SHARED ACCURACY–LENGTH PATTERNS IN REASONING COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards enables strong reasoning, and reasoning-compression methods seek shorter correct responses through different training objectives and length controls. We place their independently trained policies in a common observable coordinate defined by dataset accuracy and mean response length. Across the observed task, model, and controller extensions, low-parameter functions predict held-out policy endpoints in this accuracy–length space. On the primary math sweeps, one shared endpoint direction learned from one group of prompts captures most transferable variation on a disjoint group. Its endpoint values are approximately monotone in realized response length with nonlinear spacing. Simple exponential and hyperbolic one-parameter shapes capture this spacing with little predictive loss and remain predictive after correctness is averaged over prompts. Together, these results connect the dataset-level accuracy–length structure of completed compression sweeps to compact transferable structure already shared across individual prompts.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) uses automatically checkable task correctness as a training signal and has become a standard test bed for reasoning post-training (Lambert et al., 2025; Guo et al., 2025). However, RLVR can also produce unnecessarily long responses. *Reasoning compression* addresses this issue by changing the training objective or imposing a response-length budget to favor shorter correct responses (Arora & Zanette, 2025; Aggarwal & Welleck, 2025; Li et al., 2026a). We use *compression controller* for the method-level rule that introduces this length preference, such as a fixed penalty or an adaptive length target; a *controller setting* is one fixed configuration of that rule, and independently training policies over a grid of settings gives a *compression sweep*.

Different controllers expose different quantities as settings, so their raw settings are not numerically comparable. We instead represent each trained policy by dataset accuracy A and mean response length L, call the resulting point a *policy endpoint*, and call the endpoints from one sweep its accuracy–length (A–L) curve. **Research question 1 (RQ1) asks whether functions with few parameters can predict held-out policy endpoints in this observable A–L coordinate.** A positive result would give a controller-independent description of completed sweeps without equating controller-specific hyperparameters.

A simple dataset-level A–L curve can arise because many prompts change together across the same trained policies, or because unrelated prompt changes cancel after averaging correctness. **Research question 2 (RQ2) asks how many shared endpoint directions learned from one set of prompts are predictively useful on a disjoint set of prompts.** This separates a transferable prompt-level pattern from an aggregate curve that could arise only through averaging.

Even if one endpoint direction transfers across prompts, its relation to response length remains unresolved. **Research question 3 (RQ3) asks how the first shared endpoint direction depends on realized response length.** We answer this RQ with three tests: length-based baselines, a one-parameter shape restriction, and a final check that the prompt-learned shape remains predictive after averaging prompts. Math is the primary setting because its dense sweeps provide broad accuracy movement; multiple-choice and code tasks provide extensions with different answer formats and accuracy spans. Figure 1 summarizes the three RQs.

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.

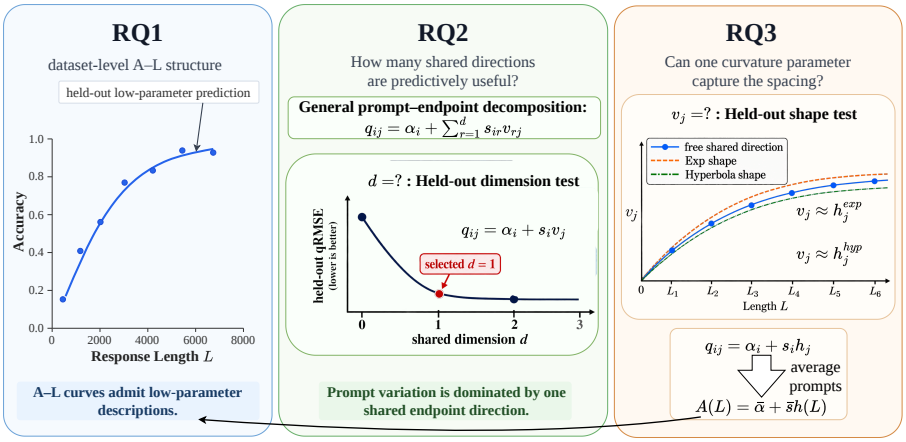


Figure 1: Paper overview. RQ1 tests held-out A–L endpoint prediction. RQ2 tests how many shared endpoint directions are useful without assuming a length relation. RQ3 first compares the learned direction with direct length-based alternatives, then tests a one-parameter length shape and whether that shape remains predictive after prompt averaging. For compactness, $v_j \equiv v_{1j}$; the displayed algebra is schematic, with the tested predictor and approximations defined in Sections 5–6.

Our main contributions are:

- Across the observed model, controller, and task extensions, heterogeneous compression sweeps admit a controller-independent A–L representation in which low-parameter functions predict held-out endpoints; fitted curves also support retrospective accuracy-change summaries and reconstruction from retained endpoints.
- On the primary math sweeps, one shared endpoint direction transfers from source prompts to disjoint target prompts, while a second direction provides little consistent additional predictive benefit.
- The first shared direction is approximately monotone but nonlinearly spaced in response length. One-parameter exponential and hyperbolic shapes preserve its held-out predictive accuracy and remain predictive after prompt averaging.

Together, these findings suggest that completed reasoning-compression sweeps have compact structure in observable A–L space and that, on the primary math setting, the dataset-level curve reflects transferable prompt-level variation rather than averaging alone.

2 Related Work

Reinforcement learning with verifiable rewards as a reasoning-compression test bed. RLVR supplies a verifiable correctness signal across reasoning tasks (Lambert et al., 2025; Guo et al., 2025), making it a controlled setting for varying the response-length preference while retaining the task-success signal. Recent RLVR work also studies how post-training changes reasoning length, efficiency, and the allocation of reasoning effort (Tan et al., 2026; Liu et al., 2025; 2026). Our focus is complementary: we analyze the geometry of the completed sweeps produced by such training procedures rather than introduce a new RLVR optimizer.

Reasoning compression. Existing reasoning-compression methods alter training through length penalties or reward shaping (Arora & Zanette, 2025; Liu et al., 2025; Peng et al., 2026), explicit length targets or limits (Aggarwal & Welleck, 2025; Hou et al., 2026), adaptive coefficients (Li et al., 2026a; Liu et al., 2026), or prompt-dependent signals (Xiang et al., 2025; Yuan et al., 2026; Yu et al., 2026). Other methods decide when to reason longer, predict budgets, or penalize redundant content (Fang et al., 2025; Tu et al., 2025; Li et al., 2026b; Cheng et al., 2026). These works optimize or select individual policies. We instead study simple A–L and prompt-level structure across completed sweeps.

Scaling laws and efficiency frontiers. Scaling-law work predicts performance as training resources, model size, or inference configurations vary, and related efficiency work studies global cost-performance frontiers (Wu et al., 2025; Bian et al., 2025). Our analysis conditions on an observed compression sweep rather than fitting a global resource-to-performance law. It neither assumes that the observed policy endpoints are Pareto-optimal nor equates controller-specific training settings. It asks how accuracy changes with realized response length across the trained policies in one completed sweep.

3 PRELIMINARIES

3.1 RLVR AND COMPRESSION CONTROLLERS

Let x denote a prompt, $y \sim \pi_\theta(\cdot \mid x)$ one sampled response, $c(x, y) \in \{0, 1\}$ correctness, $C(x, y)$ the verifiable training reward, and $\ell(y)$ response length. We call one sampled response a *rollout*. We use Linear Penalty (LP), $R_{\mathrm{LP}}(x, y) = C(x, y) - \beta\ell(y)$ with fixed β, as a simple static length-cost baseline related to prior length-penalty objectives (Liu et al., 2025; Peng et al., 2026); LEASH uses target length L_{tgt} and adaptive coefficient λ_t (Li et al., 2026a). COSINE is our fixed-coefficient ablation that replaces LP's linear penalty with a nonlinear cosine penalty; we retain this name to distinguish the specific ablation from cosine penalties generally. Appendix B gives the exact controller equations, including the prompt-dependent LP ablations.

3.2 ACCURACY–LENGTH COORDINATE

Because the controller settings are different quantities, they cannot define a common horizontal axis. Therefore, for policy π_j evaluated on N prompts $\{x_i\}_{i=1}^N$, we define

$$(L_j, A_j) = \left(\frac{1}{N} \sum_{i=1}^{N} \mathbb{E}[\ell(y) \mid x_i, \pi_j],\ \frac{1}{N} \sum_{i=1}^{N} \Pr[c(x_i, y) = 1 \mid \pi_j] \right). \tag{1}$$

As defined in the Introduction, (L_j, A_j) is a policy endpoint. Training J controller settings independently gives the compression sweep $\mathcal{C} = \{(L_j, A_j)\}_{j=1}^J$, which forms its A–L curve. At the prompt level, $q_{ij} = \Pr(\text{correct} \mid x_i, \pi_j)$ denotes the correctness probability for prompt i at policy endpoint j. We write $q_i(L_j)$ only when emphasizing the realized length coordinate. Therefore, this notation does not treat response length as a causal intervention.

4 RQ1: CAN FUNCTIONS WITH FEW PARAMETERS PREDICT A–L CURVES?

RQ1 asks whether a few parameters are enough to predict the accuracy of a held-out trained policy from its mean response length. We compare two-, three-, and four-parameter candidate families, then extend the reference sweep across models, tasks, and controllers while holding the other experimental dimensions fixed where applicable.

4.1 EXPERIMENTAL SETTINGS

We use DeepSeek-R1-Distill-Qwen-1.5B, abbreviated DSDT-1.5B, with LP on six math benchmarks as the reference setting because it has the largest number of policy endpoints and a large accuracy span. The six benchmarks are AIME24, AIME26, AMC23, MATH500, Minerva Math, and OlympiadBench (Mathematical Association of America, 2024; 2026; 2023; Lightman et al., 2024; Hendrycks et al., 2021b; Lewkowycz et al., 2022; He et al., 2024). For the model extension, we keep LP and the six math benchmarks fixed and evaluate DSDT-7B and Phi-4-mini-reasoning, abbreviated Phi-4-mini-R (Xu et al., 2025). For the task extension, we keep LP fixed and evaluate the multiple-choice tasks GPQA Diamond and MMLU-STEM together with LiveCodeBench v6 for code (Rein et al., 2024; Hendrycks et al., 2021a; Jain et al., 2025). For the controller extension, we keep DSDT-1.5B and the six math benchmarks fixed and compare LP, LEASH, and COSINE. Appendix A gives the training and model-specific evaluation configuration. Prompt-dependent LP ablations are reported in Appendix F.3.

Setting	3-par. RMSE (pp) [med., IQR]	$G_{2\to3}$ (pp)	$G_{3\to4}$ (pp)
Model extension			
DSDT-1.5B / LP	1.62 [1.36, 1.87]	1.82	-0.04
DSDT-7B / LP	2.72 [2.51, 3.16]	3.30	-1.76
Phi-4-mini-R / LP	2.56 [1.83, 3.61]	2.61	-0.12
Controller extension			
DSDT-1.5B / LP	1.62 [1.36, 1.87]	1.82	-0.04
DSDT-1.5B / LEASH	1.18 [0.57, 1.79]	2.33	-0.12
DSDT-1.5B / COSINE	1.00 [0.75, 1.30]	1.11	-0.96

Table 1: Pass@1 parameter-count comparison on the six math benchmarks. The three-parameter column selects the lowest held-out RMSE among Exp-3, Weibull-3, and Hyperbola-3; positive gain means that moving to the higher parameter class lowers held-out RMSE.

4.2 Candidate Functions, Metrics, and Held-Out Results

Candidate functions and fitting. We pre-specify function families at each parameter count. The main three-parameter families cover exponential, stretched-exponential, and algebraic saturation, where *saturation* means that predicted accuracy approaches a plateau as response length increases:

$$A_{\mathrm{Exp3}}(L) = a - be^{-L/\lambda}, \qquad A_{\mathrm{Weibull3}}(L) = a\left[1 - e^{-(L/\lambda)^s}\right],$$
$$A_{\mathrm{Hyperbola3}}(L) = a - \frac{b}{L+\delta}. \tag{2}$$

Here a is the upper accuracy level; the remaining parameters control amplitude, horizontal scale, or curvature. The primary two-parameter class contains Linear and Log baselines, while the four-parameter class contains monotone functions with one additional shape or transition parameter. Appendix Table 6 lists the complete set of candidate families. The comparison is restricted to these pre-specified families and does not claim a universal optimum over analytic functions.

Metrics and fitting. Each compression sweep contains a pre-specified number J of independently trained policy endpoints, with $7 \le J \le 16$ across the main grids; Appendix Table 7 reports the controller-specific counts. For candidate family f, leave-one-out cross-validation fits $\widehat{A}_{-j,f}$ on all policy endpoints except j and scores the root mean squared error (RMSE)

$$\mathrm{RMSE}_{\mathrm{CV}}(f) = 100\sqrt{\frac{1}{J}\sum_{j=1}^{J}\left(A_j - \widehat{A}_{-j,f}(L_j)\right)^2}\ \text{pp}, \tag{3}$$

where pp denotes percentage points. For parameter count p, let $E_p(D)$ be the lowest cross-validation RMSE among the families used in the primary p-parameter comparison on dataset D. We report

$$G_{2\to3} = \mathrm{median}_D[E_2(D) - E_3(D)], \qquad G_{3\to4} = \mathrm{median}_D[E_3(D) - E_4(D)]. \tag{4}$$

Thus positive gain means that the higher-parameter class lowers held-out error. Table 1 reports medians and interquartile ranges (IQRs). Pass@1 is primary; pass@k is the probability that at least one of k independent sampled responses is correct, averaged over prompts. Appendix C.2 reports pass@5 and pass@10.

Main fitting result. Table 1 shows $G_{2\to3} > 0$ in every model and controller summary row, with gains of 1.11–3.30 pp, while $G_{3\to4} < 0$ in every row. Thus, within the candidate families considered and the observed endpoint grids, moving from two to three parameters improves held-out prediction, whereas the four-parameter alternatives considered do not improve the aggregate summaries. Across the 30 math controller–dataset combinations, Weibull-3 and Hyperbola-3 have the lowest median RMSEs and give the lowest error in 19 combinations. We therefore take few-parameter predictability, rather than a uniquely identified analytic family, as the RQ1 result; Appendix C.2 gives the full comparison.

Model	Task	Accuracy span (pp)	ρ	Held-out RMSE (pp)	nRMSE (%)
DSDT-1.5B	Math (median)	37.60	0.98	1.62	4.3
	MMLU-STEM	25.00	0.86	1.71	6.8
	GPQA Diamond	9.90	0.47	1.47	14.8
	LiveCodeBench v6	2.61	0.68	0.68	26.2
DSDT-7B	Math (median)	51.50	0.99	2.72	5.3
	MMLU-STEM	27.20	0.85	3.88	14.3
	GPQA Diamond	18.80	0.65	2.12	11.3
	LiveCodeBench v6	6.86	0.87	0.97	14.1

Table 2: Results for math, multiple-choice, and code under LP. nRMSE is $100\times$ held-out RMSE divided by the observed accuracy span; the math row summarizes the six math datasets.

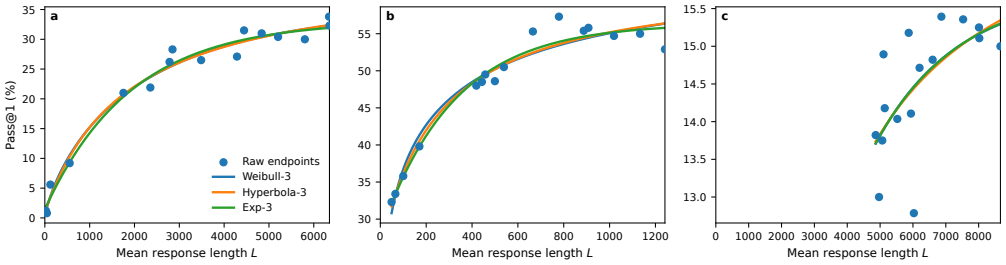


Figure 2: Representative DSDT-1.5B LP A–L curves on AIME24, MMLU-STEM, and LiveCodeBench v6. Points are policy endpoints and lines are three-parameter fits.

Task extensions. Absolute RMSE must be interpreted relative to the observed accuracy movement. Therefore, we also report

$$\text{span} = 100(\max_j A_j - \min_j A_j), \qquad \rho = \text{corr}(\text{rank}(L_j), \text{rank}(A_j)),$$
$$\text{nRMSE} = 100\,\text{RMSE}_{\text{CV}}/\text{span}. \tag{5}$$

where span is measured in percentage points, ρ is Spearman rank correlation, and nRMSE is normalized RMSE. Table 2 summarizes DSDT task extensions under LP; complete panels, including Phi-4-mini-R, are in Appendix C.3. Task extensions use only LP, whereas controller extensions use the six math benchmarks with LP, LEASH, and COSINE.

The two DSDT math sweeps have median spans of 37.6 and 51.5 pp, strong ordering, and nRMSE near 4–5%. Low-parameter functions also fit the task extensions, but the strength of the normalized evidence depends on the available accuracy span. MMLU-STEM at 1.5B has a 25.0-pp span with 6.8% nRMSE. LiveCodeBench at 1.5B has only a 2.61-pp span, so its 26.2% nRMSE is weaker; at 7B its span grows to 6.86 pp and nRMSE improves to 14.1%. Thus, the multiple-choice evidence is clearer, while the code evidence improves as the observed accuracy span and length–accuracy ordering increase. Figure 2 shows representative fits across math, multiple-choice, and code; complete per-dataset panels are in Appendix C.3.

4.3 Uses of the Fitted Accuracy–Length Curve

The fitted A–L curve also summarizes a completed sweep independently of the prompt-level analyses below.

Accuracy differences between observed policies. The fitted curve provides retrospective, non-causal summaries of finite accuracy differences between observed policy endpoints and fitted local accuracy return along the observed sweep; Appendix H reports math, multiple-choice, and code results.

Reconstructing the curve from retained endpoints. We also test how many observed policy endpoints are needed to summarize a dense completed sweep, without inferring untrained controller

settings. On the 18 math LP sweeps, where β is the explicit controller setting, we always retain $\beta = 0$ and the largest-β endpoint; the anchor budget B counts retained observed endpoints, with interior anchors matched to equal-length targets. The *stable budget* is the smallest tested B after which all larger feasible budgets remain below the chosen error threshold; Appendix H gives the full protocol. Under nRMSE $\leq 10\%$, 14/18 curves have a stable budget (median $B = 3$); under nRMSE $\leq 5\%$, 9/18 do (median $B = 9$). RQ1 establishes held-out prediction and sweep summaries but not why the A–L curve is simple. RQ2 tests whether prompts share endpoint directions across policies.

5 RQ2: How Many Shared Endpoint Directions Are Predictively Useful?

Dataset accuracy is $A_j = N^{-1} \sum_i q_{ij}$, where $q_{ij} = \Pr(\text{correct} \mid x_i, \pi_j)$. A smooth average curve can therefore arise because many prompts change similarly across policies, or because unrelated prompt changes cancel in the average. Before assuming any dependence on response length, we define the d-direction predictor

$$\widehat{q}_{ij}^{(d)} = \alpha_i + \sum_{r=1}^{d} s_{ir} v_{rj}, \tag{6}$$

where α_i is the prompt-specific mean level, $v_r = (v_{r1}, \ldots, v_{rJ})$ is a shared endpoint direction that can be reused across prompts, and s_{ir} is the coefficient for prompt i. This definition does not assume that only a few directions exist. After subtracting each prompt's mean across policy endpoints, the finite prompt–endpoint matrix is reproduced exactly when all right singular vectors are retained. RQ2 tests whether the truncated predictor with one direction generalizes to held-out prompt responses and whether a second direction adds reproducible benefit. Appendix D gives the exact matrix derivation.

5.1 Prompt Split and Error Metric

Main-text RQ2 results use DSDT-1.5B on all six math benchmarks from RQ1—AIME24, AIME26, AMC23, MATH500, Minerva Math, and OlympiadBench—with LP, LEASH, and COSINE, giving 18 controller–dataset combinations. Holding DSDT-1.5B and the six datasets fixed isolates whether the prompt-level structure transfers across the three controller sweeps. For each prompt–endpoint combination, the 16 rollouts are split 8/8 into independent halves, and prompts are partitioned into disjoint source and target sets. Shared endpoint directions are learned only from source prompts. For a target prompt, its coefficients are fitted from all but one policy endpoint using one rollout half, and the held-out endpoint is scored against the independent half; prompt folds and rollout halves are swapped and averaged. The held-out endpoint remains visible on source prompts, because RQ2 tests transfer across prompts rather than extrapolation to a policy absent from all training data. For held-out set $\mathcal{H}$, we report prompt-correctness RMSE, denoted qRMSE,

$$\text{qRMSE} = 100 \sqrt{|\mathcal{H}|^{-1} \sum_{(i,j) \in \mathcal{H}} (\widehat{p}_{ij} - \widehat{q}_{ij}^{\text{test}})^2} \text{ pp}, \tag{7}$$

where $\widehat{p}_{ij}$ is the predicted correctness probability and $\widehat{q}_{ij}^{\text{test}}$ is empirical correctness from the independent scoring half. Appendix D gives the full procedure and discusses finite-rollout scoring noise.

5.2 Held-Out Shared-Dimension Test

We compare $d \in \{0, 1, 2\}$ in Eq. (6). When $d = 0$, the sum is empty and only the prompt-specific level remains. When $d = 1$, the predictor adds the dominant endpoint direction learned from source prompts. When $d = 2$, it adds the strongest remaining orthogonal direction. We define the qRMSE gain from d to $d + 1$ as $\text{qRMSE}^{(d)} - \text{qRMSE}^{(d+1)}$, so positive gain means improved held-out prediction.

Figure 3 shows that $d = 0 \to 1$ improves held-out qRMSE in 17 of 18 controller–dataset combinations across the six math benchmarks, whereas $d = 1 \to 2$ improves in 4 of 18. Thus, one

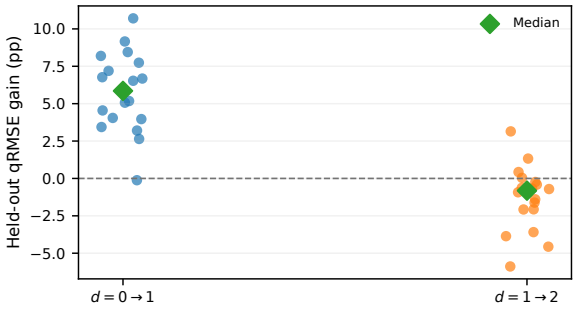


Figure 3: RQ2 held-out error after adding shared endpoint directions on the 18 controller–dataset combinations across the six math benchmarks. Each point is one combination, and diamonds mark medians. Positive qRMSE gain means that adding a direction lowers held-out error.

shared endpoint direction learned from source prompts captures most of the improvement on held-out prompts, while a second direction does not provide a consistent additional benefit. Appendix Table 10 reports the per-combination values.

Randomized smoothness check. We also compare against a smoothness-matched randomized null that preserves each prompt's variation across polynomial orders while independently randomizing its alignment with other prompts. The first shared direction captures more centered prompt-response variation than the randomized-data 97.5th percentile in 17 of the 18 controller–dataset combinations across the six math benchmarks. Appendix E gives the construction and secondary negative log-likelihood (NLL) and split-to-split cosine diagnostics.

RQ2 establishes that one transferable endpoint direction captures most of the held-out improvement, but it deliberately leaves that direction unconstrained by response length. This creates the next question: whether the learned values of v_1 across policy endpoints have a compact length-based description.

6 RQ3: How Does the First Shared Endpoint Direction Depend on Response Length?

RQ3 asks how the transferable direction v_1 depends on realized response length. We test successively whether monotonicity suffices, whether one curvature parameter reproduces the unrestricted endpoint profile, and whether the prompt-learned shape remains predictive after averaging prompts.

6.1 Length-Based Baselines

We compare the first shared endpoint direction v_1 with three alternatives under the same held-out target-prompt procedure. *Linear length* uses centered normalized response length, *length rank* keeps only endpoint ordering, and *monotone fit from source prompts* constrains the shared endpoint profile to be monotone in response length. Each alternative uses the same target-prompt intercept and amplitude fitting as v_1. We define the *qRMSE difference* as alternative qRMSE minus the qRMSE obtained with v_1, so a positive value favors the learned first direction, and report absolute cosine similarity between mean-centered endpoint vectors.

Linear length and length rank have median qRMSE differences of +1.35 and +1.44 pp and median cosine similarities of 0.854 and 0.863. The source-fitted monotone profile reaches 0.999 median cosine similarity with a -0.015 pp median qRMSE difference. Together these results identify an approximately monotone but nonlinearly spaced relation between the first shared direction and realized response length. Table 3 reports the corresponding counts and 95% interval test.

Length-based comparison	qRMSE difference	First direction lower error	95% interval above zero	Cosine similarity
Linear length	+1.35 pp	17/18	9/18	0.854
Length rank	+1.44 pp	16/18	12/18	0.863
Monotone fit (source prompts)	−0.015 pp	3/18	0/18	0.999

Table 3: Length-based comparisons for RQ3 on the six math datasets and three main controllers. qRMSE difference is comparison qRMSE minus the qRMSE of the first shared endpoint direction learned from source prompts, so positive values favor that learned direction. The interval column counts 95% confidence intervals obtained by resampling prompts that are entirely above zero.

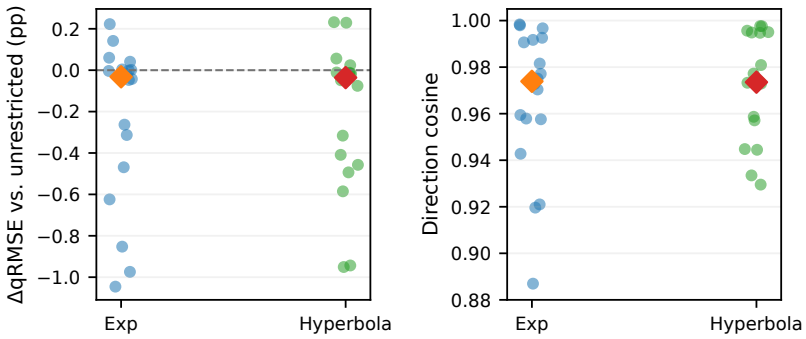


Figure 4: RQ3 one-parameter length-curve test on the 18 controller–dataset combinations across the six math benchmarks. Points are individual combinations and diamonds are medians. The left panel reports candidate qRMSE minus unrestricted-profile qRMSE, so positive values mean higher held-out error. The right panel reports cosine similarity with the unrestricted profile learned independently from target prompts.

6.2 One-Parameter Length Shape

The near-monotone alignment in Section 6.1 suggests that the remaining structure is the nonlinear spacing of direction values across response length. We next ask whether one shape parameter is enough to describe that spacing. For each source-prompt partition, we relearn an unrestricted endpoint profile $g = (g_1, \ldots, g_J)$ through

$$\widehat{q}_{ij}^{\,\text{free}} = \alpha_i + s_i g_j. \tag{8}$$

We relearn g on each source split rather than reuse the RQ2 direction v_1, so this comparison isolates the predictive cost of restricting an otherwise unconstrained one-direction profile to a length-dependent shape. We compare it with exponential and hyperbolic curves evaluated at normalized length $u = (L - L_{\min})/(L_{\max} - L_{\min})$:

$$h_\eta^{\exp}(u) = \frac{1 - e^{-\eta u}}{1 - e^{-\eta}}, \qquad h_\kappa^{\text{hyp}}(u) = \frac{(1+\kappa)u}{u + \kappa}. \tag{9}$$

For the saturating forms used here, $\eta > 0$ and $\kappa > 0$. Both satisfy $h(0) = 0$ and $h(1) = 1$, so η or κ controls the curve shape while α_i and s_i control each prompt's level and amplitude. We report Δ qRMSE as candidate-curve qRMSE minus unrestricted-profile qRMSE, so positive values mean that the shape restriction increases held-out error. We report this together with the cosine similarity defined in Section 6.1. The exponential and hyperbolic curves have cosine similarities of 0.974 and 0.974 and qRMSE differences of -0.03 and -0.04 pp. Figure 4 summarizes these comparisons. Thus, one shape parameter reproduces the unrestricted one-direction profile with little change in held-out error; the small negative median gaps do not identify either family as uniquely correct.

Geometric link to RQ2. Appendix D shows that for a centered unit length-curve direction u with $|v_1^\top u| = \cos\phi$, the rank-one reconstruction penalty satisfies $\mathcal{E}(u) - \mathcal{E}(v_1) \le \sigma_1^2 \sin^2\phi$. Thus Figure 4's cosine measures structural alignment with the first shared direction, while held-out qRMSE separately tests predictive transfer.

6.3 Does a Curve Learned from Individual Prompts Predict Dataset Accuracy?

RQ2 shows that one direction learned across source prompts is usually enough for held-out prediction, and RQ3 shows that a simple function of response length approximates that direction. Together these results support the shared-shape approximation

$$q_i(L) \approx \alpha_i + s_i h(L;\theta), \tag{10}$$

where $h(L;\theta)$ is the shared length-dependent curve and θ is its shape parameter. Averaging correctness over prompts then preserves the same functional form under this approximation:

$$A(L) = \frac{1}{N}\sum_i q_i(L) \approx \bar{\alpha} + \bar{s}\, h(L;\theta), \tag{11}$$

where $\bar{\alpha}$ and $\bar{s}$ are the mean prompt-specific level and amplitude. For this final transfer test, the held-out policy endpoint is excluded when learning θ from source prompts. We then keep θ fixed and fit only $\bar{\alpha}$ and $\bar{s}$ to the remaining dataset-average correctness values before scoring the held-out endpoint. For comparison, we fit the same family directly to those same remaining dataset endpoints and score the same held-out endpoint. The direct-fit versus prompt-learned-curve RMSE is 2.91 versus 3.47 pp for the exponential curve and 3.51 versus 3.45 pp for the hyperbolic curve. The prompt-learned shapes therefore retain dataset-accuracy error comparable to fitting the same families directly to the averaged policy endpoints, linking RQ2 and RQ3 back to the A–L curve in RQ1. Full length-curve and dataset-accuracy results are reported in Appendix F, and prompt-level results on multiple-choice and code are reported in Appendix G.

7 Limitations

Our conclusions characterize completed sweeps of independently trained policies, not causal token deletion. RQ1 compares the best family within pre-specified but unequal candidate sets and uses only 7–16 policy endpoints per sweep, so its parameter-count result concerns held-out prediction under these grids rather than a universal complexity optimum. RQ1 also scores predictions against finite-sample endpoint estimates from 16 responses per prompt, so part of the held-out error reflects evaluation noise, especially on smaller benchmarks. The model extension uses model-specific prompts, native tokenizations, and sampling temperatures, so it tests structural robustness rather than exact cross-model length equivalence; replicated-policy training variance is also unmeasured. RQ2 supports one dominant transferable endpoint direction, not an exactly rank-one response matrix. RQ3 shows that this direction is approximately monotone in realized length and closely approximated by the tested one-parameter shapes, but does not establish a causal effect of length or a unique analytic law. The prompt-level interpretation is strongest on the six math datasets; multiple-choice provides additional support in Appendices F and G, while code gives weaker and mixed evidence. Small-span regimes provide weaker normalized evidence, the LP-C/LP-D checks do not identify which prompt-level quantities change under prompt-dependent interventions, and extrapolation beyond the observed length range remains untested.

8 Conclusion

We represent independently trained reasoning-compression policies by dataset accuracy and realized mean response length. Across the observed sweeps, low-parameter functions predict held-out policy endpoints. On the primary math sweeps, one shared endpoint direction captures most transferable variation across prompts; it is approximately monotone but nonlinearly spaced in response length, and exponential or hyperbolic shapes preserve held-out predictive accuracy after prompt averaging. This gives a controller-independent description of completed sweeps, including finite accuracy-change and fitted local-return summaries and reconstruction from retained endpoints, while the prompt-level analysis identifies when aggregate simplicity reflects shared variation rather than averaging alone.

AI USE STATEMENT

Generative AI tools were used to assist language editing, paper organization and debugging analysis code. All AI-assisted text and code changes were reviewed and verified by the authors, who take responsibility for the final content.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=4jdIxXBNve.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 60770–60808. Curran Associates, Inc., 2025. doi: 10.52202/085713-2032. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/579b5b84311e122584dedab1d3b7613f-Paper-Conference.pdf.

Song Bian, Minghao Yan, and Shivaram Venkataraman. Scaling inference-efficient language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4303–4323. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bian25b.html.

Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. Optimizing length compression in large reasoning models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3232–3250, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.146. URL https://aclanthology.org/2026.acl-long.146/.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 151268–151295. Curran Associates, Inc., 2025. doi: 10.52202/085713-5060. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/de2ad3ed44ee4e675b3be42aa0b615d0-Paper-Conference.pdf.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4 edition, 2013. ISBN 9781421407944.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021b. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of LLMs via reinforcement learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=V51gPu1uQD.

Naman Jain, 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. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/94074dd5a072d28ff75a76dabed43767-Paper-Conference.pdf.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857. Curran Associates, Inc., 2022. doi: 10.52202/068431-0278. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/18abbeef8cfe9203fdf9053c9c4fe191-Paper-Conference.pdf.

Yanhao Li, Lu Ma, Jiaran Zhang, Lexiang Tang, Wentao Zhang, and Guibo Luo. LEASH: Adaptive length penalty and reward shaping for efficient large reasoning model. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2846–2856, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.129. URL https://aclanthology.org/2026.acl-long.129/.

Zheng Li, Qingxiu Dong, Jingyuan Ma, Di Zhang, Kai Jia, and Zhifang Sui. SelfBudgeter: Adaptive token allocation for efficient LLM reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21135–21156, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1063. URL https://aclanthology.org/2026.findings-acl.1063/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. Dler:

Doing length penalty right - incentivizing more intelligence per token via reinforcement learning, 2025. URL https://arxiv.org/abs/2510.15110.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng LIU, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 43279–43308, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/47795c4ae2f7d07ea2fb0d11fa2c3c90-Paper-Conference.pdf.

Mathematical Association of America. 2023 american mathematics competition (amc). MAA American Mathematics Competitions, 2023. URL https://maa.org/student-programs/amc/.

Mathematical Association of America. 2024 american invitational mathematics examination (aime). MAA American Mathematics Competitions, 2024. URL https://maa.org/student-programs/amc/.

Mathematical Association of America. 2026 american invitational mathematics examination (aime). MAA Invitational Competitions, 2026. URL https://maa.org/maa-invitational-competitions/.

Paavo Parmas, Yongmin Kim, Kohsei Matsutani, Shota Takashiro, Soichiro Nishimori, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. Ordergrad: Optimizing beyond the mean with order-statistic policy gradient estimation, 2026. URL https://arxiv.org/abs/2606.06096.

Keqin Peng, Yuanxin Ouyang, Xuebo Liu, Zhiliang Tian, Ruijian Han, Yancheng Yuan, and Liang Ding. Think dense, not long: Dynamic decoupled conditional advantage for efficient reasoning, 2026. URL https://arxiv.org/abs/2602.02099.

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. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Sijun Tan, Michael Luo, Justin Wong, Colin Cai, Xiaoxiang Shi, William Yuan Tang, Manan Roongta, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Effective RL scaling of reasoning models via iterative context lengthening, 2026. URL https://openreview.net/forum?id=I6GzDCne7U.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 15181–15207. Curran Associates, Inc., 2025. doi: 10.52202/085713-0510. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/16371a9d5fed65d6d78ca3a7fa6e598c-Paper-Conference.pdf.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for llm problem-solving. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 55824–55845, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/8c3caae2f725c8e2a55ecd600563d172-Paper-Conference.pdf.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang T. Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025. URL https://openreview.net/forum?id=oHR5d6tuXs.

Haoran Xu, Baolin Peng, Hany Awadalla, Dongdong Chen, Yen-Chun Chen, Mei Gao, Young Jin Kim, Yunsheng Li, Liliang Ren, Yelong Shen, Shuohang Wang, Weijian Xu, Jianfeng Gao, and

Weizhu Chen. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math. *arXiv preprint arXiv:2504.21233*, 2025. URL https://arxiv.org/abs/2504.21233.

Zewei Yu, Lirong Gao, Yuke Zhu, Bo Zheng, Junbo Zhao, Sheng Guo, and Haobo Wang. Stop unnecessary reflection: Training lrms for efficient reasoning with adaptive reflection and length coordinated penalty. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 4836–4854, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/08fe50bf209c57eecf0804f9f9ed639f-Paper-Conference.pdf.

Danlong Yuan, Tian Xie, Shaohan Huang, Huishuai Zhang, Zhuocheng Gong, Chong Luo, Furu Wei, and Dongyan Zhao. Shorten after you're right: Lazy length penalties for reasoning RL. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12864–12877, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.626. URL https://aclanthology.org/2026.findings-acl.626/.

A Training and Evaluation Configuration

A.1 Training optimization

Table 4 reports the shared training and rollout configuration used for the reference DSDT sweeps. PPO denotes proximal policy optimization. Configuration-specific option names are shown in monospace when needed for reproducibility. Training uses DeepScaleR (Tan et al., 2026), following the training-data choice in Liu et al. (2025).

Configuration	Value
Training data	DeepScaleR (Tan et al., 2026)
Advantage estimator	`reinforce_plus_plus_baseline`
Training epochs	2
Prompt batch size	128
Rollouts per prompt	16
PPO mini-batch size	128
Actor learning rate	1×10^{-6}
Learning-rate warmup	10 steps
Weight decay	0.1
PPO clip ratios	lower 0.20, upper 0.28; auxiliary clip constant 10
Kullback–Leibler loss	enabled, coefficient 5×10^{-4}, `low_var_kl`
Gradient clipping	1.0
Loss aggregation	`token-mean`
Maximum prompt / response length	1024 / 7168 tokens
Rollout sampling	$T = 0.8$, top-$p = 1.0$, top-k disabled

Table 4: Shared training and rollout configuration.

A.2 Evaluation protocol

Each evaluation uses the full evaluation split and 16 sampled responses per prompt. In the sampling configuration, T denotes temperature, top-p denotes nucleus-sampling probability mass, and top-k controls vocabulary truncation before sampling; top-$k = 0$ disables this truncation in our evaluator. The $T = 0.8$ value in Table 4 is the training-rollout setting; evaluation sampling is configured separately. All evaluation runs use seed 42. DSDT-1.5B and DSDT-7B use the `qwen25-math-cot` prompt with temperature 0.6; Phi-4-mini-R uses `phi4-mini-reasoning` with temperature 0.8. We sample with top-$p = 1$ and top-$k = 0$ and allow up to 38K generated tokens, with generation stopping earlier on </s>, <|im_end|>, or <|endoftext|>. All A–L analyses use the realized response lengths of the generated samples rather than this ceiling. Standard benchmarks are parsed

and scored directly; LiveCodeBench generations are executed and scored in a code sandbox. Table 5 summarizes the evaluation sampling settings.

Model	Prompt type	Split	Samples/prompt	Seed	Temperature
DSDT-1.5B	`qwen25-math-cot`	test, full	16	42	0.6
DSDT-7B	`qwen25-math-cot`	test, full	16	42	0.6
Phi-4-mini-R	`phi4-mini-reasoning`	test, full	16	42	0.8

Table 5: Evaluation sampling configuration used for the policy endpoints.

Prompt templates. For DSDT models, the system instruction is "Please reason step by step, and output your final answer within \boxed{}." The user template appends "Let's think step by step and output the final answer within \boxed{}." Phi-4-mini-R uses "Please reason step by step and put your final answer within \boxed{}." The target field is {output}. For each model, the same chat template and delimiters are used for every policy endpoint.

B Compression Controller Definitions and Related Context

This section gives the exact reward definitions summarized in Section 3. The symbols $c(x, y)$, $C(x, y)$, and $\ell(y)$ retain their main-text definitions.

Linear Penalty. LP subtracts a fixed linear length penalty,

$$R_{\mathrm{LP}}(x, y) = C(x, y) - \beta \ell(y), \qquad \beta \geq 0. \tag{12}$$

Here β is the controller setting and is fixed across prompts and training steps within one run. Therefore, $\beta = 0$ is the correctness-only RLVR anchor and larger β increases the preference for shorter responses (Parmas et al., 2026; Peng et al., 2026).

LEASH. LEASH uses target response length L_{tgt} and adaptive coefficient λ_t at training step t,

$$R_{\mathrm{LEASH}}(x, y) = \mathrm{clip}\left((2c(x, y) - 1) - \lambda_t[\ell(y)/L_{\mathrm{tgt}} - 1]_+, -1, 1\right). \tag{13}$$

Here $[z]_+ = \max(z, 0)$ is the positive-part operator, and $\mathrm{clip}(z, -1, 1)$ truncates z to $[-1, 1]$. The implementation updates the dual coefficient once per policy step using the signed mean violation,

$$\lambda_{t+1} = \mathrm{clip}\left(\lambda_t + \eta_\lambda \, \mathbb{E}_t[\ell(y)/L_{\mathrm{tgt}} - 1], \lambda_{\min}, \lambda_{\max}\right), \tag{14}$$

where η_λ is the dual step size. Thus the policy reward uses a one-sided excess-length penalty, while the dual update can both tighten and relax the effective length pressure (Li et al., 2026a).

COSINE penalty ablation. COSINE changes only the shape of the fixed LP length cost, using

$$P_{\cos}(\ell; \beta) = \beta \frac{B_{\max}}{2}\left[1 - \cos(\pi \ell / B_{\max})\right], \qquad R_{\mathrm{COSINE}}(x, y) = C(x, y) - P_{\cos}(\ell(y); \beta). \tag{15}$$

We set $B_{\max} = 7168$, equal to the training response cap in Table 4. This parameterization keeps β in the same reward-per-token units as LP and matches the LP penalty at $\ell = 0$, $B_{\max}/2$, and $B_{\max}$. We distribute the sequence-level cost uniformly over valid response tokens, so its summed contribution is exactly $-P_{\cos}$.

Prompt-dependent LP ablations. LP-C is the correctness-gated continuous LP:

$$R_{\mathrm{LP-C}}(x, y) = C(x, y) - \beta \, c(x, y) \, \ell(y). \tag{16}$$

Thus incorrect responses receive no additional length cost, whereas correct responses receive the same per-token price β as vanilla LP. LP-D instead uses the current on-policy solve rate of each prompt to weight its length cost. For prompt x_i with K training rollouts $y_{i1}, \ldots, y_{iK}$, define

$$\widehat{p}_i = \frac{1}{K} \sum_{k=1}^{K} c(x_i, y_{ik}), \qquad \widetilde{p}_i = \max(\widehat{p}_i, 1/K), \qquad w_i = (1 - \omega_{\mathrm{mix}}) + \omega_{\mathrm{mix}} \widetilde{p}_i, \tag{17}$$

where $\omega_{\mathrm{mix}} \in [0, 1]$ is the LP-D mixing coefficient. The LP-D experiments use $\omega_{\mathrm{mix}} = 0.5$. The reward for every rollout of that prompt is then

$$R_{\mathrm{LP-D}}(x_i, y_{ik}) = C(x_i, y_{ik}) - \beta w_i \ell(y_{ik}). \tag{18}$$

The weight is shared across the K rollouts of a prompt and is not normalized across prompts; $\omega_{\mathrm{mix}} = 0$ recovers LP, while $\omega_{\mathrm{mix}} = 1$ gives solve-rate weighting with the $1/K$ floor. Our training configuration uses $K = 16$ rollouts per prompt (Table 4). These two ablations change which prompts or responses receive compression pressure while retaining β as the base LP length-penalty scale.

Related reward designs. Prior work also uses correctness-gated and prompt-dependent length penalties, including penalties activated after a response is correct and penalties scaled by an online prompt solve-rate estimate (Yuan et al., 2026; Xiang et al., 2025). These designs motivate the LP-C and LP-D stress tests, but the A–L coordinate itself depends only on realized response length and evaluation accuracy, not on the form of the training reward.

C Additional RQ1 Results

C.1 Candidate families

The family comparison is fixed before evaluating held-out endpoints. Table 6 lists the formulas, scalar parameter counts, and the role of each candidate in the complexity comparison. Constant is a one-parameter reference only; the reported $G_{2\to3}$ and $G_{3\to4}$ comparisons use the primary two-, three-, and four-parameter sets stated in the table. Within each leave-one-endpoint-out fold, length is normalized by the median positive training length and each nonlinear family is fit by bounded least squares from deterministic data-adaptive starting points, retaining the lowest-residual solution. Scale and curvature parameters of the saturation families are constrained positive; the Linear and Log slopes may have either sign. The foldwise normalization changes only the horizontal parameterization, not the function class; for example, $a - be^{-x/\tau}$ with $x = L/m$ is the same family as Eq. (2) with $\lambda = m\tau$. In Table 6, $\sigma(z) = 1/(1 + e^{-z})$ is the logistic sigmoid and $\Phi(z)$ is the standard normal cumulative distribution function.

Function family	k	Form	Purpose
Constant	1	a	reference with no length effect
Linear	2	$a + bx$	non-saturating baseline
Log	2	$a + b \log x$	concave non-saturating baseline
Logistic-2 (free slope)	2	$\sigma(a + bx)$	nonlinear monotone comparison
Exp-3	3	$a - be^{-x/\tau}$	single-scale saturation
Weibull-3	3	$a[1 - e^{-(x/\tau)^s}]$	saturation with free curvature
Hyperbola-3	3	$a - b/(x + \delta)$	heavy-tail saturation
Logit-quadratic	3	$\sigma(a + bx + dx^2)$	non-monotone comparison
Stretch-4	4	$a - be^{-(x/\tau)^s}$	free-floor stretched saturation
Hill-4	4	$(a - b) + b\, x^s/(\tau^s + x^s)$	sigmoidal saturation
Logistic-4	4	$a + b\sigma((x - \mu)/w)$	generic smooth sigmoid
Probit-4	4	$a + b\Phi((x - \mu)/w)$	alternative sigmoid link
Logistic-4 (log L)	4	$a + b\sigma((\log x - \mu)/w)$	sigmoid on log length
Probit-4 (log L)	4	$a + b\Phi((\log x - \mu)/w)$	probit on log length

Table 6: Audited functions for dataset accuracy versus response length. $x = L/m$ with m the median positive endpoint length among the points used to fit that leave-one-endpoint-out fold. Constant is a one-parameter reference. The reported parameter-count gains use Linear/Log for two parameters, Exp-3/Weibull-3/Hyperbola-3 for three parameters, and the six monotone four-parameter saturation/transition families for four parameters. Logistic-2 and logit-quadratic are additional comparisons and do not define the main parameter-count result.

C.2 Complete held-out results

The main text summarizes the model and controller comparisons used for RQ1. Table 7 reports every math controller–dataset combination. The two-to-three parameter gain is positive in every listed combination, whereas the three-to-four gain is negative in most combinations; the notable positive exceptions, such as DSDT-1.5B/LP on MATH500 and DSDT-7B/LP on OlympiadBench, motivate reporting the complete per-dataset results.

In Table 8, MAE denotes mean absolute error, the 90th-percentile column is taken across controller–dataset combinations, and R^2 is the coefficient of determination. The table shows that absolute error increases from pass@1 to pass@5 and pass@10 (median RMSE 1.73, 2.35, and 3.08 pp), while median R^2 remains 0.97–0.98. The stable R^2 together with larger absolute RMSE suggests that the few-parameter accuracy–length relation remains predictive for higher-k metrics despite larger absolute errors. Table 9 shows where that advantage comes from: Weibull-3 and Hyperbola-3 have the lowest median RMSEs (1.82 and 1.94 pp) and together account for 19 of 30 family wins; the best four-parameter families do not improve the aggregate error distribution.

Model	Controller	Dataset	3-parameter held-out RMSE (pp)	Gain $2 \to 3$ (pp)	Gain $3 \to 4$ (pp)	J
DSDT-1.5B	LP	AIME24	1.75	1.38	-0.22	16
DSDT-1.5B	LP	AIME26	1.31	2.03	-0.15	16
DSDT-1.5B	LP	AMC23	0.76	2.20	-0.04	16
DSDT-1.5B	LP	MATH500	3.33	4.85	0.97	16
DSDT-1.5B	LP	Minerva Math	1.91	1.61	-0.05	16
DSDT-1.5B	LP	OlympiadBench	1.50	0.33	-0.03	16
DSDT-1.5B	LEASH	AIME24	1.70	2.46	-0.53	9
DSDT-1.5B	LEASH	AIME26	1.82	1.08	-0.20	9
DSDT-1.5B	LEASH	AMC23	2.94	2.11	0.91	9
DSDT-1.5B	LEASH	MATH500	0.54	3.79	0.03	9
DSDT-1.5B	LEASH	Minerva Math	0.38	2.59	-0.04	9
DSDT-1.5B	LEASH	OlympiadBench	0.65	2.19	-0.29	9
DSDT-1.5B	COSINE	AIME24	1.32	0.44	-0.27	7
DSDT-1.5B	COSINE	AIME26	1.56	0.20	-3.04	7
DSDT-1.5B	COSINE	AMC23	1.21	0.88	-1.79	7
DSDT-1.5B	COSINE	MATH500	0.74	2.30	-0.35	7
DSDT-1.5B	COSINE	Minerva Math	0.79	1.34	-1.57	7
DSDT-1.5B	COSINE	OlympiadBench	0.52	2.57	-0.04	7
DSDT-7B	LP	AIME24	2.63	3.93	-0.51	10
DSDT-7B	LP	AIME26	2.48	3.57	-0.97	10
DSDT-7B	LP	AMC23	1.63	0.96	-4.83	10
DSDT-7B	LP	MATH500	3.27	8.05	-3.88	10
DSDT-7B	LP	Minerva Math	5.44	3.03	-2.54	10
DSDT-7B	LP	OlympiadBench	2.81	0.60	2.35	10
Phi-4-mini-R	LP	AIME24	3.13	2.68	0.35	10
Phi-4-mini-R	LP	AIME26	1.78	2.54	0.12	10
Phi-4-mini-R	LP	AMC23	3.77	3.80	-0.75	10
Phi-4-mini-R	LP	MATH500	3.96	5.23	0.39	10
Phi-4-mini-R	LP	Minerva Math	1.58	1.50	-0.36	10
Phi-4-mini-R	LP	OlympiadBench	1.98	0.62	-0.42	10

Table 7: Per-dataset Pass@1 leave-one-endpoint-out comparison for all main math combinations. Positive gain means that increasing the parameter class lowers held-out RMSE.

Metric	N	RMSE med. [IQR]	Mean	90th pct.	Worst	MAE median	R^2 median
pass@1	30	1.73 [1.24, 2.77]	1.97	3.37	5.44	1.34	0.98
pass@5	30	2.35 [1.64, 3.98]	2.71	4.33	7.13	1.78	0.98
pass@10	30	3.08 [2.15, 4.42]	3.22	5.04	7.38	2.29	0.97

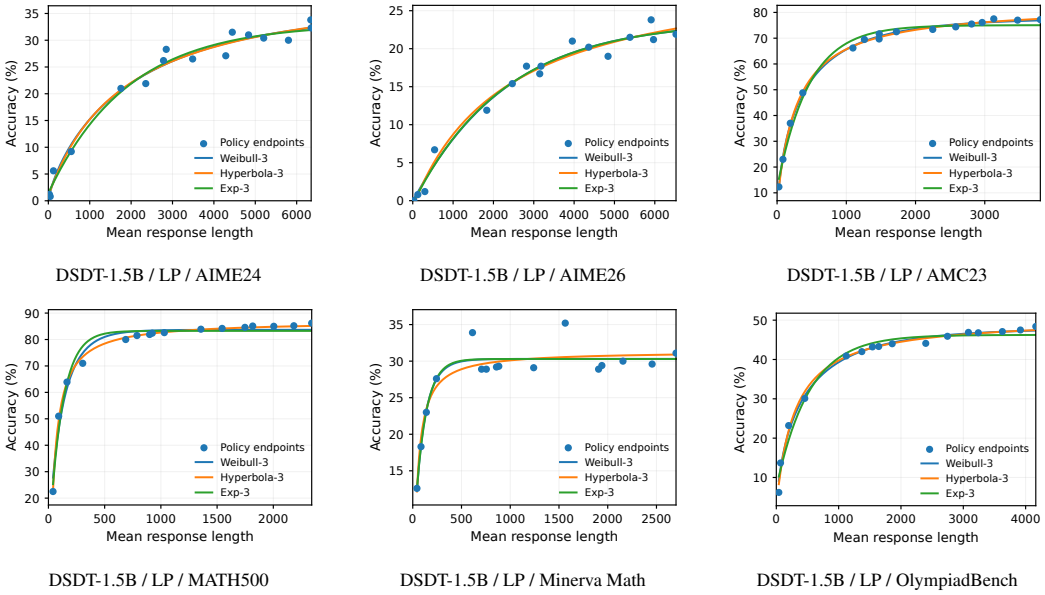
Table 8: Pass@1, pass@5, and pass@10 results for the best three-parameter family in each main math model/controller combination. RMSE and MAE columns are in percentage points.

k	Family	N	RMSE mean	RMSE med.	90th pct.	Worst	MAE median	R^2 median	Wins
1	Constant	30	14.84	13.27	22.25	32.85	10.97	-0.23	0
2	Linear	30	9.34	7.74	16.69	26.43	6.71	0.47	0
2	Log	30	4.46	3.27	8.21	11.33	2.53	0.88	0
2	Logistic-2 (free slope)	30	8.74	8.28	13.87	17.83	6.77	0.52	0
3	Exp-3	30	3.44	2.56	7.01	10.30	1.83	0.96	4
3	Hyperbola-3	30	2.53	1.94	4.67	5.99	1.50	0.97	10
3	Logit-quadratic	30	7.71	5.39	14.89	28.31	4.26	0.78	0
3	Weibull-3	30	2.22	1.82	4.22	5.52	1.50	0.97	9
4	Hill-4	30	3.17	2.38	7.38	10.55	1.73	0.96	2
4	Logistic-4	30	4.46	3.64	8.35	11.60	2.84	0.93	0
4	Logistic-4 (log L)	30	3.07	2.40	5.15	10.55	1.73	0.97	1
4	Probit-4	30	4.69	3.88	8.35	12.21	3.00	0.91	0
4	Probit-4 (log L)	30	3.07	2.30	5.15	9.92	1.79	0.96	2
4	Stretch-4	30	2.93	2.29	6.53	7.97	1.80	0.96	2

Table 9: Complete pass@1 function-family results on the main math model/controller combinations. RMSE and MAE columns are in percentage points. Wins count the curves on which a family has the lowest cross-validation RMSE among the candidate families considered.

C.3 COMPLETE A–L PANELS

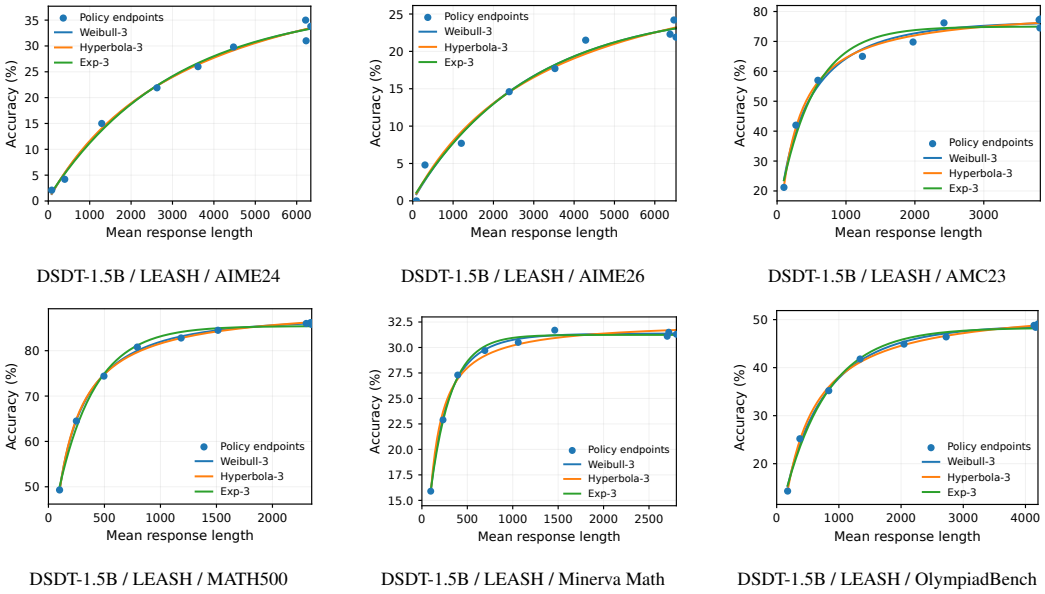
The six figures in this section report the complete three-parameter RQ1 fits across the evaluated model, controller, and task settings. Math results are grouped by controller or model, and the multiple-choice and code extensions are shown separately.



DSDT-1.5B / LP / AIME24 | DSDT-1.5B / LP / AIME26 | DSDT-1.5B / LP / AMC23

DSDT-1.5B / LP / MATH500 | DSDT-1.5B / LP / Minerva Math | DSDT-1.5B / LP / OlympiadBench

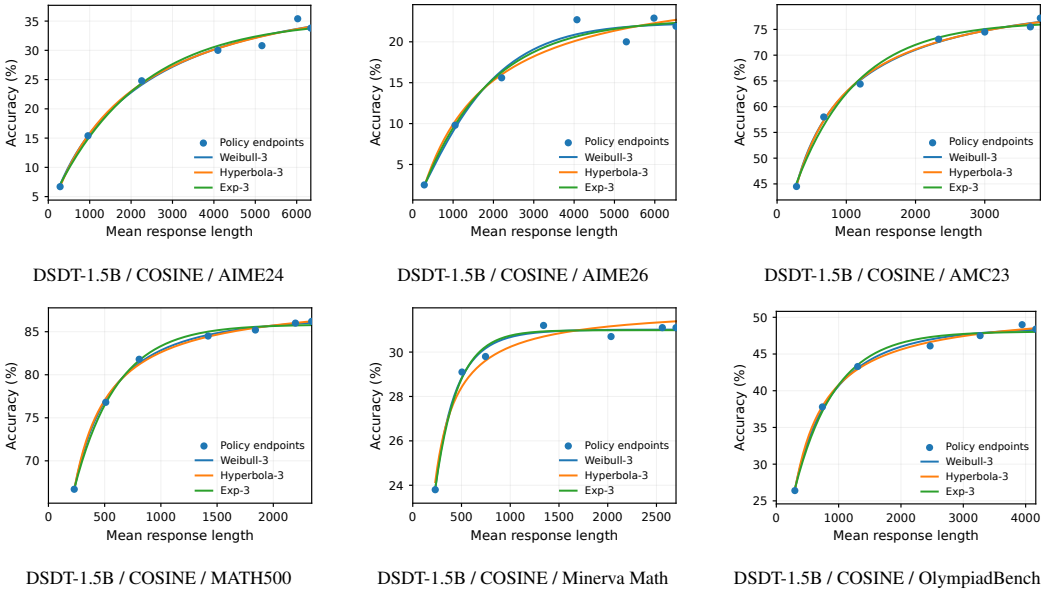
Figure 5: Math A–L endpoints and three-parameter fits for the DSDT-1.5B LP sweep.



DSDT-1.5B / LEASH / AIME24 | DSDT-1.5B / LEASH / AIME26 | DSDT-1.5B / LEASH / AMC23

DSDT-1.5B / LEASH / MATH500 | DSDT-1.5B / LEASH / Minerva Math | DSDT-1.5B / LEASH / OlympiadBench

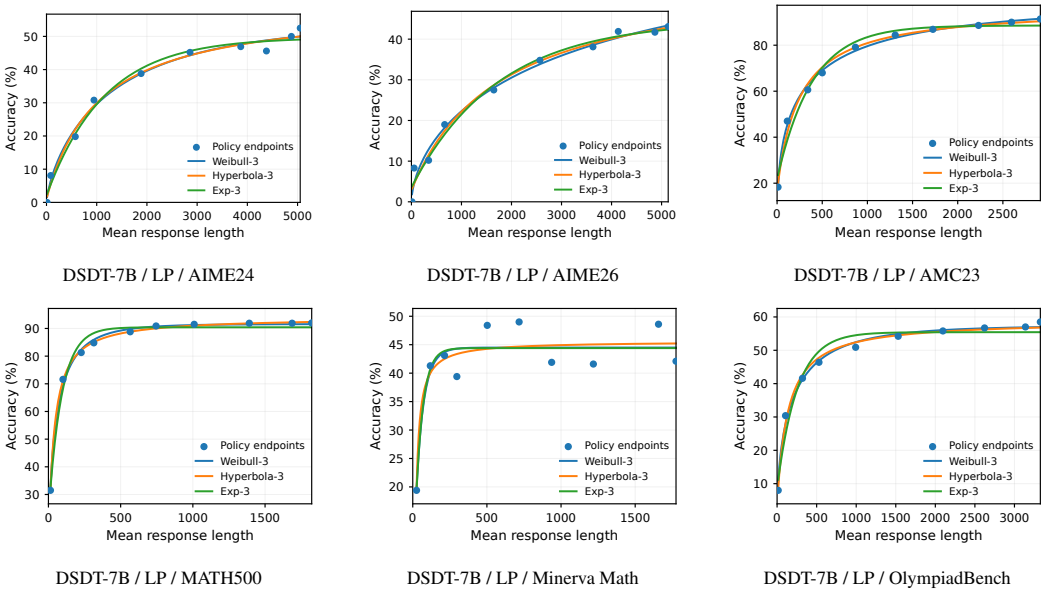
Figure 6: Math A–L endpoints and three-parameter fits for the DSDT-1.5B LEASH sweep.



DSDT-1.5B / COSINE / AIME24 | DSDT-1.5B / COSINE / AIME26 | DSDT-1.5B / COSINE / AMC23

DSDT-1.5B / COSINE / MATH500 | DSDT-1.5B / COSINE / Minerva Math | DSDT-1.5B / COSINE / OlympiadBench

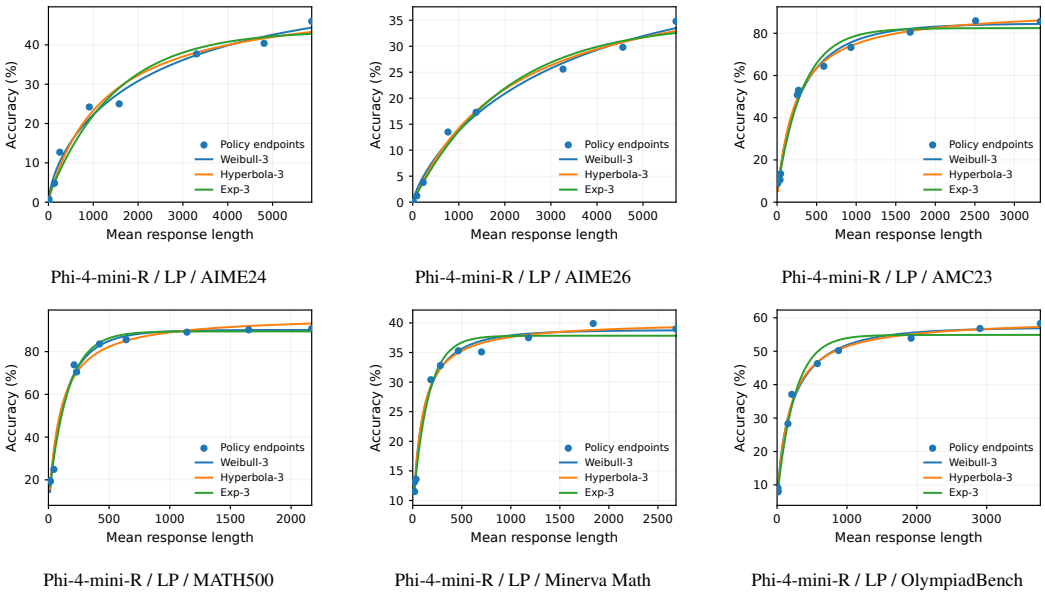
Figure 7: Math A–L endpoints and three-parameter fits for the DSDT-1.5B COSINE sweep.



DSDT-7B / LP / AIME24 | DSDT-7B / LP / AIME26 | DSDT-7B / LP / AMC23

DSDT-7B / LP / MATH500 | DSDT-7B / LP / Minerva Math | DSDT-7B / LP / OlympiadBench

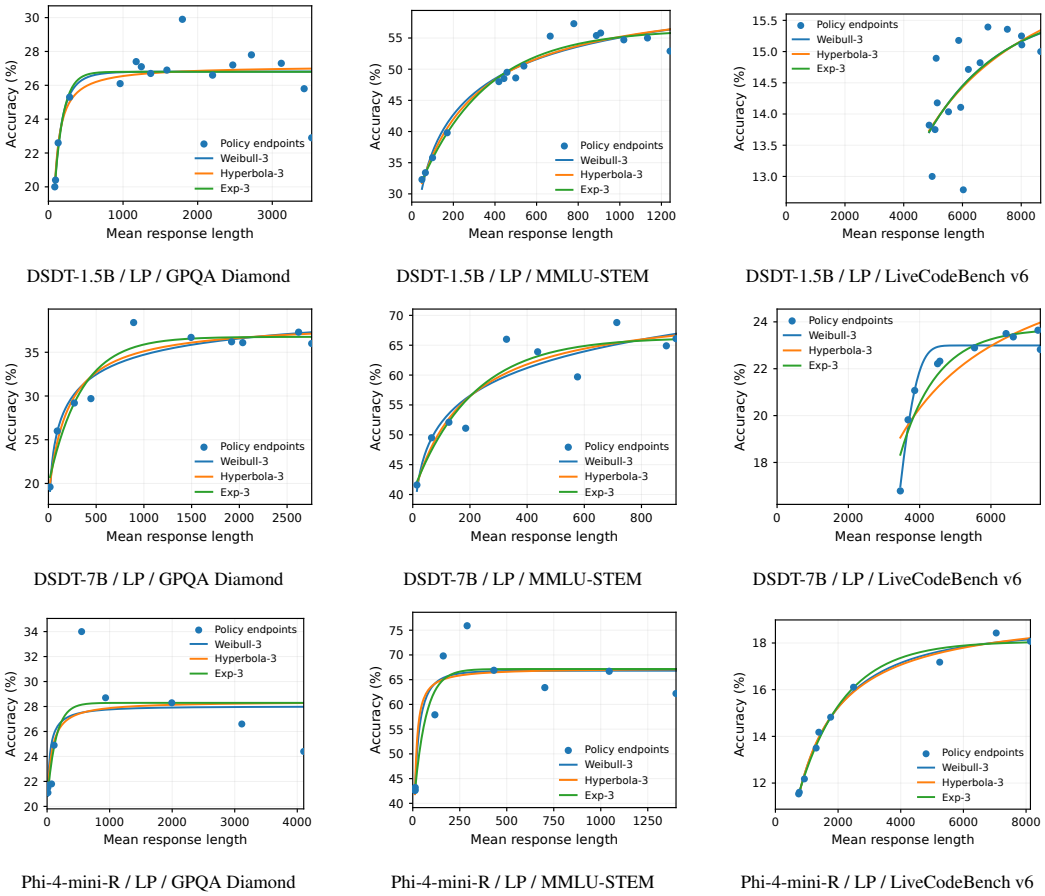
Figure 8: Math A–L endpoints and three-parameter fits for the DSDT-7B LP sweep.



Phi-4-mini-R / LP / AIME24 | Phi-4-mini-R / LP / AIME26 | Phi-4-mini-R / LP / AMC23

Phi-4-mini-R / LP / MATH500 | Phi-4-mini-R / LP / Minerva Math | Phi-4-mini-R / LP / OlympiadBench

Figure 9: Math A–L endpoints and three-parameter fits for the Phi-4-mini-R LP sweep.



DSDT-1.5B / LP / GPQA Diamond | DSDT-1.5B / LP / MMLU-STEM | DSDT-1.5B / LP / LiveCodeBench v6

DSDT-7B / LP / GPQA Diamond | DSDT-7B / LP / MMLU-STEM | DSDT-7B / LP / LiveCodeBench v6

Phi-4-mini-R / LP / GPQA Diamond | Phi-4-mini-R / LP / MMLU-STEM | Phi-4-mini-R / LP / LiveCodeBench v6

Figure 10: Multiple-choice and code A–L endpoints and three-parameter fits under LP.

D Matrix Decomposition and Held-Out Prompt/Endpoint Evaluation

Why Eq. (6) is a general matrix decomposition. Let $Q \in [0,1]^{N\times J}$ collect prompt correctness over the J policy endpoints, $Q_{ij} = q_{ij}$. Let I_J be the $J \times J$ identity matrix and $\mathbf{1} \in \mathbb{R}^J$ the all-ones vector, and define

$$H = I_J - J^{-1}\mathbf{1}\mathbf{1}^\top, \qquad R = QH. \tag{19}$$

Right multiplication by H subtracts each prompt's mean over endpoints, so R contains only how each prompt changes across trained policies. This subtraction does not limit the number of endpoint directions; each centered row may still vary freely in the $J-1$ dimensional subspace orthogonal to $\mathbf{1}$.

Let $K = \mathrm{rank}(R)$ and write the singular-value decomposition (SVD)

$$R = \sum_{r=1}^{K} \sigma_r a_r v_r^\top. \tag{20}$$

Here $\sigma_r > 0$ are singular values and $a_r \in \mathbb{R}^N$ and $v_r \in \mathbb{R}^J$ are unit left and right singular vectors. If $\alpha_i = J^{-1}\sum_j Q_{ij}$ and $s_{ir} = \sigma_r a_{r,i}$, then every observed prompt–endpoint probability has the exact representation

$$q_{ij} = \alpha_i + \sum_{r=1}^{K} s_{ir} v_{rj}. \tag{21}$$

Thus the full-rank version of Eq. (6) reproduces the finite matrix exactly when all K right singular vectors are retained. The empirical restriction is the truncation to small d. The experiment asks whether keeping only one direction, $d = 1$, predicts held-out target-prompt responses at one policy endpoint. Indexing v_r by policy endpoint also avoids assuming in advance that the direction is a function of length.

RQ3: from one unrestricted endpoint direction to a length-dependent curve. If one direction is sufficient, Eq. (6) reduces to $q_{ij} \approx \alpha_i + s_i g_j$, where g_j is unrestricted at each observed policy endpoint. RQ3 adds a second restriction: the sampled values of g should be approximated by $h(L_j;\theta)$, a function of realized response length. Let

$$h_\theta = \big(h(L_1;\theta), \ldots, h(L_J;\theta)\big)^\top.$$

After removing offset and scale, g can be any centered unit vector, whereas a candidate length-dependent direction is restricted to

$$u(\theta) = \frac{Hh_\theta}{\|Hh_\theta\|_2}. \tag{22}$$

Here $\theta = \eta$ for the exponential form and $\theta = \kappa$ for the hyperbolic form. Under the idealized exact form corresponding to Eq. (10), with $s = (s_1, \ldots, s_N)^\top$,

$$R = s(Hh_\theta)^\top, \tag{23}$$

so the centered response has rank at most one, and averaging over prompts gives the corresponding closure in Eq. (11). The implication is one-way: fitting one shared endpoint direction well does not by itself imply an exponential or hyperbolic function of response length. RQ3 therefore tests this additional requirement after RQ2 establishes how many shared endpoint directions are needed.

Error from using one direction and then restricting its shape. A centered matrix with rank one is generated by one endpoint direction shared across prompts. The connection between the RQ2 shared-dimension test and the RQ3 curve restriction therefore follows from the standard best rank-one approximation result (Eckart & Young, 1936; Golub & Van Loan, 2013). For any centered unit direction u, define

$$\mathcal{E}(u) = \min_{\gamma\in\mathbb{R}^N} \|R - \gamma u^\top\|_F^2, \tag{24}$$

where $\|\cdot\|_F$ is the Frobenius norm. Expanding the square gives

$$\|R - \gamma u^\top\|_F^2 = \|R\|_F^2 - \|Ru\|_2^2 + \|\gamma - Ru\|_2^2, \tag{25}$$

so the unique minimizer is $\gamma^\star = Ru$. With the SVD $R = \sum_r \sigma_r a_r v_r^\top$, substitution yields

$$\mathcal{E}(u) = \sum_r \sigma_r^2 \left[1 - (v_r^\top u)^2\right]. \tag{26}$$

The best unrestricted one-direction approximation is therefore $u = v_1$, with $\mathcal{E}(v_1) = \sum_{r\geq 2} \sigma_r^2$. For a direction $u(\theta)$ sampled from a one-parameter curve, let $|v_1^\top u(\theta)| = \cos\phi_\theta$ with $\phi_\theta \in [0, \pi/2]$. Then

$$\mathcal{E}(u(\theta)) - \mathcal{E}(v_1) = \sigma_1^2 \sin^2\phi_\theta - \sum_{r\geq 2} \sigma_r^2 (v_r^\top u(\theta))^2 \leq \sigma_1^2 \sin^2\phi_\theta. \tag{27}$$

The singular-value tail measures the response variation left after keeping one direction, while the angle controls the upper bound on the extra reconstruction error from requiring that direction to follow $h(L;\eta)$ or $h(L;\kappa)$. This identity describes reconstruction error on the same centered matrix. It does not bound qRMSE on held-out target prompts; the held-out cosine-similarity and qRMSE experiments test prediction on target prompts directly.

Prompt-splitting and held-out-endpoint procedure. At each prompt and policy-endpoint combination, the 16 rollouts are split into independent halves H_1 and H_2. Let $c_{ij}^{H_k}$ be the number of correct rollouts in half H_k and $n_{ij}^{H_k} = 8$ its rollout count, so $\widehat{q}_{ij}^{H_k} = c_{ij}^{H_k}/n_{ij}^{H_k}$ for $k \in \{1, 2\}$. Only prompts observed at all J endpoints are eligible; each controller–dataset group is capped at 512 prompts using a deterministic prompt-identifier rule. Eligible prompts are then partitioned deterministically by prompt identifier into disjoint source and target sets, independently of correctness or length outcomes. For RQ2, the source-prompt matrix from one rollout half uses all J policy endpoints and is centered and decomposed by SVD to obtain $v_1, v_2, \ldots$. For a target prompt and a held-out endpoint $j^\star$, only $\alpha_i, s_{i1}, \ldots, s_{id}$ in Eq. (6) are fitted using the other $J-1$ endpoints from the same rollout half. The linear prediction at $j^\star$ is clipped to $[10^{-7}, 1-10^{-7}]$ before scoring so that it is a valid Bernoulli probability. The prediction is then scored against the independent rollout half. Prompt folds and rollout halves are swapped and averaged. Because each scoring half contains eight rollouts per prompt–endpoint pair, absolute qRMSE includes finite-rollout sampling noise; predictor differences are evaluated against the same held-out targets. Because the SVD right singular vectors are orthonormal, v_2 is orthogonal to v_1 and captures the largest remaining squared variation after the first direction.

The error on individual prompts is qRMSE from Eq. (7). RQ3 uses the same source-prompt, target-prompt, and rollout separation, but its two tests use endpoint holdout differently. For the length-baseline and direction-shape comparisons in Sections 6.1–6.2, the source-prompt direction or shape may use all J endpoints; for each target prompt, only its offset and amplitude are fitted on the other $J-1$ endpoints before the held-out endpoint is scored. This directly tests whether a source-learned endpoint direction transfers across prompts at the observed policy grid. For the final dataset-accuracy transfer in Section 6.3, the held-out endpoint is also removed when learning the shape parameter from source prompts; the learned shape is then fixed while only the dataset-level offset and amplitude are fitted on the remaining endpoints. Because the normalized shapes lie in $[0, 1]$, the affine prompt fit is parameterized by its two endpoint probabilities and constrains both to $[0, 1]$. The exponential search uses $\eta \in [10^{-4}, 20]$; the hyperbolic search uses $\log\kappa \in [-6, 8]$.

Supplementary checks. The randomized smoothness check preserves each prompt's coefficient magnitudes across polynomial orders on the observed endpoint grid while breaking cross-prompt alignment; Appendix E gives the construction. It uses 1,000 randomization replicates per eligible group. The 95% intervals in Table 3 use 5,000 prompt-resampling replicates within each controller–dataset combination.

E Additional RQ2 Results

E.1 Held-Out Gains for Each Math Combination

Table 10 reports the values summarized by Figure 3.

Controller	AIME24	AIME26	AMC23	MATH500	Minerva Math	OlympiadBench	Mean
$d = 0 \to 1$ *qRMSE gain (pp)*							
LP	7.74	6.77	10.71	7.19	2.64	9.16	7.37
LEASH	8.45	6.53	5.07	6.68	3.20	8.19	6.35
COSINE	5.18	4.05	4.55	3.44	-0.12	3.97	3.51
$d = 1 \to 2$ *qRMSE gain (pp)*							
LP	-0.24	0.04	3.15	-0.41	1.33	0.43	0.72
LEASH	-0.68	-0.63	-0.71	-4.56	-1.40	-2.07	-1.67
COSINE	-0.93	-2.08	-3.86	-5.89	-1.62	-3.59	-2.99

Table 10: Held-out prompt-correctness RMSE change when increasing the number of shared endpoint directions. Positive values mean lower qRMSE after adding the direction.

E.2 Randomized Smoothness Check

For each controller–dataset combination, we map endpoint lengths to $t_j = 2(L_j - L_{\min})/(L_{\max} - L_{\min}) - 1$, evaluate Legendre polynomials of increasing degree on this exact grid, and QR-orthonormalize them. After centering each prompt, we express its endpoint profile in the nonconstant basis. A null replicate independently flips each prompt's coefficient sign at each polynomial order, with the same sign mask applied to both rollout halves. It therefore preserves each prompt's coefficient magnitude at every smoothness order while destroying cross-prompt alignment at the same endpoints.

Under this null, the fraction of centered response variation captured by the first right singular vector exceeds the randomized-data 97.5th percentile in 17 of the 18 controller–dataset combinations across the six math benchmarks, and the held-out NLL gain from $d = 0$ to $d = 1$ exceeds it in 13. The cosine similarity between independently learned first directions is not consistently above this threshold. Thus, the strongest evidence for the first shared endpoint direction comes from captured variation and held-out prediction.

E.3 NLL Check

NLL is the average negative log-likelihood of the independent held-out rollout outcomes under the predicted Bernoulli correctness probabilities. It scores the same held-out prompt–endpoint predictions as qRMSE. Table 11 shows that the first direction lowers median NLL for LP and LEASH but raises it for COSINE, while the second direction generally worsens NLL. We therefore use qRMSE as the primary error measure.

Controller	$d = 0$	$d = 1$	$d = 2$	Gain $0 \to 1$	Gain $1 \to 2$
LP	0.362	0.311	0.321	0.050	-0.010
LEASH	0.395	0.340	0.418	0.055	-0.079
COSINE	0.335	0.369	0.536	-0.034	-0.167

Table 11: Median held-out NLL across the six math datasets for the same held-out prompt procedure as Table 10. NLL is dimensionless and averaged per rollout; positive gain means lower NLL after adding the direction.

F Additional RQ3 Results and Controller Checks

F.1 Length-Based Baselines on Multiple-Choice and Code

Table 12 summarizes the DSDT-1.5B LP, LEASH, and COSINE results for each task family, using the same controller set as in the main math analysis. The length-based comparison uses the same source-prompt, target-prompt, rollout-half, and held-out-endpoint protocol as RQ2. Linear length uses centered normalized response length, length rank keeps only endpoint ordering, and the monotone fit learns from source prompts a shared endpoint direction whose values can only increase or decrease with response length. Table 12 shows that multiple-choice broadly follows the math result, whereas code is weaker and mixed: linear length and length rank are not worse than the learned direction in qRMSE, the source-fitted monotone direction is worse, and all three have substantially lower cosine alignment than in math or multiple-choice. The length-based interpretation is strongest on math, is also supported by multiple-choice, and is weaker on code.

Task	Length-based comparison	qRMSE difference (pp)	Cosine similarity
Math	Linear length	$+1.35$	0.854
Math	Length rank	$+1.44$	0.863
Math	Monotone fit (source prompts)	-0.015	0.999
Multiple-choice	Linear length	$+0.96$	0.888
Multiple-choice	Length rank	$+0.88$	0.898
Multiple-choice	Monotone fit (source prompts)	-0.003	0.996
Code	Linear length	-0.077	0.561
Code	Length rank	-0.13	0.538
Code	Monotone fit (source prompts)	$+0.11$	0.729

Table 12: Length-based comparisons on math, multiple-choice, and code. Entries are medians across eligible DSDT-1.5B LP/LEASH/COSINE cells within each task scope. Positive qRMSE differences favor the first shared endpoint direction learned from source prompts. Multiple-choice broadly follows the math result, while code shows weaker and mixed alignment rather than the same clear pattern.

F.2 Length Curve and Dataset-Accuracy Prediction

Table 13 summarizes the 18 DSDT-1.5B controller–dataset combinations across the six math benchmarks used in the main RQ3 analysis and collects the corresponding length-curve and dataset-accuracy results. Figure 11 gives the corresponding cosine similarity, held-out prompt qRMSE, and RMSE when the curve learned from prompts is used for dataset accuracy. For the unrestricted row in Table 13, cosine similarity compares independently learned source and target profiles; for the exponential and hyperbolic rows, it compares the source-learned shape-constrained direction with the unrestricted target profile. All reported cosines are absolute because the sign of a one-dimensional profile can be absorbed into the prompt-specific amplitude.

Endpoint representation	Cosine similarity	ΔqRMSE (pp)	Dataset-accuracy RMSE (pp)
Unrestricted endpoint profile	0.960	0.00	–
Exponential (η)	0.974	-0.03	2.91 / 3.47
Hyperbolic (κ)	0.974	-0.04	3.51 / 3.45

Table 13: RQ3 curve comparison on the 18 DSDT-1.5B math controller–dataset cells. Entries are medians across cells. ΔqRMSE is candidate-curve qRMSE minus qRMSE of the unrestricted endpoint profile. The last column gives RMSE for a direct fit to dataset accuracy, followed by RMSE when the shape parameter is learned from source prompts and kept fixed.

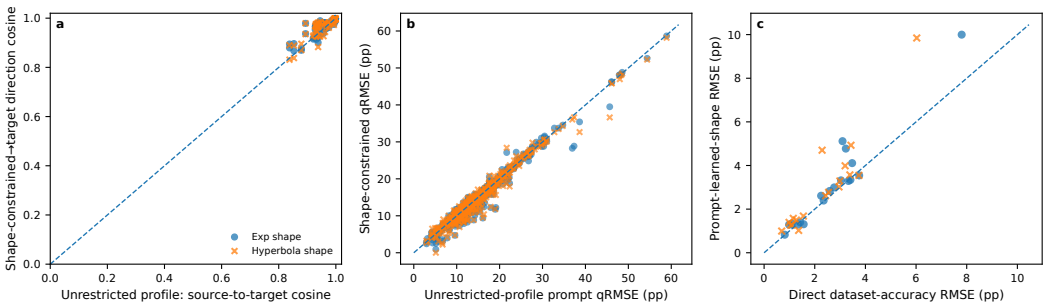


Figure 11: RQ3 length-curve and dataset-accuracy results on DSDT-1.5B math. The first two panels show cosine similarity and held-out prompt qRMSE over the endpoint-grid transfer test. The dataset-accuracy panel plots group-level RMSE over interior held-out endpoints for the prompt-learned shape versus the direct fit; Table 13 reports the corresponding per-combination summaries from the full held-out protocol.

F.3 Prompt-Dependent Controller Checks: LP-C and LP-D

LP-C and LP-D are the prompt-dependent LP ablations defined in Eqs. (16)–(18) and are used only as stress tests of the reference LP trajectory and shared-direction result. LP-C gates the continuous length cost by response correctness, whereas LP-D weights that cost by the current on-policy solve rate of each prompt. Both sweeps use the same correctness-only $\beta = 0$ policy as a common anchor; when a separate controller-specific zero-setting run was not trained, we reuse that policy rather than treat it as an additional intervention run. Each nonzero ablation policy endpoint is compared with the reference LP A–L curve at the same realized mean response length. The *observed LP length interval* runs from the minimum to the maximum realized LP length. Residuals in dataset accuracy use interpolation between neighboring LP policy endpoints within this interval, and residuals for individual prompts apply the same matched-length construction prompt by prompt. When enough policy endpoints are available, the RQ2 $d = 0$ versus $d = 1$ held-out prompt comparison is rerun on the ablation sweep. Therefore, these ablations test whether changing the prompt-dependent controller setting changes either the matched-length A–L behavior or the main shared endpoint direction. The shared $\beta = 0$ policy endpoint participates in the prompt-by-endpoint matrix but is not counted among the nonzero ablation endpoints.

For each dataset and evaluation setting, we calibrate two reference thresholds by leaving out each interior LP endpoint and interpolating it from its two neighboring LP endpoints. The dataset-level threshold is the largest absolute accuracy interpolation residual across these checks, and the prompt-level threshold is the largest prompt RMSE across the corresponding prompt-wise checks. In Tables 14 and 15, the displayed dataset residual and prompt RMSE are medians over nonzero ablation endpoints. We also report how many nonzero ablation endpoints exceed both reference thresholds. The $0 \to 1$ gain is $\mathrm{qRMSE}(d = 0) - \mathrm{qRMSE}(d = 1)$, so positive values favor one shared direction.

Math tasks. All LP-C and LP-D math endpoints lie within the observed LP length interval. No math dataset has more than one nonzero ablation endpoint exceeding both reference thresholds. LP-C has one such endpoint on AIME24; all other math combinations have none, although some Minerva Math endpoints exceed only one of the two thresholds. The first shared endpoint direction

improves held-out qRMSE in every math ablation combination with prompt-level evaluation. The shared direction therefore remains predictive under both ablations.

Method	Dataset	Dataset residual (pp)	Prompt RMSE (pp)	Gain $0 \to 1$ qRMSE (pp)	Endpoints above both thresholds
LP-C	AIME24	2.58	11.43	7.01	1
LP-C	AIME26	0.54	7.76	5.59	0
LP-C	AMC23	0.96	11.90	2.83	0
LP-C	MATH500	0.62	10.24	3.70	0
LP-C	Minerva Math	2.41	20.37	0.73	0
LP-C	OlympiadBench	0.58	13.15	5.05	0
LP-D	AIME24	0.72	7.98	7.17	0
LP-D	AIME26	1.51	6.21	7.14	0
LP-D	AMC23	1.66	11.94	6.99	0
LP-D	MATH500	1.05	11.19	5.78	0
LP-D	Minerva Math	1.70	22.98	2.97	0
LP-D	OlympiadBench	1.06	14.43	7.26	0

Table 14: LP-C/LP-D matched-length checks on math. Dataset and prompt residuals are measured against interpolation on the reference LP sweep at the same realized response length. The final column counts nonzero ablation endpoints exceeding both reference thresholds.

Multiple-choice and code. LP-C on MMLU-STEM is the only evaluated ablation for which at least two nonzero endpoints exceed both the dataset-level and prompt-level reference thresholds. No LP-D combination has more than one such endpoint. On GPQA Diamond, some endpoints exceed only one of the two thresholds for both LP-C and LP-D. For LiveCodeBench v6, the dataset-level residuals remain within the LP reference threshold for both methods. These code runs were evaluated only at dataset level, so we do not draw prompt-level conclusions from them.

Task	Method	Dataset	Dataset residual (pp)	Prompt RMSE (pp)	Gain $0 \to 1$ qRMSE (pp)	Endpoints above both thresholds	Status
Multiple-choice	LP-C	GPQA Diamond	3.48	15.31	0.82	0	Mixed
Multiple-choice	LP-C	MMLU-STEM	2.85	18.22	4.50	3	Replicated
Multiple-choice	LP-D	GPQA Diamond	1.54	14.00	0.60	0	Mixed
Multiple-choice	LP-D	MMLU-STEM	1.79	14.66	3.53	1	One endpoint
Code	LP-C	LiveCodeBench v6	0.83	–	–	0	Within
Code	LP-D	LiveCodeBench v6	0.46	–	–	0	Within

Table 15: LP-C/LP-D matched-length checks on multiple-choice and code. Per-prompt ablation evaluations are unavailable for LiveCodeBench, so code is not used for a claim about individual prompts. Status labels are defined in Appendix F.3.

G Multiple-Choice and Code Results for RQ2 and RQ3

Our main prompt-level analysis uses the six math benchmarks. Figure 12 reports the corresponding DSDT-1.5B results with LP, LEASH, and COSINE for math, multiple-choice, and code, while Figure 13 reports the LP math model extension together with the multiple-choice and code task extensions. Multiple-choice broadly follows the math pattern, whereas code retains some shared-direction structure but shows weaker and mixed evidence for the same length-based interpretation.

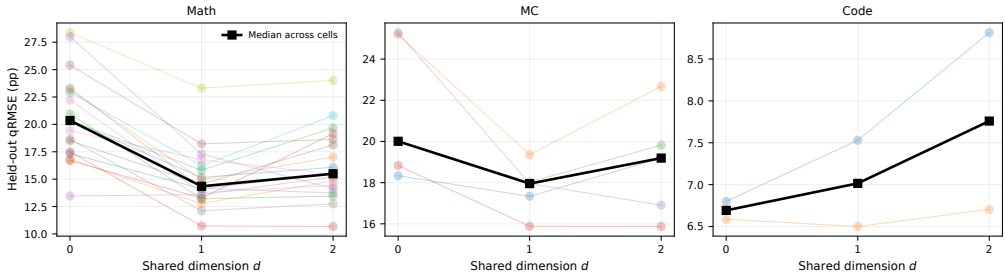


Figure 12: RQ2 held-out qRMSE as the number of shared endpoint directions increases from $d = 0$ to $d = 2$ for DSDT-1.5B with LP, LEASH, and COSINE. The math panel uses the same 18 controller–dataset combinations as the main analysis; the other panels show the multiple-choice and code task extensions. MC denotes multiple-choice.

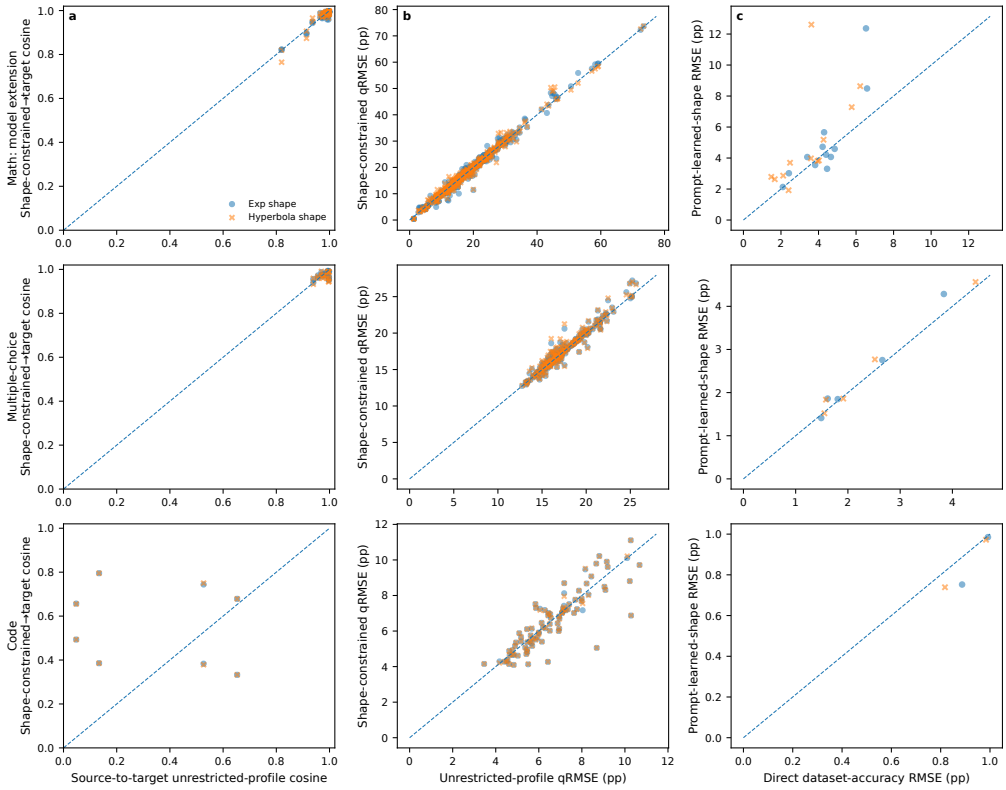


Figure 13: RQ3 shape-transfer diagnostics: cosine similarity, held-out prompt qRMSE, and RMSE when the length curve learned from prompts is reused for dataset accuracy. The top row is the LP math model extension (DSDT-7B and Phi-4-mini-R); the lower rows are the DSDT-1.5B LP/LEASH/COSINE multiple-choice and code task extensions.

H Additional Evaluations of the Fitted A–L Curve

Accuracy differences between policy endpoints. To keep units explicit, let $F_f(L) = 100\,\widehat{A}_f(L)$ denote the fitted accuracy in percentage points for each primary three-parameter RQ1 family $f \in \{\text{Exp-3}, \text{Weibull-3}, \text{Hyperbola-3}\}$, and define

$$g_f(L) = \frac{dF_f(L)}{dL},$$

the local fitted accuracy return in percentage points per response token. Because independently trained policies provide no pointwise ground-truth derivative, we validate only its integrated finite change. We hold out endpoint j, fit each family to the remaining $J-1$ endpoints, let $\nu(j)$ be the retained endpoint nearest to j in realized length, define the observed change $\Delta A_j^{\text{pp}} = 100(A_j - A_{\nu(j)})$, and predict

$$\widehat{\Delta A}_{f,j}^{\text{pp}} = \int_{L_{\nu(j)}}^{L_j} g_f(L)\,dL = F_f(L_j) - F_f(L_{\nu(j)}). \tag{28}$$

The reported prediction is the median of the three fixed family predictions; no family is selected using the held-out endpoint. RMSE and MAE are computed against ΔA_j^{pp}, and sign agreement is the fraction of pairs for which predicted and observed finite changes have the same sign, with an exactly zero observed change counted as agreement. Because the endpoints are independently trained policies, $g_f(L)$ describes the fitted sweep and is not a causal value for deleting tokens from one response. Table 16 reports the math summary, and Figure 14 extends the comparison to multiple-choice and code.

Scope	RMSE (pp)	MAE (pp)	Sign agreement
Math	3.11	1.95	83.3%

Table 16: Leave-one-endpoint-out finite accuracy-change validation on the main RQ1 math sweeps. Each prediction is the median of Exp-3, Weibull-3, and Hyperbola-3, with every family fitted without the held-out endpoint.

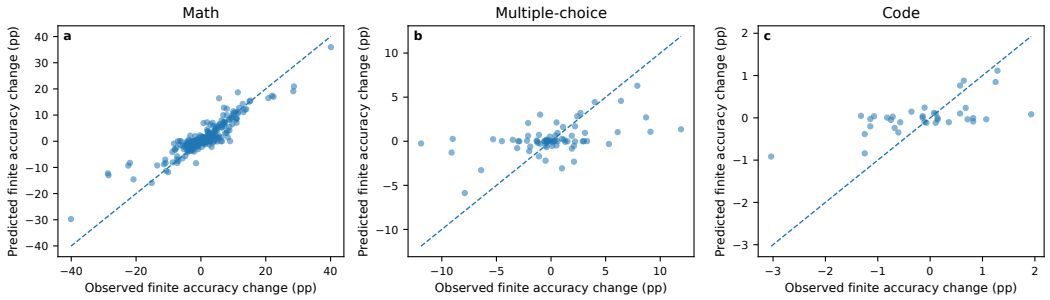


Figure 14: Observed versus predicted leave-one-endpoint-out finite accuracy changes on math, multiple-choice, and code. Predictions are medians across Exp-3, Weibull-3, and Hyperbola-3, each fitted without the held-out endpoint.

Reconstructing the curve from retained endpoints. Table 17 reports the math stable-budget summary stated in the main text.

For each LP sweep with an identified $\beta = 0$ policy and largest-β policy, those two endpoints are fixed. For anchor budget B, equally spaced target lengths are placed over the realized-length interval between these endpoints, and interior targets are matched one-to-one to distinct observed policy endpoints by minimizing total absolute target–endpoint length mismatch. Exp-3, Weibull-3, and Hyperbola-3 are fitted only to the retained endpoints, their fixed pointwise median defines the reconstructed curve, and every omitted observed endpoint is scored; omitted endpoints are never used to choose a family. nRMSE uses the full observed sweep's accuracy span. If a trained endpoint lies outside the realized-length interval between the two fixed β endpoints, it remains in the score;

Criterion	Median stable B	90th percentile	Curves covered
nRMSE $\leq 10\%$	3	7.4	14/18
nRMSE $\leq 5\%$	9	13.4	9/18

Table 17: Endpoint-fixed reconstruction on math. "Curves covered" counts sweeps with a finite stable budget under the stated threshold; stable-budget summaries are computed over those covered sweeps.

this is within-sweep extrapolation rather than extrapolation beyond the observed policy sweep. The stable-budget definition in Section 4.3 is then applied to these held-out errors.

Endpoint-fixed reconstruction requires identifying the $\beta = 0$ and largest-β policies. This mapping is available for the math and multiple-choice sweeps but not for LiveCodeBench v6, so code is excluded from the endpoint-fixed reconstruction analysis. Figure 15 shows the budget–error curves for math and multiple-choice. Table 18 reports the corresponding stable budgets for multiple-choice; as in Table 17, its median and 90th-percentile budgets are computed over covered curves only.

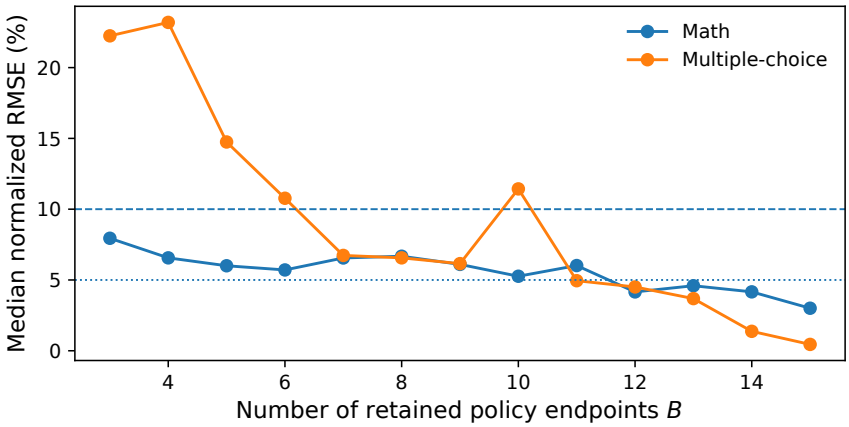


Figure 15: Endpoint-fixed reconstruction error as anchor budget increases. Each point is the median over curves for which that budget is feasible. The main text reports math; multiple-choice is shown here as an additional task.

Criterion	Median stable B	90th percentile	Curves covered
nRMSE $\leq 10\%$	8	10.2	5/6
RMSE ≤ 2 pp	6	8.0	5/6
nRMSE $\leq 5\%$	11	12.6	3/6

Table 18: Endpoint-fixed reconstruction on multiple-choice. The columns use the stable-budget definition in Section 4.3.