# From Base Rollouts to RL Reasoning: A Budgeted Search Perspective

**Anonymous ACL submission**

## Abstract

Reinforcement learning with verifiable rewards (RLVR) improves language-model reasoning, but how these gains relate to inference-time decoding and search remains unclear. We ask whether RL primarily introduces new behavior or makes reasoning trajectories already accessible to the base model easier to sample under a limited inference budget. To study this question behaviorally, we introduce a Unified Decoding Framework (UDF), which represents token-level sampling, beam-like search, tree search, and sequence-level resampling as executable policies in a shared budgeted operating space. Each operating point is evaluated with pass@$k$, self-consistency, best-of-$N$, and first-finish success, keeping generation policies separate from evaluation metrics.

Using paired Base/RL checkpoints from SimpleRL-Zoo, we test whether an RL default-policy curve can be approximated by a structured path of Base operating points $(\pi, N_{\text{Base}})$, rather than by same-budget comparisons alone. Across Math500, AIME, GPQA, and IFEval, the pass@$k$ recovery path well described by a Budgeted Operating-Point Transition Rule (BOPTR), $N_{\text{Base}} \approx \alpha N_{\text{RL}}^{\beta}$, with regime-conditioned exponents $\beta_{\text{math}} = 0.6$, $\beta_{\text{nonmath}} = 1.0$, and $\beta_{\text{floor}} = 0$. On Qwen2.5-7B, BOPTR-P1 achieves the lowest non-oracle horizontal-transfer error among the tested non-oracle rules, 3.41 pp. Across models, the formula family remains informative, but its parameters and selected policies are model-specific. Qwen models trained under the same difficulty split share useful operating-point structure, whereas cross-family models or different training distributions reveal the limits of direct transfer. We use UDF and BOPTR as behavioral diagnostics for how RL reshapes decoding and search under the tested SimpleRL-Zoo recipe, rather than as universal post-RL performance predictors.

## 1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a central post-training recipe for improving reasoning in large language models (OpenAI, 2024; Guo et al., 2025). Open recipes such as DeepSeekMath and SimpleRL-Zoo further show that simple verifiable-reward training can substantially improve mathematical reasoning in open base models (Shao et al., 2024; Zeng et al., 2025). Yet recent analyzes suggest that these gains may not always reflect entirely new reasoning behavior. Base models can sometimes approach RL-level performance under larger sampling budgets, whereas RL-trained models often dominate in low-budget regimes (Yue et al., 2025). This suggests an alternative behavioral interpretation: RL may improve *sampling efficiency* by making correct reasoning trajectories easier to sample under a fixed inference budget. Related work further suggests that RL post-training can amplify behaviors already present in the pretraining distribution (Zhao et al., 2025). These observations motivate our central question: how does RL change the observable behavior of the base model's default rollout distribution?

We view decoding on a single problem as allocating a finite rollout budget over an implicit rollout space: the reasoning trajectories induced by the prompt and model, as illustrated in Fig. 1(1). A base model may spread this budget across diverse trajectories, including correct solutions in its support that are hard to sample by default. RL can be understood as shifting the default allocation toward reward-aligned trajectories, increasing the chance of reaching successful reasoning paths under small budgets. This shift, however, need not be unique to RL-trained parameters: external decoding and search over the base model can also reallocate budget by broadening exploration, concentrating probability mass, or selecting among alternatives. We therefore test a behavioral form of the internalized-

search hypothesis: whether the RL model's default rollout behavior can be recovered by budgeted external search over the base model.

Testing this hypothesis requires addressing three challenges. First, decoding and search methods are heterogeneous: token-level sampling, beam-like expansion, tree-style exploration, and sequence-level resampling have different control structures. Second, recovery can be overstated: a large policy grid may yield post-hoc matches to an RL curve, so meaningful recovery should form a low-complexity path rather than unrelated per-budget choices. Third, such structure may depend on task regime and model family, so a rule observed on Math500 cannot be assumed to transfer unchanged to AIME, GPQA, IFEval, or models trained under different difficulty splits.

We make this question testable with **SearchLens**, a Unified Decoding Framework (UDF). UDF represents token-level sampling, beam-like expansion, tree-style exploration, and sequence-level resampling as executable policies in a shared budgeted operating space. Each inference-time method is paired with a budget to form an operating point $(\pi, N_{\text{Base}})$, while metrics such as pass@$k$, self-consistency, best-of-$N$, and first-finish success are applied only after generation. This separation is important: we do not ask only whether Base matches RL under the same policy and budget, but whether the RL default-policy curve can be recovered somewhere in the Base model's broader policy–budget landscape. We then ask whether the recovering points form a low-complexity path across budgets, benchmarks, and models, rather than a collection of unrelated post-hoc matches. Such paths identify which parts of the RL gain are already accessible through Base inference-time search, while exposing regions outside the tested operating landscape.

Across paired Base/RL checkpoints from SimpleRL-Zoo, we find that RL default-policy curves are often recoverable as Base operating paths. In the pass@$k$ channel, the recovered Base budget follows a regime-conditioned transition rule, $N_{\text{Base}} \approx \alpha N_{\text{RL}}^{\beta}$: Math500 is sublinear, GPQA and IFEval are near-linear OOD regimes, and AIME is floor-bound. Across models, the same formula family remains informative, but the multiplier and selected policy are model-specific. This yields a qualified internalized-search interpretation: RL often moves the model through the base operating landscape, but the concrete recovery path depends on both benchmark regime and model family. In particular, models trained under the same SimpleRL-Zoo difficulty split share more transferable operating-point structure, whereas OOD benchmarks, cross-family models, and different training distributions mark the limits of direct transfer.

Our contributions are:

- UDF (SearchLens), a unified budgeted operating space for external decoding and search policies, separated from evaluation metrics.
- A two-dimensional recovery analysis that asks whether RL default-policy curves can be matched by Base operating points $(\pi, N_{\text{Base}})$, rather than by same-budget comparisons alone.
- **BOPTR-P1** (SearchPath), a low-dimensional pass@$k$ rule in which the effective Base budget follows a benchmark-regime-conditioned exponent.
- Horizontal and vertical transfer analyses under SimpleRL-Zoo, showing shared operating-point structure within the same difficulty split and clear limits under OOD, cross-family, or different-training-distribution settings.

Code, configs, and analysis scripts are released at https://anonymous.4open.science/r/Searchlens_boptr-6B42/.

## 2 Related Work

**RL and reward-based post-training.** Reinforcement learning from human or verifiable rewards has become a standard approach for shaping pretrained language models. Early instruction-following work uses RLHF with PPO over human-preference reward models (Ouyang et al., 2022; Schulman et al., 2017), while DPO reformulates preference optimization as a direct objective without an explicit reward model (Rafailov et al., 2023). For reasoning, RL systems such as OpenAI o1 and DeepSeek-R1 demonstrate that reinforcement learning can substantially improve complex reasoning behavior (OpenAI, 2024; Guo et al., 2025). Open RL-with-verifiable-rewards (RLVR) recipes such as DeepSeekMath and SimpleRL-Zoo use programmatic or rule-based correctness signals to train open base models (Shao et al., 2024; Zeng et al., 2025). SimpleRL-Zoo is central to our study because it provides paired Base/RL checkpoints across model

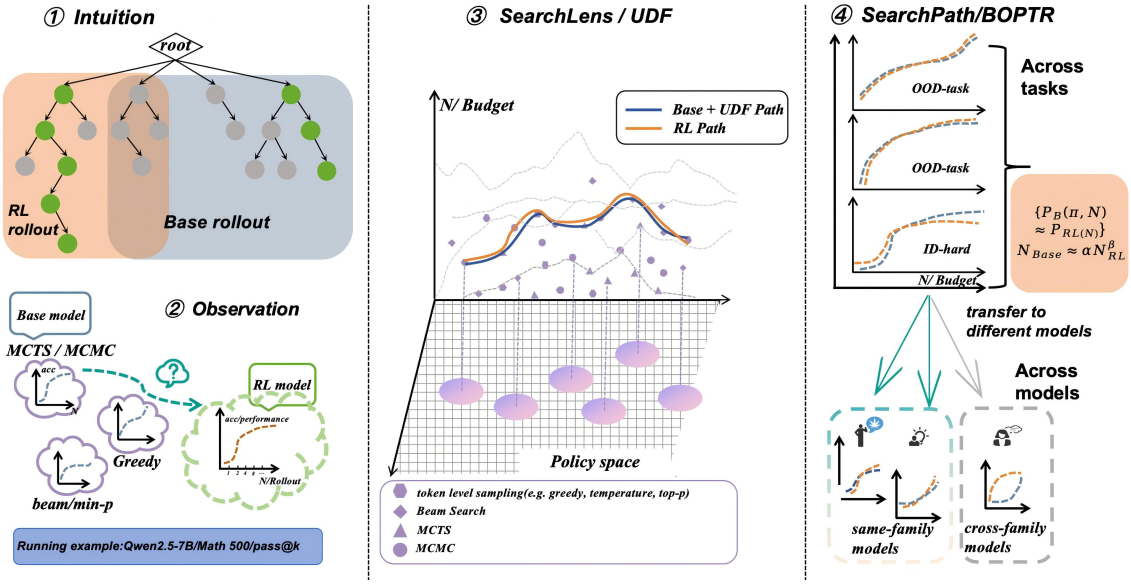


Figure 1: Overview of the analysis pipeline. We first view decoding as rollout-budget allocation over an implicit reasoning tree, then compare RL behavior with external decoding/search policies applied to the base model. SearchLens/UDF places these policies in a shared budgeted operating landscape, and SearchPath/BOPTR asks whether RL default-policy curves can be recovered as structured Base+UDF paths following $N_{\text{Base}} \approx \alpha N_{\text{RL}}^{\beta}$ across tasks and models.

scales and families under a shared verifiable-reward recipe (Zeng et al., 2025). We use these checkpoints not to propose another RL method, but to analyze how RL-trained models relate behaviorally to their base counterparts under decoding and search.

**What does RL change?** Recent work asks whether RLVR creates new reasoning abilities or primarily reweights behaviors already present in the base distribution. RLVR-trained models often outperform base models at small sampling budgets, while base models can recover much of the gap under large pass@$k$, suggesting improved sampling efficiency and possible narrowing of the reasoning boundary (Yue et al., 2025). Other studies similarly frame RL gains as amplification of pretraining patterns or distribution-level generalization beyond supervised fine-tuning (Zhao et al., 2025; Chu et al., 2025). Karan and Du (2025) further show that training-free inference-time sampling from base models can approach or outperform RL-posttrained models on several reasoning benchmarks, suggesting that some RL gains can be elicited without additional training. Together, these findings motivate a behavioral view of RL: post-training may shift the default rollout distribution toward high-reward trajectories that are already accessible to the base model. We therefore compare RL not only with the base model under its default policy, but also with the base model's full budgeted decoding and search landscape.

**Inference-time search, selection, and behavioral analysis.** Inference-time compute improves reasoning through chain-of-thought prompting, self-consistency, repeated sampling, explicit tree/graph/planner-style search, and verifier- or reward-model-based selection (Wei et al., 2022; Wang et al., 2023; Brown et al., 2024; Yao et al., 2023; Besta et al., 2024; Hao et al., 2023; Zhou et al., 2024; Cobbe et al., 2021; Lightman et al., 2024; Chow et al., 2025). Our UDF framework represents these generation and search mechanisms as budgeted policies in a common operating space, while metrics such as pass@$k$, self-consistency, best-of-$N$, and first-finish success summarize the resulting candidate sets. Because chain-of-thought rationales need not faithfully describe a model's internal computation (Turpin et al., 2023; Lanham et al., 2023; Jacovi and Goldberg, 2020), we focus on observable input–output behavior rather than parameter-level mechanisms. This positions our analysis between parameter-space studies of post-training and purely outcome-level benchmark comparisons.

## 3 SearchLens: A Unified Decoding/Search Landscape

A decoding or search method is an inference-time policy that maps a prompt to candidate responses. SearchLens, our Unified Decoding Framework (UDF), represents each method as an operating

point

$$u = (\pi, n), \tag{1}$$

where $\pi \in \Pi_{\text{UDF}}$ is a policy configuration and $n \in \mathcal{B}$ is an inference budget. Evaluation metrics are applied only after generation and are not part of the policy definition. In our experiments, $\mathcal{B} = \{1, 2, 4, 8, 16\}$. Each operating point specifies one allocation of a finite rollout budget over possible trajectories, without assuming that this space is internally represented by the base model.

**Components.** A policy $\pi$ consists of five components (Figure 2):

1. **Local policy**: transforms logits through proposal, gating, scoring, or allocation, covering greedy decoding, temperature sampling, top-$k$, top-$p$, min-$p$, and typical sampling.

2. **Controller**: specifies the search topology, including autoregressive, beam-like, branching, tree-style, or resampling controllers.

3. **Evaluator**: scores partial or complete candidates using likelihood, rollout utility, or an external verifier.

4. **Transition**: updates the state by appending, expanding, branching, or resampling.

5. **Scheduler**: allocates budget across samples, branches, or resampling steps.

**Evaluation metrics.** We summarize each candidate set with four metrics: pass@$k$ for support, SC for majority-vote concentration, best-of-$N$ for evaluator-mediated selection, and FFS for early valid-answer success. These metrics are not part of UDF; they provide complementary behavioral views of the same generated set.

**Policy geometry.** For structured analyses, we associate each operating point with

$$z(u) = [P(u), Q(u), T(u), D(u), C(u)], \tag{2}$$

where the coordinates denote pass@$k$, SC, first-finish/validity, diversity or the pass–SC gap, and cost. We combine component, behavioral, and budget distances:

$$\begin{aligned} d(u_i, u_j) = \lambda_{comp}\, d_{comp}(\pi_i, \pi_j) \\ + \lambda_z \|z(u_i) - z(u_j)\|_1 \\ + \lambda_b |\log n_i - \log n_j|. \end{aligned} \tag{3}$$

This distance is used only for path diagnostics, not for mechanistic identification.

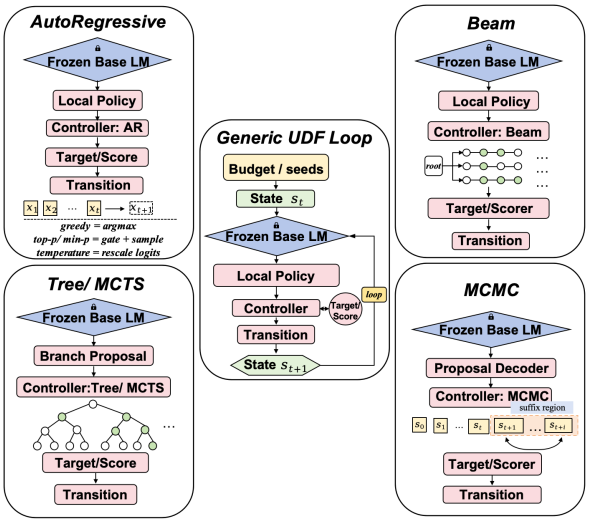


Figure 2: The Unified Decoding Framework (UDF). A frozen base language model is wrapped by a five-component policy—local policy, controller, evaluator, transition, and scheduler—which together define an operating point $u = (\pi, n)$ in the budgeted landscape. Evaluation metrics, including pass@$k$, SC, BoN, and FFS, are applied after generation and are not part of UDF.

**From SearchLens to SearchPath.** SearchLens defines the operating space; SearchPath/BOPTR asks whether the RL curve can be recovered as a low-dimensional trajectory through it. Appendix A.1 verifies that this five-component parameterization instantiates eleven mainstream decoding algorithms against vLLM, HuggingFace, and MCMC reference implementations.

## 4 Experimental Setup

We use Qwen2.5-7B/Math500 as the anchor cell for constructing the SearchLens landscape, recovery analysis, and BOPTR rule. Other model–benchmark cells test horizontal transfer, vertical transfer, and cross-family stress settings, as summarized in Table 1.

**Model pairs.** We analyze paired Base/RL checkpoints from SimpleRL-Zoo, including Qwen2.5-0.5B/1.5B/7B/14B, Qwen2.5-Math-7B, Llama3.1-8B, and Mistral-7B-v0.1 when compatible generations are available. Models trained on the SimpleRL-Zoo Hard RL-training split form the upper cohort; cross-family or weak-capacity models form the lower cohort.

**Benchmarks.** We evaluate Math500 (Hendrycks et al., 2021; Lightman et al., 2024), AIME 2024/2025, GPQA-Diamond (Rein et al., 2023), and IFEval (Zhou et al., 2023). IFEval is treated as

| Block | Settings | Role |
|---|---|---|
| Anchor | Qwen2.5-7B / Math500 | Full landscape construction and rule discovery. |
| Horizontal | Qwen2.5-7B / AIME, GPQA, IFEval | Same model, new benchmarks. |
| Vertical | Qwen2.5 family scales | Same family, different scale. |
| Stress | Llama3.1-8B, protocol-sensitive models | Cross-family and generation-protocol stress tests. |

Table 1: Evaluation blocks. We report horizontal and vertical results separately because they test different transfer hypotheses.

an instruction-following and termination stress test, with pass rate, validity, and first-finish success as primary metrics and SC used diagnostically.

**Policies, metrics, and cost.** For each model and benchmark, we evaluate token-level autoregressive policies and, when available, expanded search policies such as MCMC-like resampling, entropy branching, and tree-style controllers, with budgets $1, 2, 4, 8, 16$. Our main analyses use same-policy curves, Base+UDF envelopes, near-match policies, and low-complexity rule families; tables report MAE in percentage points relative to the RL default-policy target. The anchor cell requires approximately 461 H100 GPU-hours, dominated by eight expanded-controller runs. Generations are deduplicated at $b = 16$ and reused across smaller budgets and all four metrics.

## 5 From Same-Policy Gains to Base Search Recovery

We analyze the Qwen2.5-7B/Math500 anchor cell in two steps. First, same-policy comparisons test whether RL changes behavior under a fixed decoding policy and budget. Second, Base+UDF recovery tests whether the RL default-policy curve lies inside the base model's decoding/search landscape. Together, these establish the prerequisites for SearchPath: a measurable RL shift and a Base+UDF support region that contains it.

### 5.1 Same-Policy Gains: Where RL Improves Base

Same-policy comparison isolates post-training effects from gains due to richer inference-time policies. Let $A^c_M(\pi, b)$ denote the accuracy of model $M$ under metric $c$, policy $\pi$, and budget $b$. We define

$$\Delta^c_{same}(b) = A^c_{\mathrm{RL}}(\pi_0, b) - A^c_{\mathrm{Base}}(\pi_0, b), \quad (4)$$

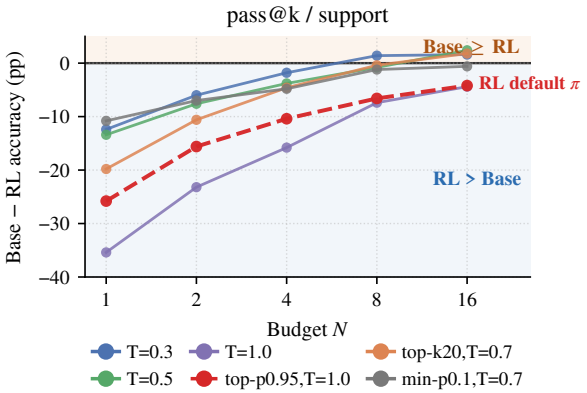


Figure 3: Same-policy Base–RL gap on Qwen2.5-7B/Math500 under pass@$k$. Each line fixes a decoding policy $\pi$ and plots $A_{\mathrm{Base}}(\pi, N) - A_{\mathrm{RL}}(\pi, N)$ over budgets $N \in \{1, 2, 4, 8, 16\}$. Positive values indicate Base $\geq$ RL. The dashed red line is the RL default policy. Base catches up only at larger budgets: the number of Base-shared policies with Base $\geq$ RL is 0/24, 0/24, 0/24, 2/24, and 13/24 across increasing budgets.

where $\pi_0$ is the RL default policy.

**Observed patterns.** Figure 3 shows three patterns. RL improves low-budget pass@$k$ and SC on math-style tasks; Base can recover or exceed RL pass@$k$ at larger budgets; and GPQA/IFEval shift metric semantics, with GPQA behaving more like concentration and IFEval better analyzed through validity and first-finish success.

### 5.2 Base+Search Envelopes: Behavioral Recoverability

We next ask whether external decoding/search over the base model can approximate the RL default-policy curve. This is a behavioral recoverability question, not a claim that RL implements an external policy.

**Per-budget support region.** For each metric $c$ and budget $b$, define the Base+UDF upper envelope:

$$E^c_{\mathrm{Base}}(b) = \max_{\pi \in \Pi_{\mathrm{UDF}},\, n \leq b} A^c_{\mathrm{Base}}(\pi, n). \quad (5)$$

Together with the low-quantile boundary of the Base policy cloud, this forms the support region in Figure 4. The recovery gap,

$$G^c(b) = A^c_{\mathrm{RL}}(\pi_0, b) - E^c_{\mathrm{Base}}(b),$$

is negative when Base+UDF exceeds the RL target and positive but small when the RL curve lies near the Base+UDF ceiling. The RL target lies inside this support region at every budget.

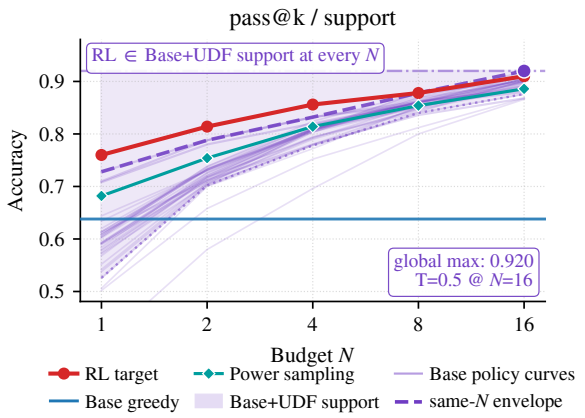


Figure 4: Per-budget Base+UDF support region on Qwen2.5-7B/Math500 under pass@$k$. Thin curves show Base policy trajectories; the shaded region spans the Base policy cloud and the Base+UDF upper envelope. Red denotes the RL target, blue Base greedy, and teal the power-sampling baseline. The RL default target lies inside the Base+UDF support region at every budget.

**Near-match sets and recovery paths.** Because exact matches can arise post hoc in a large policy space, we define near-match sets

$$\mathcal{N}_\epsilon^c(b) = \{(\pi, n) : |A^c_{\text{Base}}(\pi, n) - A^c_{\text{RL}}(\pi_0, b)| \le \epsilon\}, \tag{6}$$

and select one closest Base+UDF candidate for each RL budget to form a recovery path. This distinguishes structured recovery from unrelated per-budget lookup. This distinction is important because recovery is not a same-budget comparison. A Base model may underperform RL under the same default policy and budget, yet still contain Base+UDF operating points that match the RL target when policy and budget are allowed to vary. Conversely, an envelope match alone is not sufficient evidence for a structured explanation. We therefore treat near-match sets as the feasible region, and use the selected recovery path to test whether the matches form a coherent trajectory rather than isolated post-hoc choices.

Appendix Table 7 summarizes each recovery object and its main risk.

## 6 SearchPath: Low-Dimensional Recovery Paths

The recovery analysis in Section 5.2 shows that RL targets have multiple Base+UDF matches at each budget. SearchPath asks whether these matches form a transferable path rather than independent lookup results. We formalize this path as a Budgeted Operating-Point Transition Rule (BOPTR). BOPTR-P1, the pass@$k$ instance, uses two structural assumptions: the budget exponent $\beta_{g(d)}$ is tied to the benchmark regime $g(d)$, while model-specific variation is captured by a scalar offset $\delta_m$.

### 6.1 From Matching Sets to Paths

We define a *cell* as a model–benchmark pair $(m, d)$, and let $\mathcal{B} = \{1, 2, 4, 8, 16\}$ be the allowed budget set. Let $\text{round}_\mathcal{B}$ denote rounding to the nearest allowed budget. For each cell and channel $\text{ch} \in \{P, Q\}$, SearchPath predicts the RL pass rate through a Base operating point:

$$\widehat{N}^{\text{ch}}_{m,d}(b) = \text{round}_\mathcal{B}\left(\alpha^{\text{ch}}_{m,d}\, b^{\beta^{\text{ch}}_{g(d)}}\right), \tag{7}$$

$$\widehat{\pi}^{\text{ch}}_{m,d} = \arg\min_\pi \mathcal{L}^{\text{ch}}_{m,d}(\pi), \tag{8}$$

$$\widehat{P}^{m,d}_{\text{RL}}(b) = P^{m,d}_{\text{Base}}\left(\widehat{\pi}, \widehat{N}(b)\right). \tag{9}$$

The selector loss is

$$\mathcal{L}^{\text{ch}}_{m,d}(\pi) = \sum_{b\in\mathcal{B}} \left\| r(\pi, \widehat{N}(b)) - \bar{r}_{\text{proto}} \right\|_W + \lambda_{\text{cost}}\, C(\pi). \tag{10}$$

Here $r = [\text{rank}_P, \text{rank}_{P-Q}, \text{rank}_Q, \text{ctrl}, \log \text{cost}]$, where ctrl is the controller index. We use $W = (1, 1, 1, 0.5, 0.2)$ and $\lambda_{\text{cost}} = 0.01$. The scale parameter is decomposed as $\log \alpha_{m,d} = \mu_{g(d)} + \delta_m$, with $\delta_m$ a model-specific scalar. Different experimental protocols instantiate $\delta_m$ differently; Section 6.3 uses this decomposition to separate direct transfer, one-cell calibration baselines, and oracle fits.

We measure rule quality by $\text{mech}_P \equiv |\widehat{P}_{\text{RL}} - P_{\text{RL}}| \cdot 100$ in percentage points. Because each RL target $P_{\text{RL}}(b)$ already has multiple Base+UDF matches within $\pm 3$ pp (Appendix A.3), the rule's role is path selection rather than match discovery.

### 6.2 Cross-Benchmark Transfer: An Empirical Budget Transition Rule

We first test whether $\beta$ is better explained by the model $m$ or by the benchmark regime $g(d)$. Table 2A fits $\beta$ on the anchor model under three policy constraints: R0 allows $\pi$ to vary across budgets, R1 locks one $\pi$ across all budgets, and R2 uses the default policy. Agreement across these settings indicates that the budget relation is not merely an artifact of per-budget policy switching.

This analysis yields three regimes. Math500 is sublinear and policy-sensitive, so we set $\beta_{\text{math}} = 0.60$. GPQA and IFEval are approximately linear,

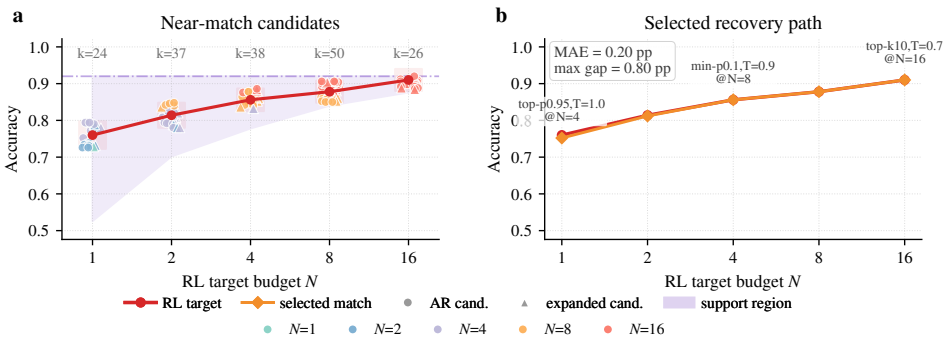


Figure 5: Equal-performance recovery on Qwen2.5-7B/Math500 under pass@$k$. **(A)** Base+UDF near-match candidates within tolerance of each RL@$N_{\mathrm{RL}}$ target. **(B)** The selected recovery path closely tracks the RL curve, with residual gaps shown in gray.

giving the shared nonmath/OOD regime $\beta_{\text{nonmath}} = 1.00$. AIME collapses to $\beta = 0$ when the policy is locked, indicating a floor-bound regime with $\beta_{\text{floor}} = 0.00$.

Table 2B evaluates this regime map against horizontal baselines. BOPTR-P1 with regime-specific $\beta$ is the strongest non-oracle rule, reducing the mean error from 6.01 pp for anchor per-budget copy to 3.41 pp. The lower error is associated with rescaling $N$ differently for OOD-nonmath cells, where directly copying the Math500 scaling would under-allocate the sample budget. The remaining gap to the oracle smooth path reflects the cost of predicting $(\alpha, \pi)$ rather than fitting them separately for each cell.

### 6.3 Cross-Model Transfer: Shared Form, Model-Specific Realization

We next test vertical transfer without target-RL information. Given the anchor $(\pi, N)$ path, Table 3A asks how well the rule predicts $P_{\mathrm{RL}}$ on new models. Direct anchor-path transfer works within the same Hard RL-training split at matched or comparable scale: the 7B, 14B, and Math-7B models obtain errors of 1.40, 2.55, and 3.73 pp. It degrades for the 1.5B downscale and for the cross-family Llama setting, where the error reaches 10.16 pp. This suggests that the shared RL-training distribution is important for direct $(\pi, N)$ transfer, even when the formula family remains useful.

**V3 as a low-dimensional offset baseline.** V3 estimates one scalar $\delta_m$ from each model's Math500 Base/RL pair and then reuses the same regime exponent and policy selector on the remaining benchmarks. As shown in Table 3B, this one-dimensional offset reduces the overall error from 5.36 to 4.44 pp. The improvement is especially pronounced in the lower cohort, where the mean error drops from 6.30 to 4.21 pp and the Llama error drops from 10.16 to 3.68 pp. This supports the decomposition $\log \alpha_{m,d} = \mu_{g(d)} + \delta_m$ as an informative structural form.

The remaining gap from V3 to the per-cell oracle is 1.7 pp. This residual reflects the cost of using one $\beta$ per regime and a behavior-coordinate selector instead of fitting each cell directly. Across the seven models, $\alpha_{\text{math}}$ spans a factor of 3.0 and the R1-locked $\pi_{\text{math}}$ varies by model (Appendix Table 9), but much of this variation is absorbed by $\delta_m$. Predicting $\delta_m$ from base-only statistics, without Math500 RL calibration, remains open.

### 6.4 Takeaways

BOPTR-P1 supports a hierarchical view of recovery. The formula family is shared, with $\beta$ determined by benchmark regime, but its realization is model-specific through $\alpha$ and the selected policy. Direct anchor-path transfer works best within the same SimpleRL-Zoo difficulty split and degrades across model families or training distributions. A one-dimensional offset absorbs much of this variation, reducing the overall error to 4.44 pp, but fully base-only prediction of that offset remains unresolved. Richer rule families that add free axes without matching training signal worsen transfer (Appendix A.4), and the strongest single base-only feature for $\delta_m$ remains within the $y$-shuffle null in the original six-model cohort (Appendix A.3.9). We therefore use BOPTR as a behavioral diagnostic for how RL relates to Base+UDF operating points under the SimpleRL-Zoo recipe, not as a universal post-RL performance predictor.

## 7 Discussion and Conclusion

Our results give a behavioral interpretation of the phrase "RL internalizes search." This does not

| benchmark | regime | R0 $\beta$ | R1 $\beta$ | R2 $\beta$ |
|---|---|---|---|---|
| Math500 | math | 0.50 | 0.70 | 0.50 |
| AIME | floor | 0.60 | 0.00 | 0.00 |
| GPQA | nonmath | 0.80 | 1.00 | 0.70 |
| IFEval | nonmath | 0.80 | 1.00 | 0.70 |

**(A)** Regime discovery. R0 allows $\pi$ to vary across budgets, R1 locks one $\pi$, and R2 uses the default $\pi$.

| baseline | Math500 | AIME | GPQA | IFEval | mean |
|---|---|---|---|---|---|
| H1 base greedy | 5.32 | — | 4.24 | 1.55 | 3.70 |
| H2 base same-$\pi$ | 12.52 | 5.00 | 8.69 | 6.21 | 8.10 |
| H3 anchor per-$b$ copy | 0.20 | 1.67 | 15.46 | 6.70 | 6.01 |
| H4 strict BOPTR ($\beta = 0.6$) | 1.12 | 1.00 | 17.17 | 6.95 | 6.56 |
| **H5 BOPTR-P1 (regime $\beta$)** | **1.12** | **2.67** | **6.77** | **3.11** | **3.41** |
| H6 2D oracle per-$b$ | 0.20 | 0.00 | 0.20 | 0.33 | 0.19 |
| H7 2D oracle smooth (R1 path) | 1.12 | 1.00 | 1.92 | 1.55 | 1.40 |

**(B)** Horizontal transfer. H5 is the strongest non-oracle rule.

Table 2: Regime discovery and horizontal transfer on Qwen2.5-7B. Errors are mean $|\widehat{P}_{\mathrm{RL}} - P_{\mathrm{RL}}|$ in percentage points over $b \in \{1, 2, 4, 8, 16\}$. BOPTR-P1 is the strongest non-oracle horizontal rule, outperforming both direct anchor copying and strict Math500 scaling.

*Upper: same Hard RL-training split*

| baseline | 7B | 14B | 1.5B | Math-7B | mean |
|---|---|---|---|---|---|
| H1 base greedy | 3.71 | 4.16 | 15.47 | 6.63 | 7.49 |
| H2 base same-$\pi$ | 8.10 | 8.29 | 7.07 | 7.18 | 7.66 |
| H3 anchor per-b copy | 6.01 | 6.92 | 14.24 | 7.58 | 8.69 |
| V1 per-cell oracle | 1.40 | 2.26 | 7.94 | 2.09 | 3.42 |
| V2 anchor path (no target-RL info) | 1.40 | 2.55 | 10.96 | 3.73 | 4.66 |

*Lower: cross-family / weak-capacity*

| baseline | 0.5B | Llama | Mistral | mean |
|---|---|---|---|---|
| H1 base greedy | 3.93 | 9.06 | 5.79 | 6.26 |
| H2 base same-$\pi$ | 7.73 | 8.37 | 8.09 | 8.06 |
| H3 anchor per-b copy | 8.32 | 9.79 | 9.44 | 9.18 |
| V1 per-cell oracle | 1.41 | 2.56 | 1.70 | 1.89 |
| V2 anchor path (no target-RL info) | 4.92 | 10.16 | 3.83 | 6.30 |

**(A)** Direct anchor-path vertical baselines without target-RL information. Each cell reports mean $\mathrm{mech}_P$ in percentage points over budgets.

| rule | structural assumption | overall mean (pp) |
|---|---|---|
| V1 per-cell oracle | none (fit $\alpha$, $\beta$, $\pi$ per cell) | 2.77 |
| V2 $\delta_m = 0$ | formula shared, no offset | 5.36 |
| **V3 $\delta_m$ from Math500** | formula + 1-D offset | **4.44** |

**(B)** Formula-decomposition ablation. V2→V3 measures the value of $\delta_m$ calibration; V3→V1 measures the residual structural cost.

Table 3: Vertical transfer and model-offset analysis across seven models. V2 tests direct anchor-path transfer without target-RL information, while V3 estimates a one-dimensional model offset $\delta_m$ from each model's Math500 Base/RL pair. V3 measures how much cross-model variation can be absorbed by this offset, not a main base-only transfer protocol.

mean that RL corresponds to a single named decoding policy, nor that the RL model literally implements an external search procedure. Rather, under the tested SimpleRL-Zoo recipe, many RL default-policy curves can be approximated by structured Base+UDF operating paths. BOPTR-P1 makes this relationship operational by selecting a Base tuple $(\widehat{\pi}, \widehat{N})$, where $\widehat{N}$ follows a regime-conditioned exponent and $\widehat{\pi}$ is chosen by behavior-coordinate matching.

This view clarifies what recovery can and cannot explain. A same-policy comparison may underestimate the relationship between Base and RL, because restricting evaluation to a single $(\pi_0, b)$ tuple hides nearby Base+UDF operating points that already match the RL target. When a smooth Base+UDF path tracks the RL curve, the corresponding gain can be read as improved access to behavior already present in the base operating landscape. When recovery fails, or when SC exposes a ceiling, the gap points to behavior not captured by the tested policy pool or by the selected behavioral channel. Thus, UDF is useful not only for matching RL behavior, but also for localizing where such matching breaks.

The resulting picture is hierarchical. The BOPTR formula family is shared across regimes, but its realization depends on both benchmark and model: task regime controls the budget exponent, while model-specific offsets and selected policies determine the concrete path. This also suggests how the analysis could extend beyond SimpleRL-Zoo. For another post-training recipe, the same operating-point view can ask which gains correspond to movement inside the base landscape and which require behavior outside the tested policy pool. However, regime exponents and model offsets should be re-estimated rather than assumed to transfer unchanged.

**Conclusion.** We introduced SearchLens/UDF as a unified behavioral map for decoding and search, and SearchPath/BOPTR as a low-dimensional rule for tracing RL behavior through this map. Our findings support a qualified internalized-search view: RL often improves access to behaviors that are already reachable from the base model under suitable inference-time policies, but the concrete recovery path is benchmark- and model-specific. Cross-family degradation and the unresolved prediction of $\delta_m$ delimit this interpretation: BOPTR is best used as a diagnostic of RL–Base relationships, not as a universal post-RL performance predictor.

## Limitations

**Behavioral, not parameter-level, equivalence.** Our recoverability and rule results are behavioral statements about the operating-point landscape: they show that Base+UDF contains operating points whose pass-rates and concentration match RL targets, and that a small low-degree rule selects such points. This is not a claim that the RL model literally executes any external decoding policy; we make no parameter-level mechanistic identification.

**Policy-pool comparability.** Some settings expose AR-only pools while others include expanded controllers (MCMC-like resampling, entropy branching, tree controllers). Recoverability statements should be interpreted relative to the UDF policies actually evaluated in each cell; we mark AR-only and expanded-pool cells separately and never compare envelopes across mismatched pools.

**Metric heterogeneity across benchmarks.** Math500, AIME, GPQA, and IFEval do not all admit the same evaluation-metric semantics. IFEval is treated as an evaluation-metric mismatch / OOD stress test, with first-finish and validity as primary metrics and SC as diagnostic; GPQA often behaves as a sharpening/concentration task rather than a support-expansion task. Cross-benchmark numbers should be read with this regime structure in mind, not as a uniform metric.

**Model-specific parameters.** The BOPTR formula family is shared, but its parameters and selected policies are model-specific. In our analysis we estimate the per-model offset $\delta_m$ from each model's Math500 Base/RL pair (V3) as a one-cell calibration baseline that bounds how much of the cross-model gap is absorbable by a low-dimensional offset. Predicting $\delta_m$ from base-only statistics, without any RL-side calibration, is currently unresolved ($n$=6 models leaves the LOOCV inside the $y$-shuffle null). We position the rule as a behavioral diagnostic under the tested SimpleRL-Zoo recipe, not as a universal post-RL performance predictor; stronger absolute-prediction or checkpoint-selection claims would require larger model cohorts, training-distribution calibration sets, or model-conditioned adapters beyond the current rule family.

**Scope of model coverage.** Results cover one Qwen-family scale sweep (0.5B/1.5B/7B/14B), one same Hard RL-training split cross-family check (Qwen2.5-Math-7B), one cross-family check (Llama-3.1-8B), and one weak-capacity check (Mistral-7B-v0.1) on paired Base/RL checkpoints from a single RL recipe (SimpleRL-Zoo). We do not claim our regime map ($\beta_{\text{math}}$=0.6, $\beta_{\text{nonmath}}$=1.0, $\beta_{\text{floor}}$=0) transfers verbatim to other RL recipes or to substantially different model families without re-anchoring.

## References


OpenAI. 2024. Learning to Reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Daya Guo et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Zhihong Shao et al. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300.

Weihao Zeng, Yuzhen Huang, and colleagues. 2025. SimpleRL-Zoo: Investigating and Taming Zero Reinforcement Learning for Open Base Models in the Wild. arXiv:2503.18892.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? arXiv:2504.13837.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. 2025. Echo Chamber: RL Post-training Amplifies Behaviors Learned in Pretraining. arXiv:2504.07912.

Aayush Karan and Yilun Du. 2025. Reasoning with Sampling: Your Base Model is Smarter Than You Think. arXiv:2510.14901.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. arXiv:2501.17161.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations (ICLR)*.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc Le, Christopher Re, and Azalia Mirhoseini. 2024. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. arXiv:2407.21787.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. In *AAAI Conference on Artificial Intelligence*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with Language Model is Planning with World Model. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024. Language Agent Tree Search Unifies Reasoning, Acting, and Planning in Language Models. In *International Conference on Machine Learning (ICML)*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's Verify Step by Step. In *International Conference on Learning Representations (ICLR)*.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Sridhar Thiagarajan, Craig Boutilier, Rishabh Agarwal, Aviral Kumar, and Aleksandra Faust. 2025. Inference-Aware Fine-Tuning for Best-of-N Sampling in Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. 2023. Measuring Faithfulness in Chain-of-Thought Reasoning. arXiv:2307.13702.

Alon Jacovi and Yoav Goldberg. 2020. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving with the MATH Dataset. In *NeurIPS Datasets and Benchmarks Track*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. arXiv:2311.12022.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-Following Evaluation for Large Language Models. arXiv:2311.07911.

# A Appendix

## A.1 UDF expressiveness: equivalence to mainstream decoding algorithms

To verify that UDF is not just a notational wrapper, we instantiate eleven mainstream decoding algorithms as UDF operating points and check that they reproduce reference implementations on the same prompts (Math500 subset, $n$=50, Qwen2.5-7B, identical seed). Tier 1–2 covers the vLLM `SamplingParams` family (greedy, temperature, top-$k$, top-$p$, min-$p$, and two repetition/frequency-penalty variants); Tier 3–4 covers external reference implementations (HuggingFace `typical_p`, HuggingFace `num_beams`, the official MCMC `power_sampling`, and an internal `entropy_tree` branching reference). All Tier 1–2 strategies match their references with output match rate $\geq 0.76$ and EM agreement $\geq 0.92$ and are flagged equivalent under the pre-registered threshold (Table 4). Tier 3–4 strategies match within $|\Delta\text{EM}| \leq 0.08$ (Table 5). This establishes that the five-component UDF parameterization is empirically expressive enough to recover the decoding policies used in published baselines, so that downstream same-policy and recovery analyses operate on an instantiation-faithful policy space.

| Strategy | Output match | EM agree | Equiv. |
|---|---|---|---|
| greedy | 0.86 | 0.92 | ✓ |
| temperature | 0.98 | 1.00 | ✓ |
| top-$k$ | 0.92 | 0.96 | ✓ |
| top-$p$ | 0.98 | 1.00 | ✓ |
| min-$p$ | 0.92 | 1.00 | ✓ |
| temp + rep. penalty | 0.76 | 0.98 | ✓ |
| top-$p$ + freq. penalty | 0.86 | 0.98 | ✓ |

Table 4: Tier 1–2 equivalence: UDF instantiation vs. vLLM `SamplingParams` reference on Math500 ($n$=50, Qwen2.5-7B). Output match is the per-prompt string-match rate; EM agree is the agreement of extracted answers. All seven strategies pass the pre-registered equivalence threshold.

| Algorithm | Reference | EM ref | EM UDF | $\lvert\Delta\rvert$ |
|---|---|---|---|---|
| typical | HF `typical_p` | 0.36 | 0.44 | 0.08 |
| beam search | HF `num_beams` | 0.52 | 0.50 | 0.02 |
| power sampling | Official MCMC `power_samp` | 0.50 | 0.48 | 0.02 |
| entropy branching | Internal `entropy_tree` ref | 0.40 | 0.40 | 0.00 |

Table 5: Tier 3–4 equivalence: UDF instantiation vs. external reference implementations on Math500 ($n$=50, Qwen2.5-7B). $|\Delta|$ is $|\text{EM}_{\text{ref}} - \text{EM}_{\text{UDF}}|$; for `power_sampling`, the reference MCMC accept ratio is 0.657; for `entropy_tree`, output match rate $= 1.00$ (token-identical). All four match within $|\Delta\text{EM}| \leq 0.08$.

## A.2 Same-policy and Recovery supplements

### A.2.1 Qualitative same-policy signatures

Table 6 catalogues the four qualitative same-policy signatures used throughout the analysis.

| Pattern | Typical metric | Example block | Interpretation |
|---|---|---|---|
| Low-budget RL gain | pass@k, SC | Math500 / AIME | Sampling efficiency improves. |
| Base high-budget recovery | pass@k | GPQA / some math settings | Diversity/support remains in base. |
| Concentration gain | SC or SC / pass@k | GPQA | RL sharpens answer distribution. |
| Termination gain | FFS / validity | IFEval / long outputs | RL improves completion reliability. |

Table 6: Qualitative same-policy signatures used throughout the analysis.

### A.2.2 Recovery analysis objects

Table 7 lists the four recovery objects (envelope / near-match set / smooth path / frozen rule), what each one is allowed to show, and its main risk.

### A.2.3 Same-policy SC companion

Figure 6 is the self-consistency (`majority_vote@k`) companion of main Figure 3.

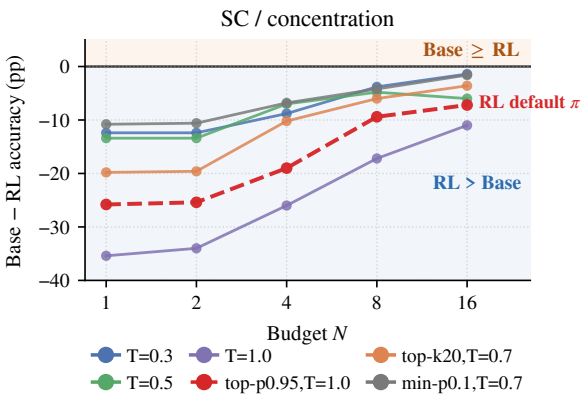


Figure 6: **Same-policy Base vs RL gap, Qwen2.5-7B / Math500 (SC).** Self-consistency (`majority_vote@k`) companion of Figure 3. Each line fixes one decoding policy $\pi$ and plots $A_{\text{Base}}(\pi, N) - A_{\text{RL}}(\pi, N)$ (pp).

### A.2.4 Support-region SC companion

Figure 7 is the self-consistency (`majority_vote@k`) companion of main Figure 4.

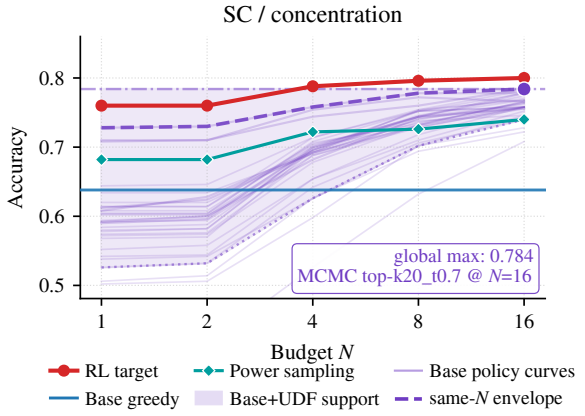


Figure 7: **Per-budget Base+UDF support region, Qwen2.5-7B / Math500 (SC).** Self-consistency (`majority_vote@k`) companion of Figure 4. Convention as in the main panel; the RL target remains inside the Base+UDF support region at every budget.

### A.2.5 Full same-policy gap map

Figure 8 extends main Figure 3 to every base-shared decoding policy (23 policies, grouped by family) at every budget.

### A.2.6 UDF sweep compute cost on the anchor cell

Table 8 reports the wall-clock and token cost of producing the Qwen2.5-7B/Math500 UDF sweep used in Section 5. We separately tally three channels: (i) **AR-Base** on the 23 base-shared policies plus the RL default `topp0.95_t1.0` (24 policies total) under the base model; (ii) **AR-RL** on the same 24 policies under the RL model; and (iii) **Expanded base**, the cross-family controllers entered in the recovery near-match search (entropy_branch with $b$=4 tree, sequence MCMC with $b$=4, MCTS with 16 iterations, and thought tree with $b \in \{4, 8\}$). All runs use vLLM with tensor-parallel $= 2$ on H100s; GPU-hours below are wall-hours $\times 2$.

**Amortization across budgets and metrics.** Each policy is generated once at the largest budget ($b$=16) and the same sample pool is then evaluated at every smaller budget $b \in \{1, 2, 4, 8\}$ and at all four metrics (pass@$k$, best-of-$n$, majority-vote, first-finish). We therefore deduplicate at the generation-run level (one row per AR policy; one row per (controller-mode, sampling-policy, internal-budget) for expanded controllers) and report the maximum wall-time within each group as the true compute cost. The 5 budgets $\times$ 4 metrics per AR policy and the additional metric/budget combinations per expanded controller appear in the raw `strategy_sweep.json` entries but do not add wall-clock cost beyond the single generation.

**Headline numbers.** The 24-policy AR sweep is cheap on both the base ($\approx$ 2.3 wall-h, $\approx$ 124M tokens) and the RL model ($\approx$ 2.1 wall-h, $\approx$ 122M tokens); roughly 4.6 and 4.1 H100 GPU-hours respectively under TP$=$ 2. Expanded controllers dominate the budget: 8 deduped (controller, sampling-policy, internal-budget) runs consume $\approx$ 226 wall-h ($\approx$ 452 GPU-hours), $\approx$ 100$\times$ the AR sweep, because each expanded sample is an internal search tree of multiple raw generations. Across all three channels the anchor cell consumes $\approx$ 230 wall-hours / $\approx$ 461 H100 GPU-hours / $\approx$ 0.31B tokens, which sets the per-cell cost we extrapolate from in the cross-model BOPTR study (Sections 6.2–6.3).

**What this excludes.** The numbers above cover only the canonical post-paper sweeps (`sweep_{base,rl}_full` plus the two `f2_*_base_full` expanded sweeps). Exploratory smaller-question runs (`f0_*`, `f1_*` validation shards) and shard-level partial sweeps that were superseded by the canonical files are not included; they would roughly double the figure. RL training cost itself is also excluded – only the post-training UDF behavioral sweep is reported.

| Analysis object | Uses RL target? | Allows lookup? | Purpose | Main risk |
|---|---|---|---|---|
| Envelope | no per-policy target | yes | Upper-bound support | Overstates practical policy. |
| Near-match set | yes | yes | Recoverability | Too many matches. |
| Smooth path | yes | constrained | Structured recovery | Hyperparameter sensitivity. |
| Frozen rule | train anchor only | no | Transfer diagnostic | Under-identification. |

Table 7: Recovery analyses and what each one is allowed to show.

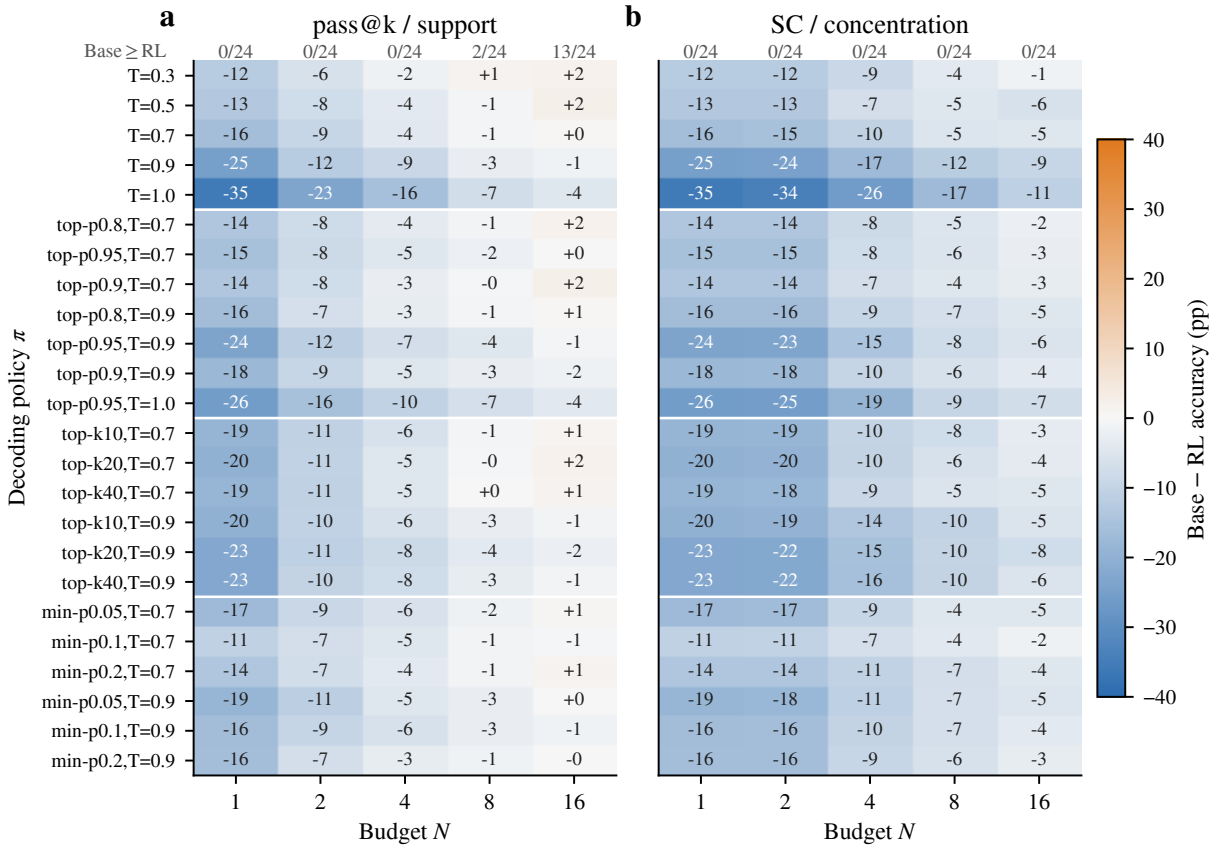


Figure 8: **Full same-policy gap map, Qwen2.5-7B / Math500.** Each cell shows $A_{\text{Base}}(\pi, N) - A_{\text{RL}}(\pi, N)$ (percentage points) at the same policy $\pi$ and budget $N$. Rows are the 23 base-shared decoding policies, grouped by family (temperature, top-p, top-k, min-p; families separated by white horizontal lines and sorted within each family by the relevant hyperparameter). Columns are budgets $N \in \{1, 2, 4, 8, 16\}$. **(a)** `pass@k`. **(b)** self-consistency (`majority_vote@k`). Blue cells: RL exceeds Base at the same $(\pi, N)$. Orange cells: Base $\geq$ RL. The strip below each panel reports, at each budget, the number of policies with Base $\geq$ RL. The diverging color scale is symmetric around zero; cell numbers are the gap rounded to the nearest percentage point. Companion to main Figure 3.

| channel | #runs | wall-h | GPU-h(TP2) | tokens(M) |
|---|---|---|---|---|
| AR-Base (24 policies) | 24 | 2.30 | 4.61 | 123.8 |
| AR-RL (24 policies) | 24 | 2.05 | 4.11 | 122.1 |
| Expanded base (dedup) | 8 | 225.93 | 451.87 | 65.9 |
| *Expanded breakdown by controller family* | | | | |
| EntTree ($b$=4) | 1 | 42.11 | 84.21 | 9.5 |
| MCMC ($b$=4, 3 samplers) | 3 | 63.58 | 127.17 | 17.8 |
| MCTS (16 iters, 2 samplers) | 2 | 63.38 | 126.76 | 19.6 |
| Tree ($b \in \{4, 8\}$) | 2 | 56.86 | 113.73 | 18.9 |
| **Anchor-cell total** | **56** | **230.29** | **460.58** | **311.8** |

Table 8: UDF sweep compute cost on the Qwen2.5-7B / Math500 anchor cell, deduplicated at the generation-run level (the same 16-sample pool is reused across budgets $\{1, 2, 4, 8, 16\}$ and metrics $\{\text{pass@}k$, best-of-$n$, majority-vote, first-finish$\}$). GPU-hours assume vLLM with tensor-parallel $= 2$ on H100s. AR sweeps cover the 23 base-shared policies plus the RL default `topp0.95_t1.0`. Expanded controllers are the cross-family entries used in the recovery near-match search (Section 5.2, Fig. 5).

## A.3 BOPTR supplements

### A.3.1 Anchor matching set

### A.3.2 Benchmark-regime $\beta$ visualization

### A.3.3 Horizontal-transfer bar chart

### A.3.4 Vertical-transfer heatmap

### A.3.5 Formula-same, parameters-different

Across the seven models the BOPTR-P1 rule keeps its structural form (regime $\beta$ shared, selector procedure shared) while two per-model knobs absorb individual differences: a continuous scale $\alpha_{\text{math}}$ and a discrete locked policy $\pi_{\text{math}}$ (Table 9). The scale spans $3.0\times$ (2.30–6.96), and the locked $\pi_{\text{math}}$ varies across models (`topp`, `minp`, and pure temperature all appear). This is the empirical content of "formula-same, parameters-different": the rule transfers, but the per-cell tuple $(\alpha, \pi)$ does not, and trying to copy any single anchor policy across models would lose the $\pi$ component every time. Figure 13 visualizes the same fact.

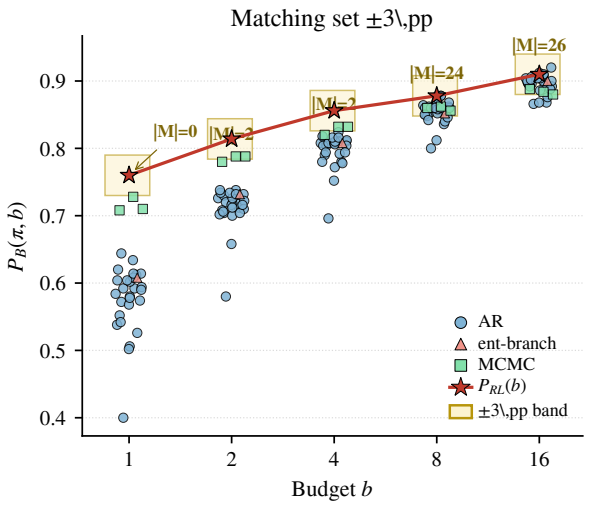


Figure 9: Anchor cell (Qwen2.5-7B/Math500): every RL operating point $P_{\mathrm{RL}}(b)$ is matched within $\pm 3$ pp by multiple Base+UDF $(\pi, N)$ points. The matching set is rich, so the rule's job is to pick a budget-monotone path, not to find scarce structure. Supplement to Section 6.1.

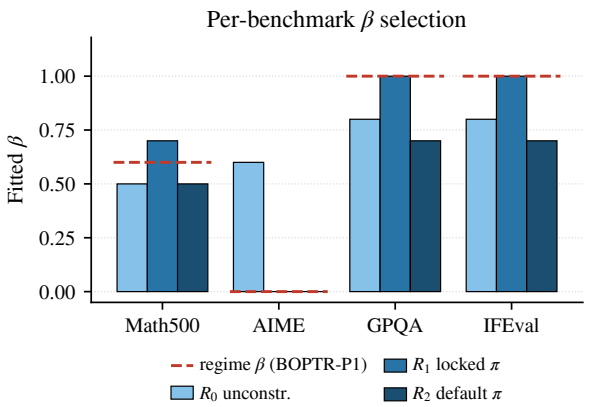


Figure 10: Per-benchmark R0/R1/R2 $\beta$ fits. The benchmark splits into three regimes — math ($\beta = 0.6$), non-math OOD ($\beta = 1.0$), and floor ($\beta = 0$) — not a universal exponent. Visualization of main Table 2A.

### A.3.6 Full BOPTR-P1 per-cell results (23 cells)

Table 10 reports the per-cell BOPTR-P1 rule parameters (regime $\beta$, anchor-calibrated $\alpha$, selected $\widehat{\pi}$) and the resulting $\mathrm{mech}_P$ per cell, with the cell-fit reference column. The anchor-vs-cell-fit gap is the transfer cost of replacing per-cell $\alpha$ fits with the 1-D $\delta_m$ calibration.

The gap column makes the transfer cost concrete. Most cells transfer at zero cost: 16/23 cells show gap $= +0.00$, meaning the anchor-calibrated $\alpha$ already matches the per-cell fit. Three cells absorb a moderate cost (qwen25_7b/aime $+1.67$, qwen25_1.5b/gpqa $+1.41$, mistral7b_v01/ifeval $+2.11$), and the two outliers — qwen25_1.5b/ifeval ($+6.10$) and mistral7b_v01/gpqa ($-5.45$, where the rule *beats* the cell fit) — both involve the smallest two models on OOD-nonmath benchmarks, where the cell-fit $\alpha$ is itself unstable. The negative gap for mistral7b_v01/gpqa is a reminder that the per-cell oracle is not a strict lower bound when the cell fit overfits five budgets with two free parameters.

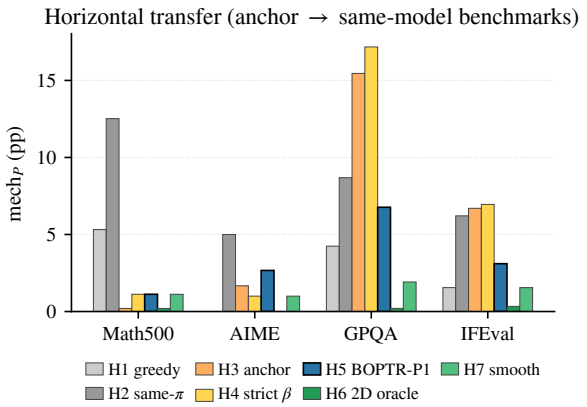


Figure 11: Horizontal-transfer baseline comparison (anchor=Qwen2.5-7B/Math500 → other benchmarks). Bars are row-mean $\mathrm{mech}_P$ over four benchmarks; H5 (regime $\beta$) is the lowest non-oracle. Visualization of main Table 2B.

| model | $\alpha_{\mathrm{math}}$ | $\pi_{\mathrm{math}}$ (R1 locked) |
|---|---|---|
| Qwen2.5-7B | 2.64 | `topp0.8_t0.9` |
| Qwen2.5-14B | 3.48 | `minp0.2_t0.9` |
| Qwen2.5-1.5B | 6.96 | `temp_0.3` |
| Qwen2.5-0.5B | 3.03 | `minp0.05_t0.7` |
| Qwen2.5-Math-7B | 2.64 | `minp0.05_t0.7` |
| Llama-3.1-8B | 3.03 | `temp_0.5` |
| Mistral-7B-v0.1 | 2.30 | `topp0.9_t0.7` |

Table 9: Formula-same, parameters-different. All seven models share the same rule with regime $\beta$ fixed; only $\alpha$ and the locked $\pi$ change per cell. $\alpha_{\mathrm{math}}$ spans $3.0\times$; $\pi_{\mathrm{math}}$ varies across models — no transferable single policy, only a transferable selection procedure. $\alpha_{\mathrm{nonmath}} = 1.0$ for every model by construction ($\beta = 1$ leaves $\alpha$ no degrees of freedom).

### A.3.7 Ablation study: selector, $\delta_m$, and $\widehat{N}(b)$

We ablate the three structural ingredients of BOPTR-P1 (Table 11): the behavior-coordinate $\pi$ selector (A1 replaces $\widehat{\pi}$ with the cell's RL default), the per-model $\delta_m$ offset (A2 sets $\delta_m = 0$), and the predicted $\widehat{N}(b)$ (A3 fixes $N = b$). Both 1-D and 2-D oracles bound the rule from below.

The largest single drop is A1 ($+2.76$ pp on the 23-cell mean), confirming that the behavior-coordinate selector is the load-bearing ingredient and that copying the RL default policy back into a base evaluation undershoots its effective sample efficiency. A3 (no $\widehat{N}(b)$ rule, $+1.17$ pp) shows the

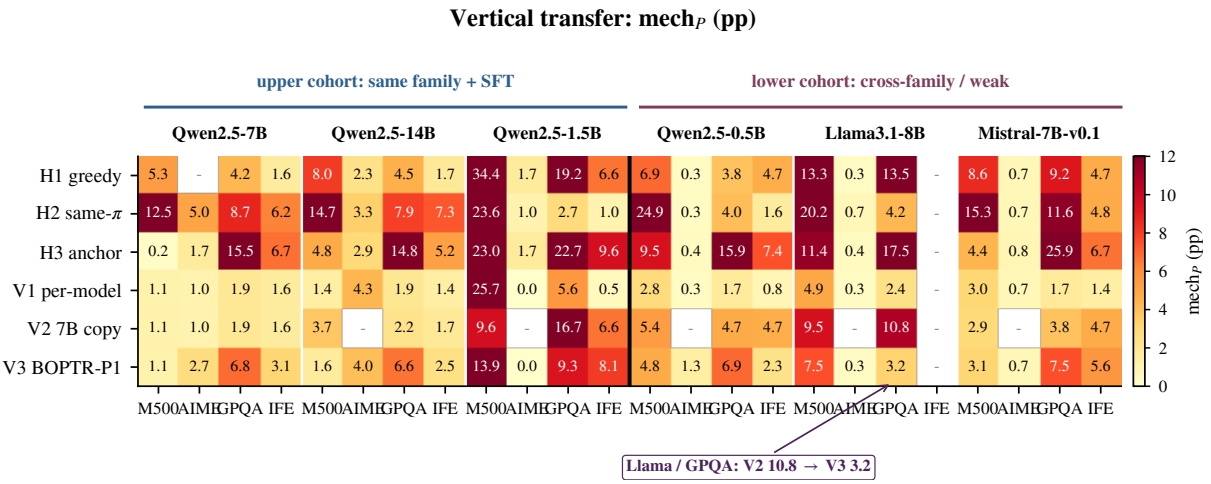


Figure 12: Vertical-transfer heatmap (7 models × 4 benchmarks). V2 (anchor path copy, no target-RL information) is competitive in the upper same Hard RL-training split cohort (including Qwen2.5-Math-7B) but breaks down on the cross-family lower cohort. Visualization of main Table 3A.

| model | benchmark | regime | $\beta$ | $\widehat{\alpha}$ | $\widehat{\pi}$ | $\text{mech}_P^{\text{cell}}$ | $\text{mech}_P^{\text{anchor}}$ | gap |
|---|---|---|---|---|---|---|---|---|
| qwen25_0.5b | math500 | math | 0.6 | 3.03 | `topp0.8_t0.9` | 4.80 | 4.80 | +0.00 |
| qwen25_0.5b | aime | floor | 0.0 | 1.82 | `minp0.1_t0.9` | 0.33 | 1.33 | +1.00 |
| qwen25_0.5b | gpqa | nonmath | 1.0 | 0.91 | `topk10_t0.9` | 6.87 | 6.87 | +0.00 |
| qwen25_0.5b | ifeval | nonmath | 1.0 | 0.91 | `topp0.9_t0.9` | 2.33 | 2.33 | +0.00 |
| qwen25_1.5b | math500 | math | 0.6 | 6.96 | `topk20_t0.9` | 13.92 | 13.92 | +0.00 |
| qwen25_1.5b | aime | floor | 0.0 | 4.19 | `topp0.9_t0.7` | 1.67 | 0.00 | −1.67 |
| qwen25_1.5b | gpqa | nonmath | 1.0 | 2.10 | `topk20_t0.9` | 7.88 | 9.29 | +1.41 |
| qwen25_1.5b | ifeval | nonmath | 1.0 | 2.10 | `topk40_t0.7` | 1.96 | 8.06 | +6.10 |
| qwen25_7b | math500 | math | 0.6 | 2.64 | `topp0.8_t0.9` | 1.12 | 1.12 | +0.00 |
| qwen25_7b | aime | floor | 0.0 | 1.59 | `topk20_t0.7` | 1.00 | 2.67 | +1.67 |
| qwen25_7b | gpqa | nonmath | 1.0 | 0.79 | `minp0.05_t0.9` | 6.77 | 6.77 | +0.00 |
| qwen25_7b | ifeval | nonmath | 1.0 | 0.79 | `topk10_t0.7` | 3.11 | 3.11 | +0.00 |
| qwen25_14b | math500 | math | 0.6 | 3.48 | `topp0.8_t0.9` | 1.64 | 1.64 | +0.00 |
| qwen25_14b | aime | floor | 0.0 | 2.10 | `topp0.9_t0.9` | 4.33 | 4.00 | −0.33 |
| qwen25_14b | gpqa | nonmath | 1.0 | 1.05 | `topk20_t0.7` | 6.57 | 6.57 | +0.00 |
| qwen25_14b | ifeval | nonmath | 1.0 | 1.05 | `topk40_t0.7` | 2.48 | 2.48 | +0.00 |
| llama31_8b | math500 | math | 0.6 | 3.03 | `minp0.1_t0.7` | 7.48 | 7.48 | +0.00 |
| llama31_8b | aime | floor | 0.0 | 1.82 | `topp0.95_t1.0` | 0.33 | 0.33 | +0.00 |
| llama31_8b | gpqa | nonmath | 1.0 | 0.91 | `topk20_t0.9` | 3.23 | 3.23 | +0.00 |
| mistral7b_v01 | math500 | math | 0.6 | 2.30 | `topp0.95_t0.7` | 3.08 | 3.08 | +0.00 |
| mistral7b_v01 | aime | floor | 0.0 | 1.38 | `topp0.95_t1.0` | 0.67 | 0.67 | +0.00 |
| mistral7b_v01 | gpqa | nonmath | 1.0 | 0.69 | `topp0.95_t0.9` | 12.93 | 7.47 | −5.45 |
| mistral7b_v01 | ifeval | nonmath | 1.0 | 0.69 | `topp0.95_t0.7` | 3.48 | 5.58 | +2.11 |

Table 10: Full BOPTR-P1 P-channel per-cell results (mean $\text{mech}_P$ over budgets, pp). "cell" = per-cell $\alpha$ fit (oracle upper bound on the rule); "anchor" = anchor-calibrated $\alpha$ ($\mu_g + \delta_m$, 1-D model offset). Transfer gap is anchor−cell. ifeval missing for llama31_8b.

budget exponent does real work on math (+5.5 pp on math regime alone) but adds nothing on non-math/floor regimes whose $\beta \in \{1, 0\}$ makes $\widehat{N}(b)$ a no-op anyway. A2 (no $\delta_m$) is the most subtle: it is slightly better than MAIN on the 23-cell flat mean (−0.31 pp), but its real role is the cross-family rescue in Table 3B where Llama drops from 10.16 to 3.68 pp. This is consistent with the $\delta_m$ story: a per-model offset matters precisely where a single anchor's scale is wrong.

### A.3.8 Q channel: rule + recoverability

Table 12 reports the Q-channel (self-consistency) BOPTR rule and the cell-level recoverability flag. The Q channel uses $\widehat{\pi}^Q = \arg\max_\pi [Q_B(\pi, 16) - 0.5\,(P_B - Q_B) - 0.01\,C]$ and $\beta^Q = 0$ outside GPQA (which is the only regime where Q scales with budget). 19/23 cells are recoverable within 0.05 of max $Q_{\text{RL}}$; the four unrecoverable cells are all (model, Math500) where RL has discovered a sharper distribution than any base policy at any budget — this is the SC-side analog of the same-policy gap in Section 5.2.

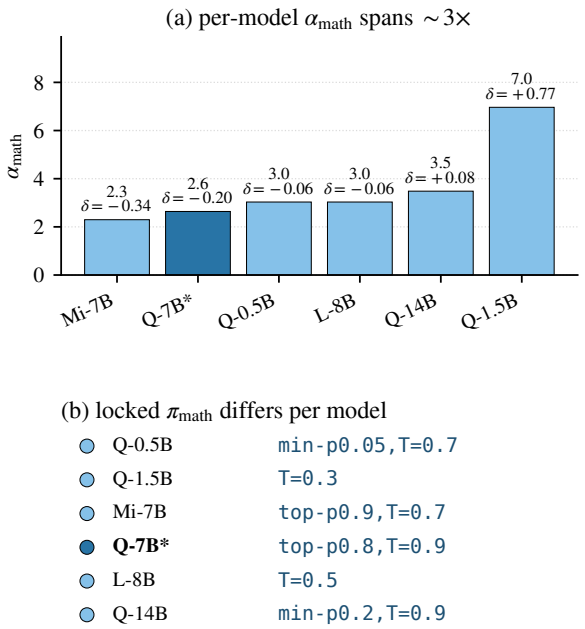


Figure 13: Parameters differ but the formula is universal: $\alpha_{\text{math}}$ spans 3.0× across the seven models, and each model picks its own R1-locked $\pi_{\text{math}}$. BOPTR absorbs all of this into the single scalar $\delta_m$.

| variant | math | nonmath | floor | 23-cell mean |
|---|---|---|---|---|
| **MAIN (BOPTR-P1)** | **5.34** | **5.61** | **1.50** | **4.47** |
| A1 no $\widehat{\pi}$ selector | 12.11 | 7.20 | 2.39 | 7.23 |
| A2 no $\delta_m$ offset | 4.78 | 5.24 | 1.56 | 4.16 |
| A3 no $\widehat{N}(b)$ rule | 10.85 | 5.24 | 1.17 | 5.64 |
| ORACLE-2D per-$b$ $(\pi, N)$ | 1.59 | 0.39 | 0.11 | 0.63 |
| ORACLE-1D per-$b$ $(\pi, N=b)$ | 7.18 | 0.54 | 0.17 | 2.17 |

Table 11: BOPTR-P1 ablation, regime-aggregated mean $\text{mech}_P$ (pp). A1 (no selector) is the largest drop on math (+6.8 pp). A3 (no compute rule) hurts most on math (+5.5 pp). A2 is slightly *better* than MAIN at the 23-cell mean (−0.31 pp); its real role (Table 3B) is closing the cross-family gap in the lower half.

The four unrecoverable Math500 cells span 0.5B/1.5B/14B Qwen and Llama-3.1-8B; the 7B and Mistral-7B Math500 cells *are* recoverable, so unrecoverability is not monotone in capacity. All four use the same $\widehat{\alpha}^Q = 16$ (the budget cap), reflecting that the Q-channel selector pushes to the maximum allowed sample budget when no base policy reaches RL's SC concentration. Floor regime is absent here because AIME does not produce a stable Q-channel signal at our scale; the table therefore covers Math500 (math) and GPQA (nonmath) only.

### A.3.9 V4 honest negative: base-only prediction of $\alpha_{\text{math}}$

A natural follow-up is whether the per-model offset $\delta_m$ itself can be predicted from base-only statistics — i.e. without any RL data at all. Table 13 and

| model | benchmark | recov | $\beta^Q$ | $\widehat{\alpha}^Q$ | $\text{mech}_Q$ (pp) |
|---|---|---|---|---|---|
| qwen25_0.5b | math500 | × | 0.0 | 16.0 | 5.32 |
| qwen25_0.5b | gpqa | ✓ | 0.6 | 0.44 | 2.22 |
| qwen25_1.5b | math500 | × | 0.0 | 16.0 | 11.08 |
| qwen25_1.5b | gpqa | ✓ | 0.6 | 1.74 | 6.16 |
| qwen25_7b | math500 | ✓ | 0.0 | 16.0 | 1.64 |
| qwen25_7b | gpqa | ✓ | 0.6 | 3.48 | 3.13 |
| qwen25_14b | math500 | × | 0.0 | 16.0 | 4.20 |
| qwen25_14b | gpqa | ✓ | 0.6 | 3.48 | 2.53 |
| llama31_8b | math500 | × | 0.0 | 16.0 | 4.72 |
| llama31_8b | gpqa | ✓ | 0.6 | 1.74 | 4.44 |
| mistral7b_v01 | math500 | ✓ | 0.0 | 16.0 | 2.40 |
| mistral7b_v01 | gpqa | ✓ | 0.6 | 1.74 | 4.04 |

*Recoverability:* 19/23 cells (83%).
*Unrecoverable:* 4 math500 cells.

Table 12: BOPTR Q-channel rule + recoverability. Selected cells; full 23-cell results in supplementary CSV. Mean $\text{mech}_Q$ (recoverable only): math 2.02 pp, nonmath 2.76 pp, floor 0.44 pp.

Figure 14 report a leave-one-model-out ridge of $\log \alpha_{\text{math}}$ on 24 cross-benchmark base features (six per benchmark) plus $\log(\text{model_size})$. With only 6 models, the top single feature (`gpqa__P0_b16`) achieves LOO pseudo-$R^2 = -0.38$ with empirical $p \approx 0.21$ over 100 $y$-shuffles — well inside the null. Multi-feature ridge does not help ($k=1$ gives LOO pseudo-$R^2 = -0.43$; $k = 2$ gives $-0.42$; $k = 3$ gives $-0.43$; ridge $\lambda = 1.0$). We treat this as a *negative result*, not noise: predicting $\delta_m$ from base structure will require either many more models or rltrain-side features (gradients, layer responses) — both left to future work.

The feature ranking is itself informative. All five top features are accuracy- or entropy-style metrics on *other* benchmarks (GPQA and Math500 base statistics), not the within-benchmark Math500 base curve — consistent with the BOPTR claim that $\delta_m$ is a cross-benchmark property of the model, not a per-cell artifact. The negative $r$ for `gpqa__P0_b16` means models with stronger base GPQA accuracy tend to need *smaller* $\alpha_{\text{math}}$, which is mechanistically sensible (a model that already concentrates on the right answers at $b=16$ does not benefit from large effective sample budgets). But with $n=6$ the in-sample $|r|=0.79$ does not survive LOO, and we report this honestly: the rule's per-model component is not yet predictable from base statistics at the scales we cover.

## A.4 Rule variants and reviewer checklists

### A.4.1 Rule variants

Table 14 positions the rule families we considered against the v5 BOPTR-P1 main diagnostic. The

| feature (top by $\lvert r \rvert$) | Pearson $r$ | LOO pseudo-$R^2$ |
|---|---|---|
| `gpqa__P0_b16` | $-0.79$ | $-0.38$ |
| `gpqa__Pmax_b16` | $-0.77$ | $-0.39$ |
| `gpqa__Dp` | $+0.64$ | $-0.46$ |
| `math500__Dp` | $+0.59$ | $-0.44$ |
| `math500__Hp` | $+0.51$ | $-0.44$ |
| *Null (100 y-shuffles, top-1):* $5\%/50\%/95\% = -0.60/-0.46/-0.32$; $p \approx 0.21$ | | |

Table 13: V4 honest-negative LOOCV ridge of $\log \alpha_{\text{math}}$ on base-only features ($n = 6$). In-sample $\lvert r \rvert$ reaches 0.79 but every LOO pseudo-$R^2$ is negative; top score sits inside the $y$-shuffle null.

two oracles bound recoverability from above (per-budget oracle) and characterize whether the recovery is coherent across budgets (smooth path). v5 keeps a low-dimensional state $[P, Q, T]$ and is the rule used throughout the main paper; v6 adds a signed sharpening-vs-expansion axis and helps on compression-dominated cells but hurts the expansion cells, demonstrating that one extra free axis is not free. v7 fully decouples support and concentration with separate adapters; with a single anchor it under-identifies and worsens transfer everywhere — this is the negative ablation cited in Section 6.4. v8-H is the planned next-step rule family: same low-dim state as v5 but with a small adapter for the model-specific component currently absorbed by $\delta_m$.

**Why these six and not more.** The two oracles (per-budget, smooth-path) are not rules we would deploy — they refit $\widehat{\pi}$ or $\widehat{N}$ per cell — but they delimit the recoverability ceiling. Per-budget oracle says "the support region contains a matching $(\pi, N)$ at every $b$," and smooth-path says "those matches connect into a budget-coherent trajectory rather than five disjoint lookups." v5 (BOPTR-P1) is the smallest rule that respects both constraints from a single calibration cell, which is why we adopt it as the main diagnostic. v6 and v7 are deliberate ablations: each adds one degree of freedom (signed sharpening axis for v6, independent support/concentration heads for v7) and each loses ground precisely where the extra parameter cannot be re-identified from one anchor. v8-H is listed as planned, not evaluated — it replaces the per-model $\delta_m$ scalar with a small base-feature adapter, which the V4 negative (Section 6.4) shows requires either more models or rltrain-side features to identify reliably.

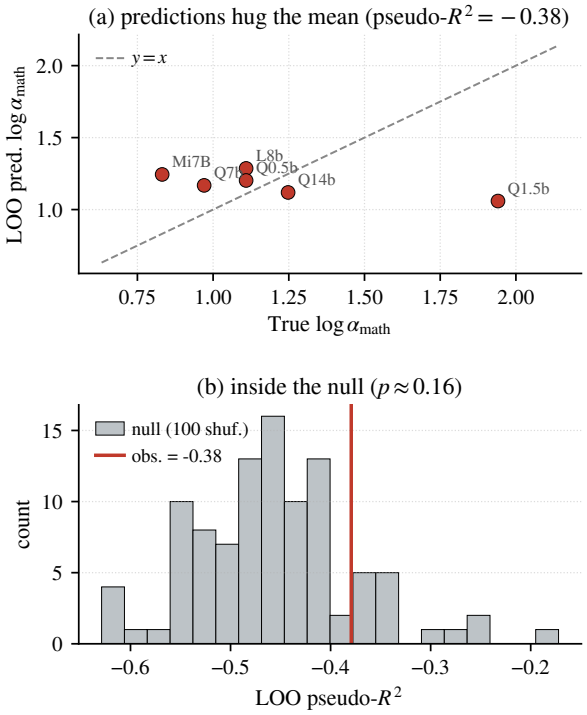


Figure 14: V4 base-only prediction of $\log \alpha_{\text{math}}$ is inside the $y$-shuffled null distribution (empirical $p \approx 0.21$). At $n = 6$ models, base statistics alone do not determine $\delta_m$; in our experiments we therefore use a one-cell calibration baseline (V3) that estimates $\delta_m$ from each model's Math500 Base/RL pair, while a base-only or base+rltrain predictor of $\delta_m$ remains an open direction.

### A.4.2 Reviewer-risk checklist

This checklist captures the failure modes we found ourselves making during draft revisions, and the guardrails that resolved them. Each item maps to a specific cell in the analysis where the unguarded claim would have been wrong.

- Do not claim parameter-level equivalence between RL and an external decoding policy (Section 6.4: the rule is a behavioral diagnostic, not a parameter recovery).
- Do not claim exact post-RL performance prediction unless a strong checkpoint-selection baseline is included (the rule predicts the operating-point shift, not the absolute pass-rate after training).
- Separate horizontal same-model transfer from vertical cross-model transfer (Tables 2B and 3A use different baseline families for a reason).
- Mark AR-only and expanded-policy pools separately (the recovery near-match search uses both; the BOPTR rule selector uses only AR).
- Treat IFEval as evaluation-metric mismatch/OOD rather than as a direct math-style

| Rule | State | Parameters | Use in paper | Interpretation |
|---|---|---|---|---|
| Per-budget oracle | scalar metric | high | upper bound | Shows recoverability but not structure. |
| Smooth path | metric-specific | medium | diagnostic | Tests whether recovery is coherent across budgets. |
| v5 shared-core | [P,Q,T] | low-medium | main diagnostic | Most stable current rule. |
| v6 signed SAT | [S,A,T] | medium | ablation | Demonstrates compression need but hurts expansion. |
| v7 axis-decoupled | [S,A,T] | high | negative ablation | Shows single-anchor under-identification. |
| v8-H planned | [P,Q,T] + adapter | low-medium | future/main candidate | Same rule family with model-specific shifts. |

Table 14: How to position rule variants. The current paper should emphasize v5 as a stable diagnostic and v6/v7 as evidence about mechanism identifiability.

SC benchmark (regime = nonmath, not math).

- Present 1.5B and Llama as stress tests when generation length or protocol behavior differs (the lower-half cohort in Table 3A is the headline failure case for the V2 direct anchor-path copy).

#### A.4.3 Recommended figures

The list below is the figure inventory we recommend for any follow-up paper or talk that extends this analysis. Items 1–4 already appear in the main paper or appendix; items 5–6 are planned for the v8-H follow-up.

1. **Main pipeline**: candidate base models, RL post-training, base UDF landscapes, same-policy and envelope curves, and analysis outputs.
2. **UDF components**: local policy, controller, evaluator, transition, and scheduler.
3. **Same-policy curves**: Base vs RL under identical policies across budgets.
4. **Envelope and near-match heatmap**: policy-by-budget landscape with RL target and recovering paths.
5. **Rule evolution**: v3/v5/v6/v7 mechanism diagrams and where each succeeds or fails.
6. **Horizontal/vertical matrix**: model-by-benchmark cells annotated by expansion, sharpening/compression, termination, or pool-limited behavior.

### A.5 Artifact documentation, licenses, and intended use

**Artifacts used and citation.** We use publicly released checkpoints, benchmarks, and software packages. These include SimpleRL-Zoo checkpoints and scripts, the evaluated model families, Math500/MATH, AIME 2024/2025, GPQA-Diamond, IFEval, vLLM, and HuggingFace/Transformers. We cite the creators of these artifacts where they are introduced in the main text and appendix. Our released artifacts consist of code, configuration files, and analysis scripts for reproducing the experiments in this paper.

**Licenses, terms, and intended use.** We use third-party checkpoints, benchmarks, and software packages under their original licenses and terms of use. All existing artifacts are used only for research and evaluation of model behavior, consistent with their stated intended use and access conditions. Our released code, configuration files, and analysis scripts are intended for research reproduction and analysis. They do not redistribute third-party checkpoints or benchmark data. Users should obtain third-party artifacts from their original sources under their original licenses and terms.

**Artifact coverage.** The evaluation artifacts cover mathematical reasoning, scientific question answering, and instruction following. Math500/MATH and AIME are English mathematical reasoning benchmarks. GPQA-Diamond is an English graduate-level science question-answering benchmark. IFEval is an English instruction-following benchmark with automatically verifiable instructions. The artifacts are not designed for demographic or social-group analysis, and we do not use them to study demographic-group representation.

**Data statistics and splits.** Math500 contains 500 problems from the MATH benchmark. AIME 2024 and AIME 2025 contain 30 problems each. GPQA-Diamond contains 198 multiple-choice science questions. IFEval contains approximately 500 prompts with verifiable instructions. We use these public benchmark evaluation sets as provided and do not create new train/dev/test splits. Our main experiments evaluate budgets $b \in \{1, 2, 4, 8, 16\}$,

paired Base/RL checkpoints, and the evaluation metrics described in Section 4.

**Personally identifying information and offensive content.** We do not collect new human-subject data or user-generated personal text. The benchmarks used in this work are publicly released research/evaluation artifacts for mathematical reasoning, science question answering, and instruction-following evaluation. We use these benchmarks only through their official evaluation prompts and aggregate metrics, and we do not release raw benchmark contents, model generations tied to individuals, or any derived dataset containing personal identifiers. We reviewed the benchmark descriptions and task schemas used in our experiments and found no fields designed to collect or expose personal identifiers. We did not apply additional anonymization because our created artifacts are code, configuration files, and aggregate analysis scripts/results rather than newly collected text data. Any risk of sensitive or offensive content is inherited from the original public benchmarks, which we cite and use under their stated research/evaluation terms.