem 4.1 gives

$$R_{\pi_0}(x') = \pi_0(\mathcal{M}(x')|x'), \qquad R_{\pi_{\mathrm{RL}}}(x') = \frac{\pi_0(\mathcal{S}(x') \cap \mathcal{M}(x')|x')}{\pi_0(\mathcal{S}(x')|x')}.$$

The base softmax has full support, so $R_{\pi_{\mathrm{RL}}}(x') = 0$ exactly when $\mathcal{S}(x') \cap \mathcal{M}(x') = \emptyset$.

For any random variable $U \in [0,1]$ and $0 \le a < b < 1$,

$$\Pr[U \le a](1-a)^N \le \mathbb{E}[(1-U)^N] \le \Pr[U \le b] + (1-b)^N.$$

The lower bound restricts the expectation to $\{U \le a\}$; the upper bound splits it over $\{U \le b\}$ and its complement. The theorem's choice of N ensures $0 < 1/(4N) < \log(4/\alpha)/N < 1$, and

$$\left(1 - \frac{1}{4N}\right)^N \ge \frac{3}{4}, \qquad \left(1 - \frac{\log(4/\alpha)}{N}\right)^N < \frac{\alpha}{4}.$$

For part (I), apply the lower bound to $R_{\pi_{\mathrm{RL}}}(x')$ with $a = 0$ and the upper bound to $R_{\pi_0}(x')$ with $b = \log(4/\alpha)/N$. This gives

$$1 - \mathrm{Pass@}\,N(\pi_{\mathrm{RL}}) \ge \alpha, \qquad 1 - \mathrm{Pass@}\,N(\pi_0) < \frac{\alpha}{2}.$$

Subtracting these failure probabilities proves the claimed base-model gap.

For part (II), apply the lower bound to $R_{\pi_0}(x')$ with $a = 1/(4N)$ and the upper bound to $R_{\pi_{\mathrm{RL}}}(x')$ with $b = \log(4/\alpha)/N$. This gives

$$1 - \mathrm{Pass@}\,N(\pi_0) \ge \frac{3\alpha}{4}, \qquad 1 - \mathrm{Pass@}\,N(\pi_{\mathrm{RL}}) < \frac{\alpha}{2}.$$

Their difference proves the claimed RL gap. Finally, the endpoint success probabilities converge to $R_{\pi_{\mathrm{RL}}}(x')$ as $\beta \downarrow 0$. Bounded convergence gives convergence of Pass@N under $\mathcal{B}$, so each strict bound persists for sufficiently small $\beta > 0$. □

D Proofs for the Scaling Laws

Throughout this appendix, we take $n = 1$, let x be the training prompt, set $m := |\mathcal{M}(x)| < M$, fix $\beta > 0$ within each training run, and write $C = t - T_0 \ge 0$. For a target prompt x', write $c(x', x) = \langle \phi(x'), \phi(x) \rangle / \|\phi(x)\|^2$. Write $\mathrm{logit}(u) = \log(u/(1-u))$, $\sigma(s) = (1 + e^{-s})^{-1}$, and abbreviate

$$\pi_x(C) := \pi_{T_0+C}(\mathcal{M}(x)|x), \qquad \pi_{x'}(C) := \pi_{T_0+C}(\mathcal{M}(x)|x').$$

Thus $\pi_x(0)$ is the probability assigned to $\mathcal{M}(x)$ at T_0, whereas π_0 denotes the fixed KL reference. Under Assumptions 1 and 3, there is a common ρ_β such that

$$R_{T_0+C}(x,z) = \rho_\beta, \qquad C \geq 0, \quad z \in \mathcal{M}(x).$$

By Eq. (B.2), $1-\rho_\beta = O(e^{-1/\beta})$. We use $\pi_x(\infty)$ and $\pi_{x'}(\infty)$ for their fixed-β limits. For a target prompt, we normalize its reward only when $R^\beta_\infty(x') > R_{T_0}(x')$. Throughout the vanishing-KL analysis below, $p := \pi_x(0)$ and $0 < p < a < b < 1$ are fixed. The term $o(1)$ is uniform on $[a,b]$ as $\beta \downarrow 0$. The remainder in the target log-linear approximation is uniform over $c(x',x) > 0$ and $\pi_{x'}(0) \in (0,1)$.

For $0 < a < b < 1$, set

$$H(u) := \frac{2u-1}{u(1-u)} + 2\mathsf{logit}(u).$$

Here $H'(u) = [u(1-u)]^{-2}$, and

$$\Delta^{(p)}_{a,b} := \log \frac{H(b)-H(p)}{H(a)-H(p)}. \tag{D.1}$$

D.1 Scalar training-prompt dynamics

Lemma D.1 (Scalar training dynamics and transfer)**.** *Under Assumptions 1 and 3, the training-strategy probability satisfies*

$$\frac{d\pi_x}{dC} = \|\phi(x)\|^2 \left(\frac{1}{m} + \frac{1}{M-m}\right) [\pi_x(C)(1-\pi_x(C))]^2 \times \{\rho_\beta - \beta\,[\mathsf{logit}(\pi_x(C)) - \mathsf{logit}\,(\pi_0(\mathcal{M}(x)|x))]\}\,. \tag{D.2}$$

Moreover,

$$\mathsf{logit}(\pi_x(\infty)) = \mathsf{logit}\,(\pi_0(\mathcal{M}(x)|x)) + \frac{\rho_\beta}{\beta}. \tag{D.3}$$

For every target prompt and $C \geq 0$,

$$\mathsf{logit}(\pi_{x'}(C)) - \mathsf{logit}(\pi_{x'}(0)) = c(x',x)[\mathsf{logit}(\pi_x(C)) - \mathsf{logit}(\pi_x(0))]. \tag{D.4}$$

Consequently,

$$\mathsf{logit}(\pi_{x'}(\infty)) = \mathsf{logit}(\pi_{x'}(0)) + c(x',x)\,[\mathsf{logit}(\pi_x(\infty)) - \mathsf{logit}(\pi_x(0))]\,. \tag{D.5}$$

If Assumption 2 also holds, $c(x',x) > 0$, and $R^\beta_\infty(x') > R_{T_0}(x')$, then

$$\frac{R_{T_0+C}(x') - R_{T_0}(x')}{R^\beta_\infty(x') - R_{T_0}(x')} = \frac{\pi_{x'}(C) - \pi_{x'}(0)}{\pi_{x'}(\infty) - \pi_{x'}(0)}. \tag{D.6}$$

Proof. For $\pi_z(C) = \pi_{T_0+C}(z|x)$, define

$$U_z(C) = \mathbb{1}_{z\in\mathcal{M}(x)}\rho_\beta - \beta \log \frac{\pi_z(C)}{\pi_0(z|x)}.$$

Differentiating the strategy objective gives

$$\frac{d}{dC}\left[\theta^\top_{T_0+C,z}\phi(x)\right] = \|\phi(x)\|^2 \pi_z(C)\left[U_z(C) - \sum_{k=1}^{M} \pi_k(C)U_k(C)\right]. \tag{D.7}$$

At $C = 0$, both the KL reference and the policy at T_0 are uniform within the valid and invalid classes. By Assumption 3, valid-strategy rewards are also symmetric, while invalid strategies have zero return. Uniqueness of the ODE solution therefore preserves both within-class symmetries. Substituting

$$\pi_z(C) = \frac{\pi_x(C)}{m} \quad (z \in \mathcal{M}(x)), \qquad \pi_z(C) = \frac{1-\pi_x(C)}{M-m} \quad (z \notin \mathcal{M}(x)),$$

into Eq. (D.7) and differentiating the softmax yields Eq. (D.2).

Using the common conditional reward ρ_β in Eq. (B.3) gives

$$\pi_\infty^\beta(z|x) \propto \pi_0(z|x) \exp\left(\frac{\mathbb{1}_{z\in\mathcal{M}(x)}\rho_\beta}{\beta}\right).$$

Taking the odds ratio between the valid and invalid classes proves Eq. (D.3).

Because $n = 1$, every strategy-parameter update is parallel to $\phi(x)$. Its logit change at x' is therefore $c(x', x)$ times its logit change at x. Taking the odds ratio between the valid and invalid classes proves Eq. (D.4). Letting $C \to \infty$ and applying Eq. (D.3) gives Eq. (D.5).

Finally, Assumption 2 fixes every target answer probability. Within $\mathcal{M}(x)$ and its complement, the target strategy proportions remain unchanged. Thus $R_{T_0+C}(x')$ depends linearly on $\pi_{x'}(C)$, up to an additive constant. Subtracting its value at $C = 0$ and normalizing by its total change at $C = \infty$ proves Eq. (D.6). □

Lemma D.2 (Relating compute and reward in the strategy-dominated regime)**.** *Fix $0 < p < a < b < 1$ and consider a family of runs satisfying Assumptions 1 and 3, with the reference policy and p, a, b fixed as $\beta \downarrow 0$. For sufficiently small β, $R_\infty^\beta(x) > R_{T_0}(x)$ and $\pi_x(C)$ increases through $[a, b]$. For $u \in [a, b]$, let C_u be the first value of C for which $\pi_x(C) = u$. Uniformly for $u \in [a, b]$,*

$$\frac{\log(C_u/C_a)}{\log(C_b/C_a)} = \frac{\log\{[H(u) - H(p)]/[H(a) - H(p)]\}}{\Delta_{a,b}^{(p)}} + o(1), \tag{D.8}$$

$$\frac{R_{T_0+C_u}(x) - R_{T_0}(x)}{R_\infty^\beta(x) - R_{T_0}(x)} = \frac{u - p}{1 - p} + o(1). \tag{D.9}$$

If, in addition, a target prompt x' satisfies Assumption 2, with $c = c(x', x) > 0$ and $R_\infty^\beta(x') > R_{T_0}(x')$, then the following identity is exact:

$$\frac{R_{T_0+C_u}(x') - R_{T_0}(x')}{R_\infty^\beta(x') - R_{T_0}(x')} = \frac{\sigma(\mathsf{logit}(\pi_{x'}(0)) + c[\mathsf{logit}(u) - \mathsf{logit}(p)]) - \pi_{x'}(0)}{\pi_{x'}(\infty) - \pi_{x'}(0)}. \tag{D.10}$$

For fixed c and $\pi_{x'}(0) \in (0, 1)$, the denominator on the right-hand side equals $1 - \pi_{x'}(0) + o(1)$.

Proof. Since $\rho_\beta \to 1$ and the reference strategy mass is fixed, uniformly for $p \le v \le b$,

$$\rho_\beta - \beta\,[\mathsf{logit}(v) - \mathsf{logit}\,(\pi_0(\mathcal{M}(x)|x))] = \rho_\beta[1 + o(1)].$$

Hence the expression in braces in Eq. (D.2) is positive on $[p, b]$ for sufficiently small β. On this interval, $d\pi_x/dC$ is bounded below by a positive multiple of $[p(1 - b)]^2$, so $\pi_x(C)$ reaches b in finite time. Separating variables, integrating, and using $H'(v) = [v(1 - v)]^{-2}$ yields, uniformly for $u \in [a, b]$,

$$\|\phi(x)\|^2 \left(\frac{1}{m} + \frac{1}{M - m}\right) \rho_\beta C_u = [H(u) - H(p)][1 + o(1)]. \tag{D.11}$$

In particular, $\log(C_b/C_a) = \Delta_{a,b}^{(p)} + o(1)$. Since $\Delta_{a,b}^{(p)} > 0$, taking logarithms and dividing by this span proves Eq. (D.8). By Eq. (D.3), $\pi_x(\infty) \to 1$.

Since $R_{T_0+C}(x) = \rho_\beta \pi_x(C)$ exactly,

$$\frac{R_{T_0+C_u}(x) - R_{T_0}(x)}{R_\infty^\beta(x) - R_{T_0}(x)} = \frac{u - p}{\pi_x(\infty) - p} = \frac{u - p}{1 - p} + o(1).$$

This also shows $R_\infty^\beta(x) > R_{T_0}(x)$ and proves Eq. (D.9).

Combining the transfer identity Eq. (D.4) with the normalized reward identity Eq. (D.6) yields Eq. (D.10). Finally, Eq. (D.5) and fixed $c, \pi_{x'}(0)$ imply $\pi_{x'}(\infty) \to 1$. □

D.2 Log-linear approximation

Lemma D.3 (Linear approximation of an inverse map)**.** *Let $g \in C^2([a, b])$ satisfy $g' > 0$ and $|g''/g'| \le K$. There exists $\alpha \in \mathbb{R}$ such that*

$$\sup_{u\in[a,b]} \left| u - \alpha - \frac{b - a}{g(b) - g(a)}[g(u) - g(a)] \right| \le \frac{K(b - a)^2}{16}. \tag{D.12}$$

Proof. Normalize the interval and the function by

$$F(t) = \frac{g(a + (b-a)t) - g(a)}{g(b) - g(a)}, \qquad f(t) = F'(t), \qquad d = K(b-a).$$

Then $f > 0$, $\int_0^1 f(t)\,dt = 1$, and $\log f$ is d-Lipschitz. If $d = 0$, then $f \equiv 1$ and the claim is exact. Hence assume $d > 0$. Let f^* be the decreasing rearrangement of f. For any two values $\lambda_1 < \lambda_2$ in the range of $\log f$, continuity and the d-Lipschitz property give

$$|\{t \in [0,1] : \lambda_1 < \log f(t) < \lambda_2\}| \geq \frac{\lambda_2 - \lambda_1}{d}.$$

Here $|\cdot|$ denotes Lebesgue measure. To pass from one level to the other, $\log f$ must remain between them over an interval of at least the displayed length. Rearrangement preserves these measures, so $\log f^*$ is also d-Lipschitz. Hence $f^*(t)/e^{-dt}$ is nondecreasing. Integrating this ratio on $[0, s]$ and $[s, 1]$ gives

$$\int_0^s f^*(t)\,dt \leq \frac{1 - e^{-ds}}{1 - e^{-d}}.$$

Therefore

$$\begin{aligned}\int_0^1 (f(t) - 1)_+\,dt &= \max_{0\leq s\leq 1}\left\{\int_0^s f^*(t)\,dt - s\right\} \\ &\leq \max_{0\leq s\leq 1}\left\{\frac{1 - e^{-ds}}{1 - e^{-d}} - s\right\} \leq \frac{d}{8}. \end{aligned} \tag{D.13}$$

For the last inequality, a direct calculation shows that d times the preceding maximum equals

$$\max_{0\leq v\leq 1}\{\log(1 - v + ve^d) - vd\}.$$

For fixed v, this expression and its first derivative with respect to d vanish at $d = 0$, while its second derivative with respect to d is at most $1/4$. Integrating this bound twice yields $d^2/8$.

Since $F(t) - t = \int_0^t (f(s) - 1)\,ds$, its oscillation is at most the right-hand side of Eq. (D.13). Returning to $[a, b]$, the residual in Eq. (D.12) therefore has oscillation at most $K(b-a)^2/8$. Centering its range proves the result. □

Lemma D.4 (Log-linear law for the training prompt)**.** *Under Assumptions 1 and 3, for any $\varepsilon_{\text{fit}} > 0$ and sufficiently small β, there exist $R_{0,x} < R_\infty^\beta(x)$ and $B_{\text{lin},x}, C_{\text{ref},x} > 0$ such that, for every $C \in [C_a, C_b]$,*

$$\frac{|R_{T_0+C}(x) - R_{0,x} - B_{\text{lin},x}\log(C/C_{\text{ref},x})|}{R_\infty^\beta(x) - R_{0,x}} \leq \frac{b-a}{1-p}\left[\frac{b-a}{8}\max_{v\in\{a,b\}}\frac{|1-2v|}{v(1-v)} + \frac{\Delta_{a,b}^{(p)}}{16}\right] + \varepsilon_{\text{fit}}. \tag{D.14}$$

Proof. Choose the fitted intercept $R_{0,x} := R_{T_0}(x)$. Since

$$H'(u) = \frac{1}{[u(1-u)]^2}, \qquad \left|\frac{H''(u)}{H'(u)}\right| = \frac{2|1-2u|}{u(1-u)},$$

Lemma D.3 gives $\alpha \in \mathbb{R}$ such that

$$\sup_{u\in[a,b]}\left|u - \alpha - \frac{b-a}{H(b) - H(a)}[H(u) - H(a)]\right| \leq \frac{(b-a)^2}{8}\max_{u\in\{a,b\}}\frac{|1-2u|}{u(1-u)}. \tag{D.15}$$

Let $F(u) = \log(C_u/C_a)/\log(C_b/C_a)$ and $G(u) = \log\{[H(u) - H(p)]/[H(a) - H(p)]\}/\Delta_{a,b}^{(p)}$. By Eq. (D.8), $F = G + o(1)$ uniformly. The identity

$$\frac{H(u) - H(a)}{H(b) - H(a)} = \frac{e^{\Delta_{a,b}^{(p)}G(u)} - 1}{e^{\Delta_{a,b}^{(p)}} - 1}$$

and Lemma D.3, applied to $e^{\Delta_{a,b}^{(p)} s}$ on $[0,1]$, show that, after adding a suitable constant, $F(u)$ approximates this normalized change in H with error at most $\Delta_{a,b}^{(p)}/16 + o(1)$. Combining this bound with Eq. (D.15), adjusting the constant term, and using Eq. (D.9) gives

$$\inf_{\alpha' \in \mathbb{R}} \sup_{u \in [a,b]} \left| \frac{R_{T_0+C_u}(x) - R_{0,x}}{R_\infty^\beta(x) - R_{0,x}} - \alpha' - \frac{b-a}{1-p} F(u) \right| \le \frac{b-a}{1-p} \left[\frac{b-a}{8} \max_{v \in \{a,b\}} \frac{|1-2v|}{v(1-v)} + \frac{\Delta_{a,b}^{(p)}}{16} \right] + o(1).$$

Choose

$$B_{\mathrm{lin},x} = \frac{[R_\infty^\beta(x) - R_{0,x}](b-a)}{(1-p)\log(C_b/C_a)} > 0,$$

and choose $C_{\mathrm{ref},x} > 0$ so that $-B_{\mathrm{lin},x} \log C_{\mathrm{ref},x}$ equals the resulting constant term. Because $\pi_x(C)$ is strictly increasing, C_u covers $[C_a, C_b]$ as u ranges over $[a,b]$. For sufficiently small β, the normalized remainder is at most $\varepsilon_{\mathrm{fit}}$. Multiplying the preceding bound by $R_\infty^\beta(x) - R_{0,x}$ gives

$$\sup_{C \in [C_a, C_b]} \left| R_{T_0+C}(x) - R_{0,x} - B_{\mathrm{lin},x} \log \frac{C}{C_{\mathrm{ref},x}} \right| \le [R_\infty^\beta(x) - R_{0,x}] \left\{ \frac{b-a}{1-p} \left[\frac{b-a}{8} \max_{v \in \{a,b\}} \frac{|1-2v|}{v(1-v)} + \frac{\Delta_{a,b}^{(p)}}{16} \right] + \varepsilon_{\mathrm{fit}} \right\}.$$

This proves the lemma. □

Lemma D.5 (Log-linear law for a target prompt)**.** *Under Assumptions 1 to 3, let x' satisfy $c = c(x', x) > 0$ and $R_{T_0+C_b}(x') > R_{T_0+C_a}(x')$. For any $\varepsilon_{\mathrm{fit}} > 0$ and sufficiently small β, there exist $B_{\mathrm{lin},x'}, C_{\mathrm{ref},x'} > 0$ such that*

$$\sup_{C \in [C_a, C_b]} \frac{|R_{T_0+C}(x') - R_{T_0}(x') - B_{\mathrm{lin},x'} \log(C/C_{\mathrm{ref},x'})|}{R_{T_0+C_b}(x') - R_{T_0+C_a}(x')} \le \frac{(1+c) \log \frac{b(1-a)}{a(1-b)} + \Delta_{a,b}^{(p)}}{16} + \varepsilon_{\mathrm{fit}}. \tag{D.16}$$

Proof. Write $Q(u) = \pi_{x'}(C_u)$. The exact transfer identity gives $\mathrm{logit}(Q(u)) = \mathrm{logit}(Q(a)) + c[\mathrm{logit}(u) - \mathrm{logit}(a)]$. As a function of $s = \mathrm{logit}(u)$,

$$\frac{dQ}{ds} = cQ(1-Q), \quad \left| \frac{d}{ds} \log \frac{dQ}{ds} \right| \le c, \quad \frac{dH}{ds} = \frac{1}{u(1-u)}, \quad \left| \frac{d}{ds} \log \frac{dH}{ds} \right| \le 1.$$

Applying Lemma D.3 first to $Q(\sigma(s))$ and then to $H(\sigma(s))$, and combining the two approximations, gives a constant $\alpha^\star$ such that

$$\sup_{u \in [a,b]} \left| Q(u) - \alpha^\star - (Q(b) - Q(a)) \frac{H(u) - H(a)}{H(b) - H(a)} \right| \le \frac{[Q(b) - Q(a)](1+c)}{16} \log \frac{b(1-a)}{a(1-b)}.$$

Replacing the normalized change in H by $F(u) = \log(C_u/C_a)/\log(C_b/C_a)$, as shown in the proof of Lemma D.4, adds at most $(Q(b) - Q(a))[\Delta_{a,b}^{(p)}/16 + o(1)]$. By Eq. (D.10), multiplying by $[R_{T_0+C_b}(x') - R_{T_0+C_a}(x')]/[Q(b) - Q(a)]$ converts the result exactly into a bound for the unnormalized target reward, with no additional error term. Choose

$$B_{\mathrm{lin},x'} = \frac{R_{T_0+C_b}(x') - R_{T_0+C_a}(x')}{\log(C_b/C_a)} > 0$$

and choose $C_{\mathrm{ref},x'} > 0$ so that $-B_{\mathrm{lin},x'} \log C_{\mathrm{ref},x'}$ equals the resulting constant term. The error from replacing the normalized change in H by $F(u)$ is independent of c and $\pi_{x'}(0)$. Hence, for sufficiently small β, this error is at most $\varepsilon_{\mathrm{fit}}$ for every target. Thus

$$\sup_{C \in [C_a, C_b]} \left| R_{T_0+C}(x') - R_{T_0}(x') - B_{\mathrm{lin},x'} \log \frac{C}{C_{\mathrm{ref},x'}} \right| \le [R_{T_0+C_b}(x') - R_{T_0+C_a}(x')] \left\{ \frac{1+c}{16} \log \frac{b(1-a)}{a(1-b)} + \frac{\Delta_{a,b}^{(p)}}{16} + \varepsilon_{\mathrm{fit}} \right\}.$$

Dividing by the positive reward gain $R_{T_0+C_b}(x') - R_{T_0+C_a}(x')$ proves the lemma. □

D.3 BENCHMARK LOG-SIGMOID APPROXIMATION

We prove the benchmark log-sigmoid result of Section 6. Let $\mathcal{B}$ be a benchmark distribution over target prompts, and write $R(t) := \mathbb{E}_{x'\sim\mathcal{B}}[R_t(x')]$. The fitted lower level R_0 below need not equal the base-model score $R(0)$ or the observed score $R(T_0)$ before the fitting window. The fitted upper level A need not equal the fixed-β limiting reward.

For $x' \sim \mathcal{B}$, let

$$r_+(x') := \mathbb{E}_{\pi_{T_0}(\cdot|x')}[R_{T_0}(x', z) \mid z \in \mathcal{M}(x)],$$
$$r_-(x') := \mathbb{E}_{\pi_{T_0}(\cdot|x')}[R_{T_0}(x', z) \mid z \notin \mathcal{M}(x)],$$
$$d(x') := r_+(x') - r_-(x').$$

The conditional expectations are well defined because the softmax policy has full support and $0 < |\mathcal{M}(x)| < M$. When $c(x', x) > 0$, the mass on $\mathcal{M}(x)$ increases across the fitting interval, so the positive reward gain assumed in the main text is equivalent to $d(x') > 0$. For $0 < p < a < b < 1$, set

$$\delta_{a,b} := \mathsf{logit}(b) - \mathsf{logit}(a) = \log\frac{b(1-a)}{a(1-b)}, \qquad \Gamma^{(p)}_{a,b} := \frac{1}{2}\inf_{\lambda>0} \mathrm{osc}_{u\in[a,b]}\left\{\mathsf{logit}(u) - \lambda\log[H(u) - H(p)]\right\}, \tag{D.17}$$

where $p = \pi_{T_0}(\mathcal{M}(x)|x)$ and H is defined above. Thus $\delta_{a,b}$ is the training-strategy log-odds span, and $\Gamma^{(p)}_{a,b}$ measures the best approximation of those log odds by a linear function of log compute. Set $R_0 := \mathbb{E}_{x'\sim\mathcal{B}}[r_-(x')]$, $D := \mathbb{E}_{x'\sim\mathcal{B}}[d(x')]$, and $A := R_0 + D$.

Theorem D.6 (Benchmark log-sigmoid law with capped feature overlap)**.** *Assume Assumptions 1 and 3 with one training prompt and benchmark distribution $\mathcal{B}$. Suppose Assumption 2 holds for $x' \sim \mathcal{B}$. Consider the vanishing-β family of Lemma D.2. Suppose $c(x', x), d(x') > 0$ for $x' \sim \mathcal{B}$, and let $c_{\max} > 0$ be a finite upper bound on $c(x', x)$ for these prompts. For any $0 < c_0 \le c_{\max}$ and $\varepsilon_{\mathrm{fit}} > 0$, sufficiently small β admits $B, C_{\mathrm{mid}} > 0$ such that*

$$\sup_{C\in[C_a,C_b]}\left|\frac{R(T_0+C) - R_0}{A - R_0} - \frac{1}{1 + (C_{\mathrm{mid}}/C)^B}\right| \le \tanh\left(\frac{c_0^2\delta_{a,b}^2}{256} + \frac{c_0\Gamma^{(p)}_{a,b}}{4}\right) + \frac{\mathbb{E}_{x'\sim\mathcal{B}}\left[d(x')\tanh\left(\delta_{a,b}(c(x',x) - c_0)_+/8\right)\right]}{\mathbb{E}_{x'\sim\mathcal{B}}[d(x')]} + \varepsilon_{\mathrm{fit}}, \tag{D.18}$$

where $(q)_+ := \max\{q, 0\}$. For $c_0 = c_{\max}$ and fixed $p < a < 1$, the corresponding raw-reward error is $O((b-a)^2)$ as $b - a \to 0$, provided β is sufficiently small.

Proof of Theorem D.6. Fix $0 < c_0 \le c_{\max}$. Let $z(C) = \mathsf{logit}(\pi_{T_0+C}(\mathcal{M}(x)|x)) - \mathsf{logit}(p)$. By Lemma D.1, the target mass of $\mathcal{M}(x)$ is

$$\pi_{T_0+C}(\mathcal{M}(x)|x') = \sigma\left(\mathsf{logit}(\pi_{T_0}(\mathcal{M}(x)|x')) + c(x', x)z(C)\right).$$

Target answer-feature orthogonality fixes each conditional reward at its base value, independently of t and β. By Eq. (4.5) and training-strategy symmetry, the target's strategy proportions within $\mathcal{M}(x)$ and its complement equal their base values at T_0 and remain fixed thereafter. Thus $r_\pm(x')$ and $d(x')$ do not vary with β. Under the theorem's assumptions, $D > 0$, and

$$R(T_0 + C) = R_0 + DF(z(C)), \qquad F(z) := \frac{\mathbb{E}_{x'\sim\mathcal{B}}\left[d(x')\sigma\left(\mathsf{logit}(\pi_{T_0}(\mathcal{M}(x)|x')) + c(x', x)z\right]\right.}{D}.$$

In particular, $A - R_0 = D > 0$. Put $z_{\mathrm{mid}} := [z(C_a) + z(C_b)]/2$ and define the capped mixture

$$\widetilde{F}(z) := \frac{\mathbb{E}_{x'\sim\mathcal{B}}\left[d(x')\sigma\left(\mathsf{logit}(\pi_{T_0}(\mathcal{M}(x)|x')) + (c(x', x) - c_0)_+ z_{\mathrm{mid}} + \min\{c(x', x), c_0\}z\right)\right]}{D}.$$

The original and capped logits agree at $z = z_{\mathrm{mid}}$. Since $|z - z_{\mathrm{mid}}| \le \delta_{a,b}/2$ for $z \in [z(C_a), z(C_b)]$ and $|\sigma(s) - \sigma(t)| \le \tanh(|s-t|/4)$,

$$\sup_{z\in[z(C_a),z(C_b)]}|F(z) - \widetilde{F}(z)| \le \frac{\mathbb{E}_{x'\sim\mathcal{B}}\left[d(x')\tanh\left(\frac{\delta_{a,b}(c(x',x)-c_0)_+}{8}\right)\right]}{D}. \tag{D.19}$$

We next bound the curvature of the capped mixture. Write $L(z) := \mathsf{logit}(\widetilde{F}(z))$. Any positive mixture of logistic functions with slopes in $(0, c_0]$ satisfies

$$0 < L'(z) \le c_0, \qquad |L''(z)| \le \frac{c_0^2}{4}. \tag{D.20}$$

The derivatives of the integrand up to order two are uniformly bounded on compact z-intervals because $0 < d(x') \le 1$ and $0 < c(x', x) \le c_{\max}$. Dominated convergence therefore permits differentiation under the benchmark expectation. For completeness, let $t = e^{c_0 z}$. Each component $q_{x'}(t)$ of $\widetilde{F}$ has odds proportional to $t^{s_{x'}}$ for some $s_{x'} \in (0, 1]$, so $q_{x'}/(1 - q_{x'})$ is concave in t. Hence $(1 - q_{x'})\partial_{tt} q_{x'} + 2(\partial_t q_{x'})^2 \le 0$. Taking the d-weighted average, Cauchy–Schwarz gives

$$\left(\frac{\mathbb{E}_{x'\sim\mathcal{B}}[d(x')\partial_t q_{x'}]}{D}\right)^2 \le (1 - \widetilde{F})\frac{\mathbb{E}_{x'\sim\mathcal{B}}[d(x')(\partial_t q_{x'})^2/(1 - q_{x'})]}{D}.$$

Consequently,

$$(1 - \widetilde{F})\partial_{tt}\widetilde{F} + 2(\partial_t \widetilde{F})^2 \le 0,$$

so $\widetilde{F}/(1 - \widetilde{F})$ is concave in t. The same argument applied to the complementary components in t^{-1} shows that $(1 - \widetilde{F})/\widetilde{F}$ is concave in t^{-1}. Also, $\partial_z \widetilde{F} = \mathbb{E}_{x'\sim\mathcal{B}}[d(x')s_{x'}c_0 q_{x'}(1 - q_{x'})]/D$, and $\mathbb{E}_{x'\sim\mathcal{B}}[d(x')q_{x'}(1-q_{x'})]/D \le \widetilde{F}(1-\widetilde{F})$; thus $0 < L' \le c_0$. Writing $h = L'/c_0 \in (0, 1]$, concavity of the odds in t yields $L''/c_0^2 \le h(1 - h)$, while concavity of the complementary odds in t^{-1} yields $-L''/c_0^2 \le h(1 - h)$. This proves Eq. (D.20).

Let ℓ interpolate L at $z_{\mathrm{mid}} - \delta_{a,b}/(2\sqrt{2})$ and $z_{\mathrm{mid}} + \delta_{a,b}/(2\sqrt{2})$. The interpolation remainder and Eq. (D.20) give

$$\sup_{z\in[z(C_a),z(C_b)]} |L(z) - \ell(z)| \le \frac{c_0^2\delta_{a,b}^2}{64}, \qquad 0 < \ell' \le c_0.$$

By the definition of $\Gamma_{a,b}^{(p)}$, for every $\eta > 0$ there exist $\lambda > 0$ and $\gamma \in \mathbb{R}$ such that

$$\sup_{u\in[a,b]} |\mathsf{logit}(u) - \mathsf{logit}(p) - \gamma - \lambda \log[H(u) - H(p)]| \le \Gamma_{a,b}^{(p)} + \eta.$$

The compute–strategy relation in Lemma D.2 gives $\log(C_u/C_a) = \log\{[H(u) - H(p)]/[H(a) - H(p)]\} + o(1)$ uniformly on $[a, b]$. Consequently, $\ell\big(\gamma + \lambda \log[H(a) - H(p)] + \lambda \log(C/C_a)\big)$ is affine in $\log C$ with positive slope and approximates $L(z(C))$ within $c_0^2\delta_{a,b}^2/64 + c_0\Gamma_{a,b}^{(p)} + c_0\eta + o(1)$. Applying σ gives a log-sigmoid fit to $\widetilde{F}(z(C))$ with error at most $\tanh(c_0^2\delta_{a,b}^2/256 + c_0\Gamma_{a,b}^{(p)}/4 + c_0\eta/4) + o(1)$.

The normalized benchmark reward in Eq. (D.18) is exactly $F(z(C))$. Combine this with Eq. (D.19) and let $\eta \downarrow 0$; the remaining uniform $o(1)$ is absorbed into $\varepsilon_{\mathrm{fit}}$. For the short-window claim, $\delta_{a,b} = O(b-a)$. In the definition of $\Gamma_{a,b}^{(p)}$, choose λ so that $\mathsf{logit}(u) - \lambda \log[H(u) - H(p)]$ has zero derivative at a; Taylor's theorem then gives $\Gamma_{a,b}^{(p)} = O((b - a)^2)$. Taking $c_0 = c_{\max}$ and $\varepsilon_{\mathrm{fit}} = (b - a)^2$ makes the normalized error $O((b - a)^2)$. Since $A - R_0 \le 1$, so is the raw error. □

Proof of Theorem 6.1. Take $c_0 = c_{\max}$ in Theorem D.6; the second term vanishes. This gives the theorem directly when $R_0 > 0$. If $R_0 = 0$, apply that bound with $\varepsilon_{\mathrm{fit}}/2$ and choose a positive fitted lower level $\eta < A$ instead. Since $R(T_0 + C) \le A$ on the window,

$$\left|\frac{R(T_0 + C) - \eta}{A - \eta} - \frac{R(T_0 + C)}{A}\right| \le \frac{\eta}{A - \eta},$$

so a sufficiently small η absorbs the difference into the remaining $\varepsilon_{\mathrm{fit}}/2$. For the short-window rate, take $\eta = O((b - a)^2)$. □

D.4 Benchmark log-linear approximation

The single-target bound also applies to the benchmark average. Its feature overlap is weighted by each prompt's reward gain over the fitting interval:

$$\bar{c} := \frac{\mathbb{E}_{x'\sim\mathcal{B}}\left[c(x',x)\big(R_{T_0+C_b}(x') - R_{T_0+C_a}(x')\big)\right]}{R(T_0+C_b) - R(T_0+C_a)}. \tag{D.21}$$

The log-linear theorem assumes $\bar{c} < \infty$; unlike the log-sigmoid theorem, it does not require a uniform bound on feature overlap. The gains in these weights are determined by $d(x')$, $\pi_{T_0}(\mathcal{M}(x)|x')$, and $c(x',x)$ through the exact transfer identity above.

Theorem D.7 (Benchmark log-linear law)**.** *Assume Assumptions 1 to 3 with one training prompt and $0 < p < a < b < 1$. Suppose $c(x',x), d(x') > 0$ for $x' \sim \mathcal{B}$ and $\bar{c} < \infty$. For any $\varepsilon_{\text{fit}} > 0$ and sufficiently small β, there exist $R_0, B_{\text{lin}}, C_{\text{ref}} > 0$ such that*

$$\sup_{C\in[C_a,C_b]} \frac{|R(T_0+C) - R_0 - B_{\text{lin}}\log(C/C_{\text{ref}})|}{R(T_0+C_b) - R(T_0+C_a)} \le \frac{(1+\bar{c})\delta_{a,b} + \Delta^{(p)}_{a,b}}{16} + \varepsilon_{\text{fit}}. \tag{D.22}$$

If $c(x',x) \le c_{\max} < \infty$ on the benchmark, the raw-reward error is $O((b-a)^2)$ for fixed $p < a < 1$ as $b - a \to 0$, provided β is sufficiently small.

Proof of Theorem D.7. Apply Lemma D.5 for $x' \sim \mathcal{B}$. Its compute–strategy approximation error is uniform in $c(x',x)$ and the initial target strategy mass, so one sufficiently small β makes the promptwise bounds hold for draws from the benchmark distribution. The fitted slope is

$$B_{\text{lin},x'} = \frac{R_{T_0+C_b}(x') - R_{T_0+C_a}(x')}{\log(C_b/C_a)} > 0.$$

For this slope, choose the fitted constant term as the midpoint of the range of $R_{T_0+C}(x') - B_{\text{lin},x'}\log C$ on $[C_a, C_b]$. This choice minimizes the uniform promptwise error. Because rewards lie in $[0,1]$ and C_a, C_b are fixed, these slopes and constant terms are bounded and may be averaged. Their expectation is a log-linear fit with positive slope $B_{\text{lin}} = \mathbb{E}_{x'\sim\mathcal{B}}[B_{\text{lin},x'}]$. For any $R_0 > 0$, choose $C_{\text{ref}} > 0$ so that the averaged constant term equals $R_0 - B_{\text{lin}}\log C_{\text{ref}}$. The triangle inequality bounds its error by the expectation of the promptwise errors. Each is proportional to the positive gain $R_{T_0+C_b}(x') - R_{T_0+C_a}(x')$. Dividing by the benchmark gain and using Eq. (D.21) proves the displayed bound. For the short-window claim, $\bar{c} \le c_{\max}$, while $\delta_{a,b}$ and $\Delta^{(p)}_{a,b}$ are $O(b-a)$. The exact transfer identity gives

$$\frac{d}{du}\pi_{x'}(C_u) = \frac{c(x',x)\pi_{x'}(C_u)(1-\pi_{x'}(C_u))}{u(1-u)},$$

so the benchmark reward gain is $O(b-a)$ because $c(x',x) \le c_{\max}$. Taking $\varepsilon_{\text{fit}} = b - a$ proves the raw-reward claim. □

D.5 Visualizing the approximation bounds

We visualize the the magnitude of approximation error bound in Theorems 6.1 and 6.2. Specifically, we consider a single target with unit reward contrast and obtain its reward gain over each fitting window. The results are shown in Figure 6.

Neither functional form is uniformly favored. For $c = 0.5$, the displayed log-sigmoid bound is smaller throughout the plotted range. For $c = 1$, it remains smaller over most windows, but the log-linear certificate becomes smaller when both a and $b - a$ are large; for example, at $(a, b-a) = (0.65, 0.30)$ the two bounds are approximately 4.90% and 1.75%, respectively.

E Experiment Details and Further Results

E.1 Training setup

RL. We run GRPO in verl with a learning rate of 10^{-6} for 200 steps. Each step samples 512 prompts and 16 responses per prompt, followed by one PPO epoch with a minibatch size of 128

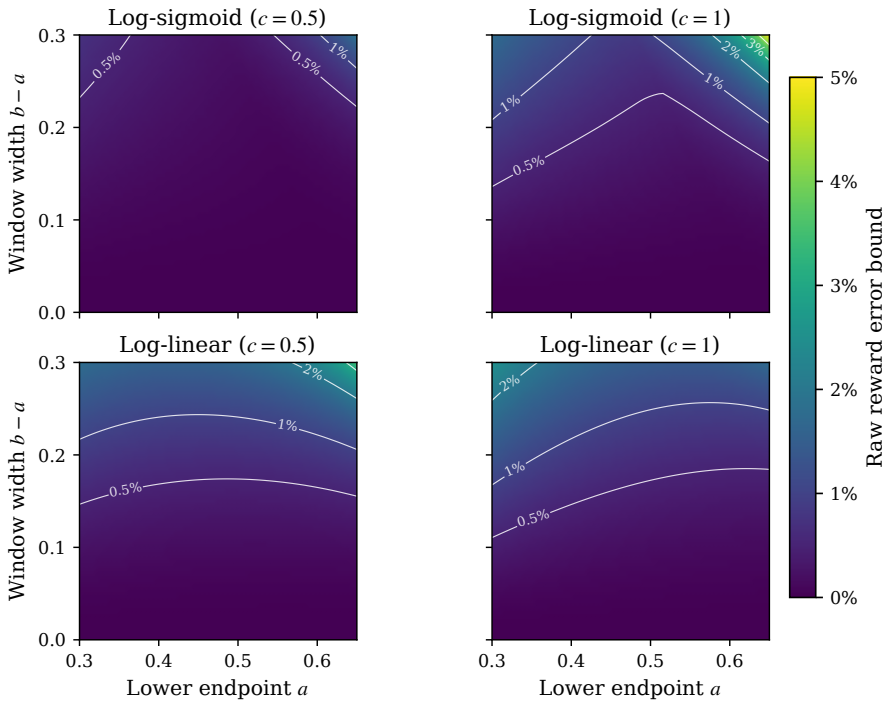


Figure 6: **The theoretical approximation bounds are a few percentage points in a representative regime.** We use $p = 0.1$, initial target-strategy mass $q_0 = 0.5$, unit reward contrast, and feature overlap $c \in \{0.5, 1\}$. We vary $a \in [0.30, 0.65]$ and $b - a \in [0, 0.30]$, with the zero-width boundary shown by continuity. All panels use the same raw-reward color scale, and contours report percentage-point error. The asymptotic term ε_{fit} is omitted.

prompts. Training rollouts use temperature 1, top-p 1, a 2,048-token prompt limit, and a 16,384-token response limit. The lower and upper clipping parameters are 0.2 and 0.28. Mathematics rewards indicate final-answer correctness, while coding rewards measure the fraction of ten tests passed.

SFT. We use batch size 512, learning rate of 5×10^{-5}, a two-epoch cosine schedule with 10% warmup, and a 16,384-token sequence limit. We evaluate the 1.5-epoch checkpoint (step 117).

E.2 Transfer and Forgetting

Benchmarks. We evaluate transfer and forgetting across four domains: (1) **Math**: MATH500 (Hendrycks et al., 2021), AIME24, AIME25, and OMEGA (Sun et al., 2025b); (2) **Coding**: LiveCodeBench (Jain et al., 2024), CodeContests (Li et al., 2022), and DELTA (Sun et al., 2025a); (3) **Knowledge and Planning**: GPQA-Diamond (Rein et al., 2023), HEAD-QA (Vilares & Gómez-Rodríguez, 2019), ACPBench (Kokel et al., 2025), ToolUse (Tang et al., 2023), and Science (Feng et al., 2024); (4) **General Capabilities**: CoQA (Reddy et al., 2019), IFEval (Zhou et al., 2023), HaluEval (Li et al., 2023), MC-TACO (Zhou et al., 2019), WildJailbreak (Jiang et al., 2024), and WildGuardTest (Han et al., 2024).

Evaluation protocol. We use native chat templates. Sampled reasoning tasks use temperature 0.6, top-p 0.95, top-k 20, and a maximum generation length of 32,768, with three responses per problem for MATH500, GPQA-Diamond, 10 responses for LiveCodeBench, 32 for AIME, and one for DELTA. CodeContests uses Eurus-2-RL's 377-problem validation subset. Greedy decoding caps are 32,768 for ACPBench, CoQA, and HaluEval, and 4,096 for ToolUse, Science, and safety. HEAD-QA and MC-TACO use likelihood scoring; IFEval uses one response. We retain native task

metrics, including strict instruction-level IFEval accuracy, CoQA F1, DELTA test pass rates, and WildGuard-based refusal accuracy.

Complete scores. Table 1 reports the absolute scores underlying Figure 2.

Table 1: **Mathematics gains accompany transfer to some tasks and forgetting on others.** Scores (%; higher is better) for the initializations, thinking-enabled SFT controls, and step-200 RL checkpoints, rounded to one decimal. Colors indicate unrounded changes from the matched initialization.

	Qwen3-14B-Base				Gemma-3-12B-IT		
Benchmark	Base	SFT	Math RL	Math+Code RL	Base	SFT	Math RL
Math							
MATH500	75.5	87.7	87.4	88.3	80.9	80.3	82.9
AIME 2024	14.3	63.0	33.1	46.2	25.9	31.4	28.4
AIME 2025	14.0	48.7	25.2	31.4	17.4	23.9	21.8
OMEGA	11.4	21.3	17.7	25.1	14.0	12.0	13.8
Coding							
LiveCodeBench	29.4	18.8	35.3	36.7	30.5	8.9	30.8
CodeContests	21.0	11.9	23.6	30.2	24.4	6.9	21.0
DELTA	11.4	3.0	11.2	11.4	9.6	0.5	13.9
Knowledge and Planning							
GPQA-Diamond	42.4	56.1	47.3	49.7	38.1	32.8	35.9
ACPBench	54.3	61.8	66.1	72.1	69.3	49.6	65.0
HEAD-QA	36.1	36.4	37.0	36.7	39.2	34.3	38.7
ToolUse	30.9	4.1	22.7	20.6	43.3	9.3	52.6
Science	42.2	0.0	46.4	47.7	36.7	0.0	32.5
General Capabilities							
CoQA	11.8	1.6	53.6	61.1	52.3	1.6	78.6
IFEval	66.2	22.0	59.6	56.7	69.3	23.9	68.1
HaluEval	35.8	0.0	29.1	30.6	61.2	0.0	60.2
MC-TACO	50.0	66.7	63.5	71.7	76.5	67.4	68.1
WildJailbreak	29.4	3.9	21.0	24.7	27.3	6.2	25.6
WildGuardTest	74.0	24.5	66.3	70.6	77.1	48.9	75.9

Teal: gain; rust: loss. GPQA uses robust extraction; OMEGA averages three splits; DELTA reports BouncingSim test pass rate. IFEval uses strict instruction-level accuracy. Gemma RL CodeContests averages three responses per problem; Gemma base and RL LiveCodeBench average ten where available (Appendix E.2).

E.3 PASS@N

Evaluation protocol. We sample 1,024 independent responses on MATH500, AIME 2024/2025, GPQA-Diamond, CodeContests, and BoxNet at temperature 0.7, top-p 0.95, and a maximum generation length of 16,384.

Complete results. Table 2 reports the finite-budget coverage results underlying Figure 3.

Table 2: Detail breakdown of Pass@N results for the RL and base models in Figure 3.

Benchmark	Checkpoint	Pass@N (%)					
		1	**4**	**16**	**64**	**256**	**1024**
Qwen3-14B-Base							
MATH500	Base	73.00	86.24	91.36	94.17	95.92	97.80
	Math RL	87.13	91.23	93.42	94.84	95.71	96.60
	Math+Code RL	88.13	91.95	93.98	94.85	95.35	95.80
AIME 2024	Base	15.66	32.31	51.42	66.11	75.90	90.00
	Math RL	33.45	54.16	67.22	74.20	78.89	83.33
	Math+Code RL	44.42	62.24	71.08	76.70	81.15	86.67
AIME 2025	Base	12.01	22.79	33.23	45.97	62.32	76.67
	Math RL	24.03	36.59	51.24	64.58	74.47	83.33
	Math+Code RL	31.19	45.61	58.38	67.91	79.07	90.00
CodeContests	Base	19.74	32.32	42.66	51.33	57.92	63.13
	Math RL	25.83	36.80	46.27	53.17	58.81	63.66
	Math+Code RL	31.63	41.74	48.40	52.37	56.08	60.48
GPQA-Diamond	Base	36.70	69.10	91.05	97.79	99.84	100.00
	Math RL	45.95	70.72	84.49	92.38	95.82	97.47
	Math+Code RL	45.41	70.67	85.07	92.73	96.22	98.48
BoxNet	Base	3.83	9.89	17.90	25.20	30.08	34.00
	Math RL	9.88	20.67	28.66	33.88	40.45	51.00
	Math+Code RL	11.24	21.85	30.25	36.58	47.59	67.00
Gemma-3-12B-IT							
MATH500	Base	81.31	87.60	90.62	92.45	93.88	94.80
	Math RL	82.73	88.31	91.24	93.24	94.21	94.40
AIME 2024	Base	24.37	39.25	50.69	62.13	69.67	80.00
	Math RL	27.76	43.21	54.77	63.13	70.50	76.67
AIME 2025	Base	18.13	27.34	36.17	45.13	57.42	66.67
	Math RL	20.54	29.82	39.40	47.81	55.10	60.00
CodeContests	Base	23.27	34.12	42.32	48.20	53.02	55.70
	Math RL	21.24	31.23	39.82	45.61	49.82	54.91
GPQA-Diamond	Base	39.01	64.47	82.58	90.94	94.83	96.97
	Math RL	38.59	66.14	85.24	93.22	96.40	97.47
BoxNet	Base	9.08	11.41	12.84	14.14	15.02	16.00
	Math RL	7.31	14.39	19.19	22.19	25.33	30.00

RL gain RL loss

E.4 SCALING LAWS

Fitting procedure. For each benchmark, let $(C_j, R_j)_{j=1}^m$ be its positive-compute evaluations, where C is cumulative compute in units of 10^2 GPU-hours and scores lie in $[0, 1]$. We fix R_0 to the measured base-model score. The zero-compute evaluation therefore determines R_0 but is excluded from both fitting losses because $\log C$ is undefined at $C = 0$. All positive-compute evaluations receive equal weight.

For the log-sigmoid law in Eq. (6.1), we solve

$$\left(\widehat{A}, \widehat{B}, \widehat{C}_{\text{mid}}\right) \in \underset{\substack{0 \le A \le 1,\ B>0,\\ C_{\text{mid}}>0}}{\arg\min} \sum_{j=1}^{m} \left[R_j - R_0 - \frac{A - R_0}{1 + (C_{\text{mid}}/C_j)^B} \right]^2 . \tag{E.1}$$

To solve this problem, set $q_j = [1 + (C_{\text{mid}}/C_j)^B]^{-1}$. For fixed (B, C_{mid}), the unconstrained optimum is $R_0 + \sum_j q_j(R_j - R_0)/\sum_j q_j^2$; we clip it to $[0, 1]$ to obtain A. We then minimize the profiled loss over $(\log C_{\text{mid}}, \log B)$ using bounded, multi-start Nelder–Mead method. We search $\log B \in [-14, 7]$ and use $C_{\text{mid}} \in [10^{-12}, 10^{28}]$ in normalized compute units. This finite numerical range is effectively unbounded for the reported data, and none of the fits reaches a midpoint boundary. Among all local solutions, we retain the one with the smallest squared error; the numerical search does not certify global optimality.

For the log-linear law in Eq. (6.2), we first fit an unrestricted intercept and slope by ordinary least squares of R_j on $\log C_j$. Because the slopes in all reported fits are nonzero, this is equivalent to solving

$$\left(\widehat{B}_{\text{lin}}, \widehat{C}_{\text{ref}}\right) \in \underset{\substack{B_{\text{lin}} \in \mathbb{R},\\ C_{\text{ref}}>0}}{\arg\min} \sum_{j=1}^{m} \left[R_j - R_0 - B_{\text{lin}} \log(C_j/C_{\text{ref}}) \right]^2 . \tag{E.2}$$

Here $\widehat{B}_{\text{lin}}$ is the fitted slope, and $\widehat{C}_{\text{ref}}$ is the unique compute at which the fitted line equals R_0.

The step-based fits reported below use the same procedure with optimizer step as compute. Steps are divided by 200 for sigmoid conditioning, and the fitted compute scales are converted back to steps.

For either fitted curve $\widehat{R}(C)$, we report

$$\mathsf{R}^2 = 1 - \frac{\sum_{j=1}^{m} \left[R_j - \widehat{R}(C_j) \right]^2}{\sum_{j=1}^{m} (R_j - \overline{R})^2}, \qquad \overline{R} = \frac{1}{m} \sum_{j=1}^{m} R_j, \tag{E.3}$$

using the same positive-compute evaluations as in the corresponding fit; this quantity is defined whenever the denominator is nonzero.

GPU-hour fits. Figure 7 shows the fitted scaling curves, and Table 3 reports their coefficients.

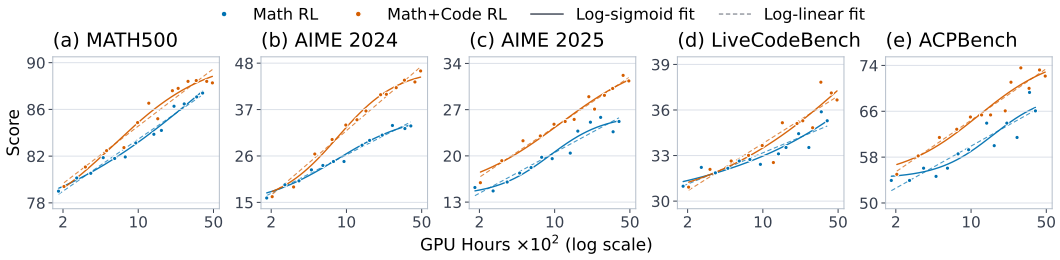


Figure 7: **Scaling behavior of RL compute.** Both laws fit the observed training regime effectively.

Table 3: **Log-sigmoid and log-linear fits using GPU hours.** A is the fitted log-sigmoid asymptote on the raw $[0, 1]$ score scale. C_{mid} and C_{ref} are in units of 10^2 GPU-hours. Both laws fit the observed training regime effectively.

		Log-sigmoid				Log-linear		
RL data	**Benchmark**	B	A	C_{mid}	**Fit R^2**	$B_{\text{lin}}(\times 100)$	C_{ref}	**Fit R^2**
Math	MATH500	0.54	1.000	41.7	0.982	2.76	0.6	0.975
Math	AIME 2024	1.23	0.361	8.1	0.981	5.66	1.2	0.982
Math	AIME 2025	1.71	0.262	9.1	0.942	3.83	1.8	0.920
Math	LiveCodeBench	0.39	1.000	$>10^4$	0.881	1.28	0.5	0.845
Math	ACPBench	1.69	0.693	15.8	0.861	4.56	2.9	0.851
Math+Code	MATH500	1.00	0.904	5.9	0.965	3.05	0.5	0.948
Math+Code	AIME 2024	1.56	0.466	8.3	0.987	9.52	1.6	0.981
Math+Code	AIME 2025	0.96	0.358	11.2	0.962	4.72	1.1	0.973
Math+Code	LiveCodeBench	0.59	0.485	89.3	0.873	1.94	1.0	0.857
Math+Code	ACPBench	1.19	0.762	11.7	0.926	5.59	1.6	0.934

Scaling with training steps. As a robustness check, we use optimizer step as the compute metric. We fit both laws through step 200 and evaluate their extrapolations through step 350. Figure 8 shows the fitted curves using optimizer steps, and Table 4 reports their coefficients. Figure 9 shows the corresponding fitting and extrapolation regimes. Although GPU-hours per step increase as average rollout lengths grow, $\log(\text{GPU Hours})$ is approximately linear in $\log(\text{Steps})$ over these runs; the two compute metrics therefore give similar qualitative conclusions.

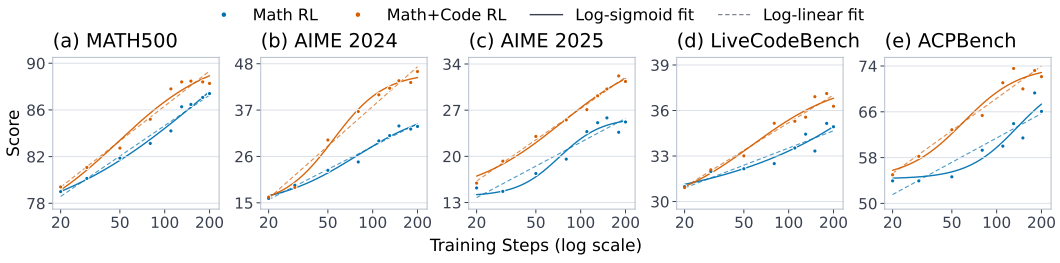


Figure 8: **Training-step scaling on five benchmarks.** Points show measured Qwen3-14B scores for Math RL and Math+Code RL. Solid and dashed curves are log-sigmoid and log-linear fits to optimizer step, respectively. Each panel uses its own truncated score scale.



Figure 9: **Scaling behavior using optimizer steps.** Both laws are fit through step 200 and extrapolated through step 350, yielding the same qualitative picture as GPU-hour compute.

Table 4: **Log-sigmoid and log-linear fits using optimizer step.** A is the fitted log-sigmoid asymptote on the raw $[0, 1]$ score scale. C_{mid} and C_{ref} are in steps.

		Log-sigmoid				Log-linear		
RL data	**Benchmark**	B	A	C_{mid}	**Fit R^2**	$B_{\text{lin}}(\times 100)$	C_{ref}	**Fit R^2**
Math	MATH500	0.76	1.000	207.1	0.985	3.77	8.7	0.976
Math	AIME 2024	1.61	0.379	77.3	0.979	7.83	16.5	0.976
Math	AIME 2025	3.05	0.259	72.1	0.953	5.22	20.9	0.902
Math	LiveCodeBench	0.53	1.000	$>10^4$	0.895	1.64	8.1	0.867
Math	ACPBench	2.60	0.723	137.2	0.899	6.14	31.3	0.827
Math+Code	MATH500	1.33	0.909	48.1	0.976	4.32	8.1	0.962
Math+Code	AIME 2024	2.55	0.458	54.5	0.992	13.57	17.6	0.977
Math+Code	AIME 2025	1.40	0.362	74.2	0.983	6.84	14.3	0.990
Math+Code	LiveCodeBench	1.44	0.380	56.9	0.961	2.63	11.1	0.956
Math+Code	ACPBench	2.29	0.740	59.7	0.956	8.15	17.8	0.944

For Math RL on LiveCodeBench, the step-based fit reaches $A = 1.000$ and gives the unconstrained estimate $C_{\text{mid}} = 2.0 \times 10^4$, far beyond the observed range; its midpoint should therefore be treated as weakly identified.

F Benchmark Cases and Evidence Limits

This section separates three kinds of evidence behind the interpretations in Sections 4 to 6: measured scores or judged responses, the requirements of particular questions, and hypotheses about latent reasoning strategies. The first two do not identify the third. We describe exact evaluation items where saved records permit it; otherwise, benchmark-design examples are explicitly illustrative,

not inspected model responses. All comparisons below use the protocols in Appendix E.2. The HaluEval example uses saved responses for a matched question, ToolUse uses saved evaluation records, and the coverage examples use per-question judgments from the campaigns underlying Figure 3. For the Qwen step-200 RL checkpoints, response text for the AIME and CodeContests coverage samples was not retained; those analyses use saved judgments, not reconstructed RL traces.

F.1 CoT ROUTE CLASSIFICATION AND MEASUREMENT

Questions and responses. The analysis in Section 3.1 compares Qwen3-14B-Base with Qwen3-Math-RL, archived as `qwen3-14b-math-ppo128-step200`, using 16 responses per model and question with seeds 0 to 15. Fixed pools contain 50 questions each from MATH500, GPQA, CodeContests, and LiveCodeBench, and 30 each from AIME24 and AIME25. Answer-masking and content checks exclude 12 MATH500 questions with single-character or multi-value answers or answers appearing in the problem, one AIME25 question whose answer appears in the problem, and one CodeContests question missing essential image content. Six further questions fail route-extraction acceptance checks, `aime24-0008`, `aime24-0028`, `aime25-0019`, `aime25-0020`, `aime25-0027`, and `aime25-0029`, leaving 240 questions. Content eligibility uses neither model responses nor correctness, and correctness is not a selection criterion. These responses are separate from the 1,024-response coverage records in Section 2.3.

Route extraction and grouping. An LLM extracts an outcome-free strategy description with supporting reasoning or code excerpts from each response after embedded-turn cleanup and answer masking. Submitted programs remain route evidence. The route is the method attributed to the adopted conclusion. When that attribution is unavailable, the most developed method is selected, with the first substantive method breaking ties. The pooled base and RL descriptions are grouped within each question, with model identity, correctness, and explicit final-answer labels withheld. Problem representation and the key mechanism determine route identity. Equivalent algebraic forms, local errors, completion differences, and manual or coded execution of the same method do not split a route. Independent model audits and targeted repairs precede acceptance of each question's grouping.

Identifiable routes remain classified in truncated or repetitive responses. Responses with absent or unclear routes form a separate category excluded from route counts. It contains 406 base responses, including 358 of the 800 base responses on GPQA, and no RL responses. Every question retains at least one identifiable response from each model. Embedded-turn cleanup can remove solution evidence. For example, five inspected base responses to `aime25-0006` retain only a removal marker after cleanup. Route annotations describe the strategies visible in these preprocessed inputs.

Route counts and base-sample membership. For each question, let m be the smaller number of identifiable responses from the two models. We average the number of distinct routes over all subsets of m identifiable responses from each model, and then average equally over questions within each benchmark and across all 240 questions. The overall mean therefore weights benchmarks by their question counts.

For Figure 4(b), we compute the fraction of each question's 16 RL responses whose route does not occur among the original 16 base responses to that question. This check uses the original samples rather than the equal-size subsets. Because every question has 16 RL responses, the mean over questions equals the pooled fraction of 487 of 3,840 responses (12.7%).

Uncertainty. For each benchmark and all 240 questions, we use 5,000 question-bootstrap resamples with seed 20260924. Questions are sampled with replacement, keeping each question's two model counts together, to recompute the means and paired difference. The 95% confidence intervals are the 2.5th and 97.5th percentiles of the bootstrap values. Figure 4(a) shows intervals for each model's mean, and the main text reports the paired difference. Panel (b) uses the same resampling design for the mean question-level fraction of RL responses with routes absent from the base samples. The intervals condition on the selected questions, saved responses, and fixed route annotations.

F.1.1 CONCRETE ROUTE EXAMPLES

Each example has 16 identifiable responses per model, so the equal-size comparison uses all responses. Mechanisms below follow the accepted groupings and reasoning or code excerpts in saved descriptors, which document attempted operations rather than evaluation outcomes.

Concentration in array comparisons. `livecodebench-0088` minimizes $|\text{nums}[i] - \text{nums}[j]|$ subject to $|i - j| \geq x$. Eight base responses binary-search a sorted list and compare neighboring values, six scan index pairs with nested loops, and two use indices sorted by value. All 16 Qwen3-Math-RL responses use the binary-search route, reducing the route count from three to one.

Binary search in pattern matching. `livecodebench-0163` finds positions of pattern a within distance k of an occurrence of pattern b. All 16 base responses compare occurrence positions directly. Five Qwen3-Math-RL responses do the same; 11 binary-search the occurrence list of b at the lower boundary of $[i - k, i + k]$. This route is absent from the base samples, and the route count rises from one to two.

Fifth-root algebra and cosine pairing. `math500-0069` evaluates $z + 1/z + z^2 + 1/z^2$ for $z^5 = 1$ and $z \neq 1$. Thirteen base responses replace reciprocal powers by z^4 and z^3 and use $1 + z + z^2 + z^3 + z^4 = 0$. Three write $z = e^{2k\pi i/5}$ and pair inverse powers into cosine terms. All 16 Qwen3-Math-RL responses use the power-sum reduction, reducing the route count from two to one.

Divisibility in integer bases. `aime25-0000` sums integer bases $b > 9$ for which 17_b divides 97_b. Fifteen base responses introduce an integer quotient k and rearrange $9b + 7 = k(b + 7)$ to restrict b. One enumerates bases and tests $(9b + 7) \bmod (b + 7) = 0$. All 16 Qwen3-Math-RL responses use the algebraic route, reducing the route count from two to one.

Fewer graph routes with transitive closure. `livecodebench-0256` asks whether superiority relations determine a unique strongest programmer. The base and Qwen3-Math-RL samples contain six and five routes. Base routes include topological ordering and candidate-specific graph traversal. Four RL responses compute transitive closure of a Boolean reachability matrix, adding connections through intermediate nodes. This route is absent from the base samples despite the lower RL route count.

Split enumeration within a serving-count search. `livecodebench-0371` maximizes combined integer servings of two dishes under ingredient limits. The base and Qwen3-Math-RL samples contain six and four routes. One RL response binary-searches the total s and enumerates splits $x = 0, \ldots, s, y = s - x$ within each feasibility test. This route is absent from the base samples. A distinct binary-search route without split enumeration occurs in one base and two RL responses.

F.2 TRANSFER, FORGETTING, AND OUTPUT BEHAVIOR

ToolUse output behavior. ToolUse requires a specified `Action` name and structured `Action Input` arguments (Shenfeld et al., 2026a). Qwen's score falls from 30.9% to 22.7% under math RL, while Gemma's rises from 43.3% to 52.6% (Table 1). One saved Qwen item (evaluation index 5) gives a concrete output-level failure. On a request for five axolotl physical-characteristic facts with source URLs, the base emits one `Action: getAxolotlFacts` call with category "physical characteristics" and limit 5 and is graded correct. The RL response first makes the same call, but then says "Please provide the output from the getAxolotlFacts tool," simulates further interaction, and repeats calls. The saved RL output contains three `Action:` lines; the scorer requires the exact call and arguments and grades it incorrect. This item directly exhibits an interaction/format failure, not evidence that RL selected a target-invalid reasoning strategy. The full ToolUse score change cannot be explained by this one case; matched action-name, argument, and extra-call error rates would quantify its prevalence.

Table 5: AIME questions responsible for Qwen's changed empirical Pass@1024 coverage under math RL. Entries are correct responses out of 1,024; task descriptions paraphrase the evaluated prompts.

Set	Index	Requirement	Base	Math RL
2024	8	Count maximal placements of two colors of chips on a 5×5 grid.	3	0
2024	25	Count octagon colorings satisfying a rotation condition.	1	0
2025	11	Find the area of a bounded region defined by nonlinear inequalities in a plane.	0	24
2025	27	Compute a modular remainder from a long rational recurrence.	0	1

Other transfer contrasts. Qwen's math RL improves LiveCodeBench from 29.4% to 35.3% but changes DELTA only from 11.4% to 11.2% in Table 1. LiveCodeBench asks for programs passing executable tests (Jain et al., 2024), whereas DELTA includes BouncingSim, which asks for a physics simulator and here reports per-test rather than full-program success (Sun et al., 2025a). A route that helps derive a contest algorithm need not provide the numerical modeling or implementation needed for a simulator. But different test metrics and answer code can also explain the contrast; the table does not distinguish these from strategy selection. Likewise, OMEGA mixes problems that extend a familiar method, compose skills, or reframe the problem (Sun et al., 2025b). A fixed strategy library can select among available routes but cannot by itself certify that RL invented a new one.

Outside these reasoning-focused tasks, CoQA requires answers grounded in the supplied passage (Reddy et al., 2019), whereas IFEval checks explicit output constraints, HaluEval grades a specified label, and the safety tasks probe harmful-request handling and benign-request over-refusal (Zhou et al., 2023; Li et al., 2023; Jiang et al., 2024; Han et al., 2024). CoQA improves under all three RL runs in Table 1, while IFEval, HaluEval, and both safety scores decline. These scores do not show that the same latent route is shared across all such prompts. One matched HaluEval QA item (evaluation index 0) contains a candidate claim that *First for Women* predates *Arthur's Magazine*; the supplied evidence puts *Arthur's Magazine* earlier, so the claim is a hallucination. Qwen base answers "Yes" correctly, while math RL answers "No" incorrectly. Both are valid labels: at least this loss is a judgment reversal, not an answer-format failure. Across the saved HaluEval QA subset, base and RL correct counts are $2{,}970/10{,}000$ and $1{,}758/10{,}000$; dialogue also falls ($4{,}919$ to $4{,}064$), while summarization rises slightly ($2{,}847$ to $2{,}896$). The aggregate loss therefore mixes distinct question types and cannot be attributed to one formatting mechanism. For safety, harmful-request failures and benign over-refusal still need to be separated. Both SFT controls score zero on HaluEval, but their aggregate scores alone cannot distinguish an invalid label from an incorrect judgment.

F.3 Prompt-level Pass@N evidence and limitations

For the benchmark distribution $\mathcal{B}$, any end-to-end policy π satisfies the exact identity

$$\text{Pass@}\,N(\pi) - \text{Pass@}\,N(\pi_0) = \mathbb{E}_{x' \sim \mathcal{B}}\left[(1 - R_{\pi_0}(x'))^N - (1 - R_{\pi}(x'))^N\right]. \tag{F.1}$$

Thus the macro-level reversals in Figure 3 reflect changes in the distribution of promptwise success probabilities. They do not identify which questions moved or whether the change arose from strategy selection, answer construction, execution, or formatting. We therefore examine the available per-question records below.

Benchmark dependence: AIME 2024 versus AIME 2025. For the Qwen coverage campaigns, each question has 1,024 sampled responses, reused to estimate Pass@N at different budgets. The per-question records show that the two-question AIME differences at $N = 1024$ arise as follows. Indices below are zero-based positions in the evaluated sets, not claims about official problem numbering.

There are no oppositely directed coverage changes on the other AIME questions: base versus math RL reaches 27 versus 25 of 30 AIME 2024 items, and 23 versus 25 of 30 AIME 2025 items. Three of the four boundary items have only one to three successes on their reached side. Treating questions

as paired sampled items gives a standard error of about 4.6 percentage points for either $2/30$ coverage difference, before accounting for response-sampling uncertainty. The exact discordant-pair test has two-sided $p = 0.5$ in each set. These observations demonstrate changed *sampled* coverage, not stable benchmark rankings or literal absence of a correct strategy. With 0 successes in 1,024 independent draws, even a one-sided 95% binomial bound permits underlying success probability up to approximately 0.29%.

Training-data dependence: CodeContests. The CodeContests reversal has a different structure. Qwen's base and mathematics-only RL checkpoints reach 238 and 240 of the same 377 questions, whereas mathematics-plus-code RL reaches 228. Thus adding code training does not mechanically expand coverage on a code benchmark. The mathematics-plus-code checkpoint nevertheless has higher Pass@1 than the base. There are 22 base-only and 12 RL-only questions. Two informative examples from the judged records are:

- `codecontests-1192` asks for the sum of all pairwise absolute differences among up to 10^5 strengths, multiplied by the maximum and reduced modulo $10^9 + 7$. The base solves it in $375/1024$ draws; mixed RL solves it in $0/1024$. Every saved RL judgment for this item is a timeout. Among base judgments, 375 succeed, 447 time out, and 202 are wrong. A judged-correct base rollout explicitly begins, "This can be done by sorting the array of strengths and then using the formula for the sum of absolute differences in a sorted array," and implements the resulting $O(n \log n)$ calculation. The RL response text is unavailable, so the timeouts cannot be assigned to a particular algorithm or code defect.
- `codecontests-1332` asks for the minimum number of subset toggles needed to turn on each prefix of a lamp array, where no element belongs to three subsets. The base solves it in $0/1024$ draws and mixed RL in $74/1024$. The base judgments comprise $1{,}021$ wrong answers and three timeouts; the RL judgments comprise 74 successes, 907 wrong answers, and 43 timeouts. This shows a gain on a different algorithmic demand, but the surviving judgments do not reveal the RL solution route.

Treating questions as exchangeable paired items, the standard error of the $10/377$ coverage difference is about 1.5 percentage points; the exact discordant-pair test gives two-sided $p \approx 0.12$. This does not include uncertainty from which rare successes happened to appear in 1,024 draws. Code training uses a fraction-of-tests reward, whereas coverage requires a fully correct program. Algorithm choice, implementation, runtime, test coverage, and output format can each affect these records. The timeout evidence for `codecontests-1192` in particular warns against explaining the reversal as measured deletion of a reasoning strategy.

Base-model dependence: AIME 2025 and BoxNet. Under mathematics RL, sampled AIME 2025 coverage rises from $23/30$ to $25/30$ for Qwen but falls from $20/30$ to $18/30$ for Gemma. This cross-family reversal is descriptive: with only 30 questions, it does not by itself identify the strategies responsible. Gemma's BoxNet results provide a more detailed example of how the success distribution can change within one model family.

On BoxNet, Gemma's math RL lowers Pass@1 from 9.1% to 7.3% but raises observed Pass@1024 coverage from $16/100$ to $30/100$ questions. The task requires a legal sequence of coordinate-specific moves and a structured final plan (Stojanovski et al., 2026). Among the questions reached by each policy, its sampled mean success fraction drops from roughly 57% under the base to 24% under RL. This is observed breadth versus concentration: the policies both reach 16 items, RL alone reaches 14, and neither reaches the remaining 70. On the 16 shared items, the base has $9{,}296$ correct draws and RL $6{,}477$; the 14 newly reached items contribute $1{,}011$ RL correct draws. These counts explain the lower Pass@1 despite broader coverage, as correct draws on newly reached questions do not offset losses on shared questions.

One matched item, `boxnet-0098`, makes the question-level distinction visible. The prompt places red, blue, and green boxes near one agent, but the blue target is in an adjacent square; the blue box must first be moved there. The base has $0/1024$ successful draws, whereas Gemma RL has $749/1024$. In seed 0001 the parsed base plan contains only `move(box_red,target_red)`, scoring one third. The RL plan first moves the red box to its target, then uses `move(box_blue,square[0.5,1.5])` followed by `move(box_blue,target_blue)`, and finally moves the green box to its target; it scores one.

Both outputs are valid fenced JSON, so this particular difference is plan completeness rather than a formatting failure. A matched pair of rollouts still does not establish a change in latent strategies or prove that RL created a route absent from the base distribution.

Together, these records give prompt-level counterexamples to any single-factor account of the aggregate curves. The same Qwen math checkpoint loses sampled coverage on two AIME 2024 questions but gains it on two AIME 2025 questions; CodeContests contains both base-only and RL-only successes; and Gemma's BoxNet checkpoint broadens question coverage while reducing success on the questions reached by both policies. The observed reversals can therefore combine benchmark composition, algorithm or plan choice, execution failures, and probability redistribution. They are compatible with strategy selection, but do not identify it without trace-level evidence.

The 30% coverage also does not establish the stronger low-regularization regime of Theorem 5.1 (II), which would imply expected coverage above 50% under its assumptions. The model-agnostic identity Eq. (F.1), not the ideal decoder theorem alone, is what directly interprets these empirical curves. If every question retains positive success probability, both policies' Pass@N eventually approach one; zero observed successes at $N = 1024$ cannot establish literal strategy erasure.

F.4 What the fitted scaling coefficients do and do not measure

The fits in Figure 7 and table 3 use evaluations through step 200 from two multi-prompt GRPO runs. Our theorems also concern benchmark means, but assume one training prompt and idealized strategy dynamics. The fitted exponent B therefore cannot be identified with a single prompt's feature overlap. The table's B_{lin} column reports $B_{\mathrm{lin}}(\times 100)$ in percentage points per unit natural log of GPU-hours; dividing by 100 gives the raw-reward convention used by the theoretical slope. The benchmark log-linear bound depends on gain-weighted overlaps in Eq. (D.21), not a single prompt's overlap. Promptwise transitions with different onset times can also average into a nearly log-linear curve over an interval longer than an individual prompt's local window.

LiveCodeBench requires executable programs, not just a plausible pre-answer plan. Under math-only and math-plus-code RL, respectively, Table 3 gives $(B, B_{\mathrm{lin}}(\times 100)) = (0.37, 1.18)$ and $(0.92, 1.86)$. The math-only fit places C_{mid} around 3.15 million GPU-hours and reaches $A = 1.000$, far beyond the observed compute range; this extrapolative midpoint is weakly identified. The fitted slopes therefore neither identify a shared reasoning route nor show that answer construction has reached a plateau. MATH500 and the two AIME sets also have different baselines, problem mixes, and attainable gains. The AIME items in Table 5, for example, range from combinatorial counting to a long rational recurrence; benchmark-average scaling curves do not identify which question types improve with compute. The distinct fitted exponents need not be different feature overlaps. These in-sample fits describe the measured fitting window; the coefficients alone do not establish an observed plateau or a universal long-range forecast.

Across these cases, the directly observed changes are extra tool calls, different factual judgments, timeouts, and more complete action plans. None measures the feature overlap $c(x', x)$ or identifies the selected strategy set $\mathcal{S}(x')$. Testing that mechanism would require labeling comparable rollouts by their pre-answer strategy, measuring how strategy frequencies change under RL, and reporting correctness conditional on strategy while separately tracking format and execution failures.