BEHAVIORAL RESPONSE SPECTROSCOPY: PRE-RL RESPONSES REVEAL SHARED STRUCTURE IN RL-INDUCED BEHAVIORAL CHANGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) improves a target reward but often changes other properties of a model's responses as well. Can any of these accompanying changes be anticipated before training? We show that variation already present in the starting model contains a measurable signal of later change. Behavioral Response Spectroscopy (BRS) uses multiple answers to the same prompt to measure which response properties tend to accompany higher reward. We then compare these pre-RL directions with the behavioral changes produced by actual RL across ten models and four tasks. Our strongest test asks whether this structure transfers across objectives: we exclude the training reward from the measured directions and use separate prompts for evaluation. On two reasoning tasks, a two-dimensional space built from other reward directions captures 50.7% of post-RL displacement energy across five Qwen2.5 models and 73.2% across five other models, compared with 24.8% and 36.2% for principal components computed from the same responses without reward information. Matched controls reveal a prominent length-associated behavioral mode: multiple response properties co-vary with length, while different rewards can orient this mode toward either shorter or longer answers. This directional regularity is not confined to the original surface features: in an embedding-based behavioral representation, alignment remains positive after pre-RL length adjustment, although the relative advantage of BRS over PCA varies across representations and model cohorts. Overall, pre-RL responses already reveal a transferable component of how RL will reshape behavior.

1 INTRODUCTION

Reinforcement learning changes more than the quantity it is asked to optimize. Training a language model for correctness, for example, can improve its task score while also changing response length, repetition, or expressions of uncertainty. Such accompanying changes matter because the optimized reward records only one part of what training does. Reward overoptimization and broader behavioral shifts make this gap increasingly important (Gao et al., 2023; Rafailov et al., 2024; Betley et al., 2026; Denison et al., 2024). Yet these changes are usually characterized only after training. We ask a prospective question: *can some of the behavioral changes induced by RL be anticipated from the starting model, before any RL takes place?*

Our starting point is the variation already present in stochastic responses. Given the same prompt, a model may produce a concise answer, a longer derivation, or a response that checks and revisits its reasoning. These properties need not vary independently. If higher-reward answers also tend to be shorter, use more verification language, or repeat conclusions more often, then favoring those answers may select an entire bundle of properties rather than only the rewarded quantity. This suggests that the starting policy may already contain directions along which later training will move its behavior. The challenge is to distinguish such reward-relevant directions from variation that is merely common.

We introduce **Behavioral Response Spectroscopy (BRS)** to measure this structure. For each prompt, BRS compares multiple sampled responses and asks which behavioral properties systematically accompany higher reward. After within-prompt centering, reward–behavior covariance defines

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.

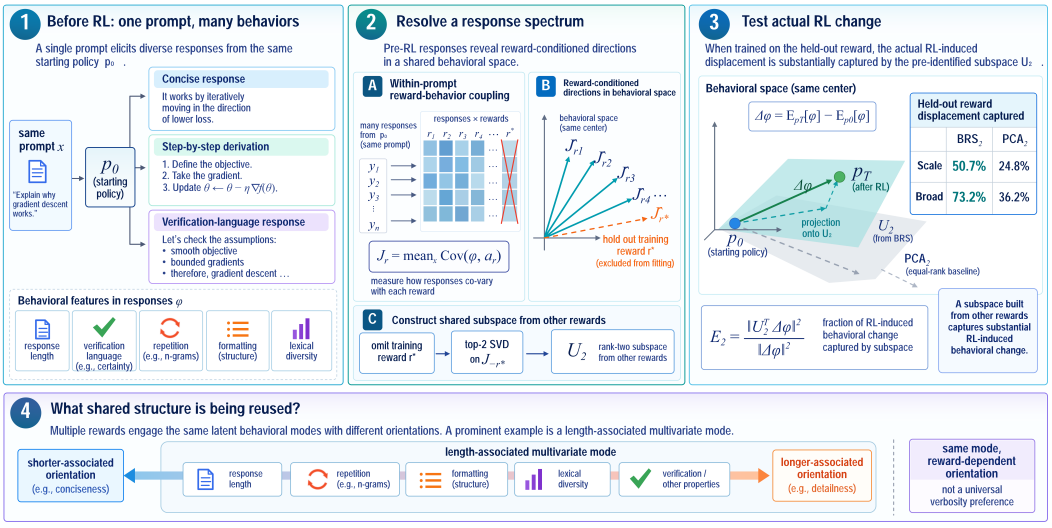

Figure 1: **Pre-RL responses reveal shared structure in later behavioral change.** Repeated responses define reward-conditioned directions. The primary transfer test excludes the training reward, builds a rank-two space from the remaining probes, and measures its capture of actual RL displacement on disjoint evaluation prompts. Same-reward endpoint alignment is a separate directional test; checkpoint re-measurement is a secondary extension (Section 6).

a direction in behavior space; repeating the measurement for several reward probes produces a response spectrum. A local distributional-reweighting identity motivates the construction: shifting probability toward higher-advantage responses changes expected behavior, to first order, in the covariance direction. This suggests a concrete empirical question for parameter training: *how much of this pre-RL response geometry is preserved in the behavioral displacement produced by actual RL?*

We test this correspondence across ten language models and four tasks. In a first endpoint test, a reward's pre-RL direction is compared with the subsequent change in behavioral properties outside the optimized objective. Positive alignment is common across both reasoning cohorts and also appears in code generation and summarization. Our stronger test then removes the training reward itself. For each reasoning condition, we construct a rank-two subspace using only the remaining reward probes and evaluate it on separate prompts. Across five Qwen2.5 models, this held-out-reward subspace captures **50.7%** of post-training behavioral displacement energy, compared with **24.8%** for an equal-rank PCA baseline using the same responses without reward information. Across five additional models, the corresponding values are **73.2%** and **36.2%**. Every model has a positive median paired BRS–PCA gain in the original surface-feature representation, and reward-permutation and covariance-shaped nulls test explanations based on generic low-dimensional variation.

What shared structure do these rewards engage? A prominent answer is **response length—but not simply as a scalar preference for verbosity**. Properties that co-vary with length form a multivariate behavioral direction, and rank-matched controls containing this direction capture a large fraction of the same post-training displacement. Its orientation is reward-dependent: different rewards can align the associated behavioral bundle with either longer or shorter answers. In several correctness runs, for example, the frozen correctness score increases while mean response length decreases. We therefore view length as an observable index of an important shared behavioral mode.

We next ask what remains beyond this length-associated mode. Replacing the original surface features with a common embedding-based behavioral representation preserves positive endpoint alignment, including after removing fitted pre-RL associations with length. The broader directional regularity is therefore visible under a substantially different representation. After length adjustment, however, the relative advantage of BRS over behavior-only PCA varies across representations and model cohorts. Taken together, the results identify a transferable component of RL-induced behavioral change before training, with length providing an interpretable shared mode and the remaining structure depending more strongly on how behavior is represented.

As a secondary analysis, we re-measure the response geometry as the policy evolves. Refreshing BRS at intermediate checkpoints improves prediction of the next training interval in the original trajectories, while extension settings show more variable gains.

Our contributions are:

- **Pre-RL response variation anticipates later behavioral change.** We introduce BRS, a measurement of reward-conditioned behavioral directions from repeated responses, and test these directions against changes produced by actual RL.
- **The measured structure transfers across held-out rewards.** On two reasoning tasks across ten models, rank-two spaces constructed without the training reward capture substantially more post-RL displacement than behavior-only PCA, using disjoint prompts for measurement and evaluation.
- **Length indexes an interpretable shared behavioral mode.** Matched controls show that a substantial part of the transferable structure is a multivariate bundle associated with response length, whose orientation varies with reward. An embedding-based representation retains positive directional alignment after pre-RL length adjustment, while residual advantages over PCA depend on representation and model cohort.

2 From Response Variation to Behavioral Change

2.1 Why reward–behavior covariance gives a direction

Let $p_0(y \mid x)$ be the starting policy, $r(y)$ a reward, and $\phi(y) \in \mathbb{R}^d$ observable response features. The policy may already be instruction-tuned. At a fixed prompt, a KL-regularized improvement with advantage a solves

$$q_\eta = \arg\max_q \{\mathbb{E}_q[a] - \eta^{-1}\mathrm{KL}(q\|p_0)\}, \qquad q_\eta(y \mid x) \propto p_0(y \mid x)e^{\eta a(y,x)}. \tag{1}$$

The tilt gives higher-advantage responses more probability. Expanding their expected features yields the local response identity

$$\mathbb{E}_{q_\eta}[\phi \mid x] - \mathbb{E}_{p_0}[\phi \mid x] = \eta\,\mathrm{Cov}_{p_0}(\phi, a \mid x) + O(\eta^2). \tag{2}$$

Covariance measures which behaviors accompany a small objective improvement. Actual RL changes shared parameters: its score kernel can preserve or rotate this direction and affects unsampled answers and other prompts (Jacot et al., 2018; Tomihari, 2026). Appendix B derives this parameter-space response. Whether coupling survives finite training is empirical; BRS predicts direction, while step size, duration, and parameter coupling also determine magnitude.

Appendix C.4 illustrates the estimator using twelve stored answers to an ARC question and their reward, length, repetition, and lexical-diversity contributions.

For each of N prompts, sample K answers and form $a_{ik} = (r_{ik} - \bar{r}_i)/(s_{r,i} + \epsilon)$. With within-prompt feature mean $\bar{\phi}_i$, the sample response is

$$\widehat{J}_r = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}\sum_{k=1}^{K}(\phi(y_{ik}) - \bar{\phi}_i)a_{ik}. \tag{3}$$

A positive coordinate of $\widehat{J}_r$ means that higher-reward answers tend to have more of that feature; a negative coordinate means they tend to have less. Prompt centering isolates variation among answers to the same question. Reward standardization gives that variation a common scale; constant-reward groups contribute zero. Stacking reward-specific directions gives $J = [\widehat{J}_{r_1}, \ldots, \widehat{J}_{r_m}]$. Raw-reward covariance $G_r = \mathbb{E}_x[\mathrm{Cov}_{p_0}(\phi, r \mid x)]$ omits advantage normalization. Appendix B explains this distinction and its consequences for reward mixtures.

2.2 Why different rewards can share a behavioral mode

A simple factor model makes the shared-direction idea concrete. At a fixed prompt, suppose $\phi = Az + \varepsilon$ and $r = b_r^\top z + \xi$, with residuals uncorrelated with each other and with z, and $\mathrm{Cov}(z \mid x) =$

$\mathrm{diag}(\lambda_1, \ldots, \lambda_q)$. Then

$$\mathrm{Cov}(\phi, r \mid x) = \sum_{j=1}^{q} \lambda_j b_{r,j} A_{:j}. \tag{4}$$

Columns of A describe shared feature patterns; reward loadings b_r select their mixtures. A dominant contribution produces a shared direction with reward-dependent sign and strength, while secondary factors yield reward-specific components. We do not fit latent factors; this model motivates observable covariance and displacement comparisons.

We use the *length-covariance direction* (Length) as such an index. Let $\ell = \widehat{J}_{\log(1+L)}$, where L is generated-token count, and define

$$\widehat{J}_r^{\mathrm{length}} = \mathrm{sign}((\widehat{J}_r)_{\mathrm{length}})\ell. \tag{5}$$

The vector describes how all measured properties co-vary with length. The reward contributes only its pre-RL orientation; rewards favoring shorter answers reverse it. Full BRS retains the complete reward-conditioned vector. Comparing the two asks how much of that vector's predictive content is shared across rewards.

Length alignment, transfer from other rewards, and length-regression residuals test distinct aspects of this structure: a transferable shared component can leave little predictable variation after removal. The feature map, probes, and scaling jointly define the spectrum.

2.3 Comparing predicted and realized off-target change

For a trained policy p_T, we measure $\Delta\phi = \mathbb{E}_{p_T}[\phi] - \mathbb{E}_{p_0}[\phi]$ and score $\cos(M\widehat{J}, M\Delta\phi)$ under a fixed off-target mask M. Correctness training excludes accuracy. Reasoning structure rewards exclude log length, short-answer indicators, step markers, and equation density. The retained coordinates measure the changes that accompany improvement in the optimized feature. Lexical diversity is additionally removed in a sensitivity test. The surface endpoint inventory uses the retained features in their original units; the cross-fitted subspace and embedding tests use discovery-set SD scaling. Cosine measures direction; feature changes provide magnitude evidence.

Top-3-of-12 selection (abbreviated BoN-3 in tables and figures) supplies a same-bank baseline: the feature shift obtained by retaining the three highest-reward responses from each group of twelve, with uniform treatment of ties. Selection also reweights the response distribution (Beirami et al., 2025); it tests whether hard selection captures the same behavioral tendency as the covariance measurement.

2.4 Models, tasks, and evaluation protocol

Table 1 spans ten models and four tasks: GSM8K and ARC reasoning (Cobbe et al., 2021; Clark et al., 2018), MBPP code generation (Austin et al., 2021), and CNN/DailyMail summarization (Hermann et al., 2015). Reasoning uses correctness, equation-density, and step-density training rewards; the native tasks each use quality and structure rewards. This gives 100 model–task–reward conditions with three training seeds each. Correctness uses a frozen task-specific answer parser. Its numeric decimal-handling limitation (Appendix A.5) means GSM8K scores reflect that scorer, rather than independently verified numerical correctness.

We organize the reasoning study into two non-overlapping groups. **Scale** contains all five Qwen2.5 sizes: 3B, 7B, 14B, 32B, and 72B. **Broad** contains Phi-3.5-mini, SmolLM2-1.7B, OLMo-1B, OLMo-7B, and Mistral-Nemo-12B. Each group has 30 conditions. Scale measures variation within the Qwen family, while Broad measures variation across the other model checkpoints.

The common configuration uses 200 GRPO steps (Shao et al., 2024), rank-16 LoRA (Hu et al., 2022), learning rate 8×10^{-6}, KL coefficient 0.02, and training groups of eight. Measurement uses 256 prompts and twelve answers per prompt, temperature 0.9, top-p 0.95, and a 384-token generation cap. Reasoning, code, and summarization use 12, 11, and 7 features, respectively (Appendices A and H).

Condition-level comparisons cover all 60 reasoning conditions and the 20 native-task conditions with matched raw banks for Phi-3.5-mini, Qwen2.5-7B/14B, OLMo-7B, and Mistral-Nemo-12B.

Table 1: **Endpoint alignment across ten models and four tasks.** Median cosine across reward conditions after averaging three seeds: three rewards for GSM8K/ARC, two for MBPP/CNN. F: Full BRS; L: Length; B: BoN-3. Task-specific masks exclude optimized coordinates. Task medians aggregate ten model entries; model medians aggregate twelve task–method entries. Model medians and the bottom-right summary describe the mixed-method inventory, rather than BRS alone.

	GSM8K			ARC			MBPP			CNN/DailyMail			Model median
Model	F	L	B	F	L	B	F	L	B	F	L	B	
Scale													
Qwen2.5-3B	0.201	0.532	0.135	0.748	0.670	0.679	0.574	0.706	0.091	0.778	0.690	0.641	**0.656**
Qwen2.5-7B	0.623	0.037	0.816	-0.181	0.180	0.124	0.100	0.610	-0.032	0.940	0.994	0.899	**0.395**
Qwen2.5-14B	0.728	0.363	0.725	0.570	0.633	0.455	0.159	0.392	0.259	0.539	0.385	0.362	**0.424**
Qwen2.5-32B	0.408	0.231	0.395	0.592	0.288	0.625	0.396	0.578	-0.028	0.682	0.742	0.553	**0.481**
Qwen2.5-72B	0.876	0.586	0.793	0.326	0.572	0.514	0.688	0.624	0.163	0.714	0.882	0.691	**0.656**
Broad													
Phi-3.5-mini	0.454	0.727	0.324	0.451	0.957	0.558	0.413	0.505	-0.012	0.756	0.854	0.733	**0.532**
SmolLM2-1.7B	0.990	0.987	0.972	0.932	0.900	0.941	0.812	0.931	0.447	0.972	0.956	0.909	**0.937**
OLMo-1B	0.970	0.905	0.962	0.485	0.936	0.653	0.621	0.842	0.214	0.884	0.936	0.791	**0.863**
OLMo-7B	0.885	0.910	0.716	0.793	0.945	0.737	0.058	0.319	-0.118	0.258	0.841	0.284	**0.727**
Mistral-Nemo-12B	-0.074	0.294	0.092	0.294	-0.145	0.416	0.742	0.893	-0.164	0.939	0.960	0.961	**0.355**
Task median	**0.676**	**0.559**	**0.721**	**0.528**	**0.652**	**0.592**	**0.494**	**0.617**	**0.040**	**0.767**	**0.868**	**0.712**	0.634

The full endpoint table retains the ten-model inventory. Training uses separate training-split prompts. Cross-fitting divides measurement prompts into discovery (fitting geometry) and evaluation (measuring displacement); the OLMo-7B/Mistral-Nemo-12B reasoning and matched native banks additionally separate generation pools.

We average seeds and prompt folds within each condition, then report medians across conditions. Table 1 additionally aggregates rewards within model–task cells before computing task medians. Paired gains are medians of within-condition differences. Condition-level intervals jointly resample prompts and seeds; cohort intervals resample five whole model clusters, retaining their tasks and rewards. We report distributions alongside summaries (Agarwal et al., 2021); Appendix C specifies uncertainty and provenance.

3 Pre-RL Directions Anticipate Off-Target Change

3.1 Behavioral coupling predicts more than reward improvement

We first ask whether the direction associated with a reward before training aligns with later change outside its objective. Full BRS has positive alignment in 26 of 30 Scale conditions and 25 of 30 Broad conditions. Median cosine is 0.541 and 0.704, respectively. Thus, the covariance measurement anticipates a recurring component of off-target change across both model groups.

Table 1 extends this view to four tasks. Across the ten model entries, Full/Length median alignment is 0.494/0.617 for code generation and 0.767/0.868 for summarization. Figure 2 shows the condition distributions in the matched-bank subset. Positive cosine supports same-reward directional alignment; the next section tests sharing across rewards.

3.2 A Length-Associated Mode Can Point Toward Shorter Answers

Length provides a compact probe of this coupling. Its median alignment is 0.520 in Scale and 0.902 in Broad; BoN-3 gives 0.521 and 0.631. The full covariance vector and these simpler measurements therefore recover overlapping predictive structure. A length-indexed vector describes a bundle of features. The reward sets its sign, orienting the prediction toward either shorter or longer answers.

The paired Full-minus-Length medians are -0.057 $[-0.170, 0.203]$ in Scale and -0.029 $[-0.212, 0.016]$ in Broad. These intervals include zero, placing Full and Length within the uncertainty of the paired endpoint comparison. The cross-reward experiment below addresses a different question: whether a subspace measured under other objectives contains the eventual change.

The correctness runs for OLMo-7B and Mistral-Nemo-12B illustrate the shorter-answer orientation. On both reasoning tasks, the frozen correctness score increases while mean length decreases; three of the four length intervals exclude zero. These effects connect directional alignment to measured output changes (Appendix D).

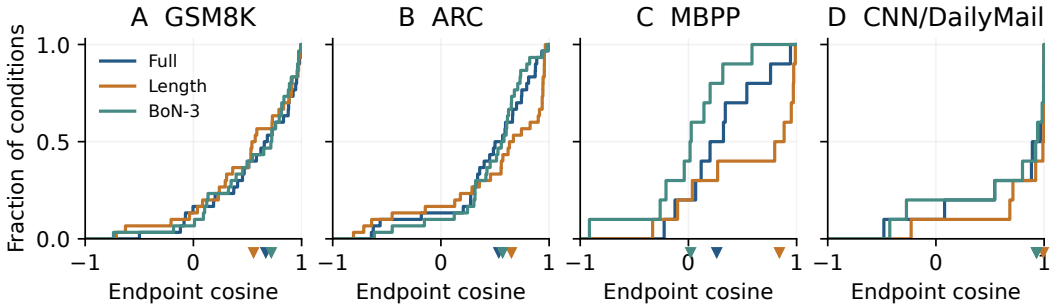

Figure 2: **Pre-RL predictors align with off-target change across tasks.** Empirical distributions of condition-mean cosine in the paired subset: 30 conditions per reasoning task and ten per native task. Curves further right indicate stronger alignment; triangles mark medians. Full condition estimates and paired intervals appear in Appendix J.

4 Shared Response Directions Transfer Across Rewards

4.1 Testing transfer on held-out rewards and prompts

For each reasoning condition, we omit the training reward from nine pre-RL reward probes and retain the leading left singular vectors U_k of the remaining spectrum. Discovery prompts determine the response directions, scaling, and subspace. Separate evaluation prompts determine training displacement. A common off-target mask retains seven coordinates before constant features are removed. All methods use discovery-set standard deviations to express these coordinates in the same units.

The primary outcome is captured displacement energy,

$$E_k = \frac{\|U_k^\top \Delta\phi\|_2^2}{\|\Delta\phi\|_2^2}, \qquad C_k = \cos(U_k U_k^\top \widehat{J}_r, \Delta\phi). \tag{6}$$

E_k tests whether the subspace contains the observed movement. The complementary projected cosine C_k tests its orientation. Behavior-only PCA selects directions using only the same within-prompt feature covariance. Comparing equal ranks isolates the contribution of reward conditioning.

E_k is sign-invariant and measures where change occurs. For C_k, the held-out reward's pre-RL vector supplies orientation within the other-reward subspace. Our headline result uses E_2. Omitting a probe does not remove correlations among related rewards, so transfer concerns this inventory; family-level holdouts would test unrelated objectives.

Table 2: **Shared transfer and its relation to length.** Median captured displacement energy; parentheses give condition counts. The trained reward is excluded from J_2. All methods share discovery scaling and evaluation prompts. $L + R$ adds a random orthogonal direction to Length and reports its exact expected capture. $L + P$ adds the leading behavior principal component orthogonal to Length. All columns except L_1 have rank two.

Cohort	L_1	J_2	PCA_2	$L + R$	$L + P$
Scale (30)	0.453	0.507	0.248	0.544	0.611
Broad (30)	0.639	0.732	0.362	0.699	0.738

4.2 Reward conditioning identifies transferable structure

Two response directions capture 50.7% of displacement energy in Scale and 73.2% in Broad, compared with 24.8% and 36.2% for two behavior-only principal components (Table 2). The median

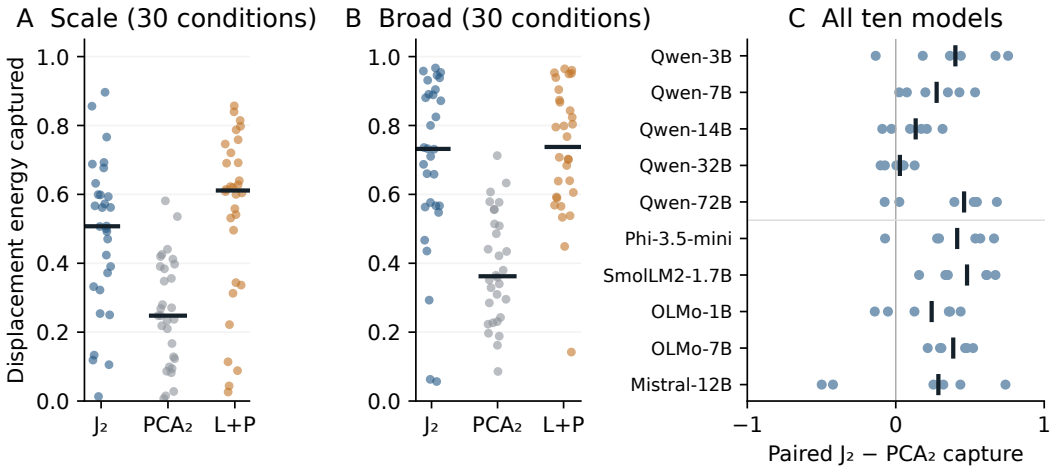

Figure 3: **Shared transfer spans both model groups. A–B:** every condition's rank-two capture, with medians marked by black bars. $L + P$ combines Length with residual PCA. **C:** paired J_2–PCA_2 differences for all 60 reasoning conditions. Each black tick marks a model's median; the divider separates Scale from Broad. Points average three seeds and reciprocal prompt folds.

paired gains are 0.177 $[0.052, 0.395]$ and 0.343 $[0.306, 0.409]$, using whole-model bootstrap intervals. The gain is positive for every model after aggregating its six reasoning conditions (Figure 3C). The result connects reward-conditioned variation before training to movement under an objective excluded from subspace construction.

Empirical nulls test alternative explanations for this capture. Isotropic random planes provide a dimension baseline; covariance-shaped random planes preserve a preference for high-variance directions. A third null permutes rewards jointly among responses to each prompt, preserving dependence among reward probes while breaking their pairing with behaviors. Both cohorts exceed all three nulls ($p = 1/513$ for each test). For example, reward-permutation 95th-percentile capture is 0.289 in Scale and 0.322 in Broad (Appendix E).

4.3 Length Indexes a Shared Multivariate Behavioral Mode

The matched length controls identify an interpretable shared mode: response properties that covary with length form a multivariate direction that different rewards can engage. If Length captures energy e in d dimensions, adding an isotropic orthogonal direction captures $e + (1 - e)/(d - 1)$ in expectation. Adding the leading principal component in Length's orthogonal complement gives an equally sized, data-dependent control. These controls reach 0.544/0.611 in Scale and 0.699/0.738 in Broad. Their strong capture locates a substantial part of the transferable structure in this length-associated behavioral bundle. Its orientation depends on the reward and can point toward shorter as well as longer answers, as illustrated in Section 3.

Length indexes an observed shared association; it does not establish that length causes other changes. We next test directional regularity and residual reward-conditioning gains under a different representation.

5 Behavioral Alignment Beyond the Original Features

5.1 A uniform embedding evaluation across all ten models

We ask whether directional regularity extends beyond surface features and how relative subspace gains change after length adjustment. We repeat the reasoning analysis with a fixed MiniLM encoder (Wang et al., 2020) and 30 contrastive behavior anchors. Each anchor contrasts two descriptions, such as checking evidence versus stating conclusions without checking. The projection of a response embedding onto that direction supplies a continuous coordinate. All ten models use the same encoder, templates, token budget, and scoring pipeline.

The primary expanded representation retains 26 coordinates, excluding four anchors closely related to the training rewards: procedural reasoning, symbolic reasoning, problem decomposition, and quantitative reasoning. The full 30-coordinate representation is a sensitivity analysis. Both use 256 prompts, twelve responses per bank, and reciprocal prompt folds. Subspace construction omits the training reward; endpoint cosine separately uses that reward's own pre-RL vector. Encoding uses a 192-wordpiece prefix. Discovery responses determine SD scaling and a frozen regression on log length for adjusted coordinates. Appendix G gives the complete protocol and per-model results.

Table 3: **Directional alignment in expanded behavior coordinates.** Median Full BRS cosine in the 26-axis primary representation; 30 conditions per cohort. Linear adjustment removes pre-RL within-prompt length dependence before prediction and evaluation. Both columns use disjoint prompts and discovery SD scaling. Joint resampling results appear in Appendix G.

Cohort	Raw coordinates	Length-adjusted coordinates
Scale	0.632	0.644
Broad	0.669	0.264

In the expanded coordinates, median alignment is 0.632/0.669 in Scale/Broad and remains positive after length adjustment (0.644/0.264; Table 3). These positive point estimates describe residual endpoint alignment in the embedding coordinates. Appendix G reports the bootstrap distributions obtained by refitting the full measurement pipeline whose calibration against a zero-cosine null remains unassessed. The subspace comparisons below then assess the relative contribution of reward conditioning against matched baselines in this representation.

5.2 Separating length-associated and residual variation

Excluding the length coordinate preserves behaviors that co-vary with length. A stronger diagnostic removes their fitted pre-RL dependence on log length. We fit linear or cubic-spline maps on discovery prompts, freeze them, and recompute response directions and displacement on held-out prompts. Both fits evaluate the same trained policies in length-adjusted coordinates. This procedure parallels regression controls for length bias in evaluation (Dubois et al., 2024), here applied to multivariate behavioral change.

On the five-model summarization subset, Full cosine falls from 0.887 with length proxies excluded to -0.179 after linear adjustment; all ten residual condition intervals include zero. By contrast, residual Full alignment in the OLMo-7B/Mistral-Nemo-12B reasoning subset is 0.282/0.303 under linear/spline adjustment. Appendix F gives matched distributions and subspace controls.

What remains beyond length? Positive residual endpoint cosine and superior held-out-reward subspace capture are distinct findings. Table 4 holds the PCA comparator fixed across representations.

Table 4: **Paired capture gains across representations.** Median $J_2 - \mathrm{PCA}_2$ differences, with 95% intervals; 30 conditions per cohort. Surface intervals resample model clusters; anchor intervals jointly resample prompts and seeds conditional on the measured models.

Representation	Length treatment	Scale	Broad
Surface	Raw	$0.177\ [0.052, 0.395]$	$0.343\ [0.306, 0.409]$
Surface	Linear adjustment	$-0.131\ [-0.242, -0.107]$	$0.002\ [-0.091, 0.108]$
26-anchor	Raw	$0.160\ [0.071, 0.175]$	$-0.001\ [-0.040, 0.021]$
26-anchor	Linear adjustment	$0.077\ [-0.011, 0.129]$	$-0.054\ [-0.109, 0.000]$

Surface spline adjustment gives the same qualitative pattern as linear adjustment. As a second control, raw anchor $J_2 - (L + P)$ gains are $0.177\ [0.085, 0.178]$ in Scale and $-0.024\ [-0.061, -0.010]$ in Broad. Thus, the residual advantage over PCA is representation- and cohort-dependent (Appendices F and G).

Positive residual endpoint alignment therefore coexists with representation- and cohort-dependent BRS–PCA gains. Surface features measure counts and markers; anchors measure similarity to fixed descriptions. Validating checks or reasoning steps additionally requires response-level labels.

6 Temporal Extension: Re-Measuring Response Geometry

Fresh answers from p_t define J_t for the next interval; frozen BRS uses J_0, and momentum extrapolates the previous displacement. Across 24 original GRPO trajectories, median cosines are 0.590, 0.362, and 0.430, respectively, with paired refresh gain 0.076 $[0.024, 0.195]$. Disjoint answer pools retain positive median gains. Extension settings yield mixed gains with all condition intervals spanning zero, leaving their benefit uncertain (Appendix I).

7 Related Work

Reward optimization and accompanying change. Preference-based reward learning (Christiano et al., 2017) and language-model RLHF (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022) optimize a low-dimensional signal over rich outputs. Reward overoptimization appears under RL and direct alignment (Gao et al., 2023; Rafailov et al., 2024); narrow-task training can also affect broader behavior (Betley et al., 2026; Denison et al., 2024). Behavioral-shift auditing asks whether two models differ (Richter et al., 2025). BRS asks how much of the direction of that difference can be anticipated from the starting policy. Deterministic benchmark rewards make the scoring rules explicit when studying this coupling.

Length as a training and evaluation factor. Length correlations in RLHF (Singhal et al., 2024) motivate training corrections (Shen et al., 2023; Bu et al., 2025). Length also affects model-based judgments (Zheng et al., 2023; Dubois et al., 2024), while GRPO normalization can produce length-related effects (Liu et al., 2025). Our contribution is a response-space interpretation: covariance with length defines a vector of accompanying feature changes. Reward-dependent orientation, shortening examples, and residualization distinguish this vector from a general preference for verbose answers.

Sampling and learning structure. Repeated sampling exposes low-probability solutions (Brown et al., 2024), and RLVR analyses compare the solution coverage of starting and trained models (Yue et al., 2025). Best-of-n and distributional alignment study how selection changes probabilities (Beirami et al., 2025; Yang et al., 2024). BRS instead measures the joint properties that change alongside the rewarded property, then tests their transfer across objectives. Its geometry lies in generated text, complementing directions in hidden representations (Park et al., 2024). Parameter-space analyses (Jacot et al., 2018; Tomihari, 2026) explain why a local distributional response motivates an empirical predictor of finite training.

8 Conclusion

Pre-RL response variation contains structure that reappears in the behavioral changes produced by RL. On two reasoning tasks across ten models, directions measured under other rewards locate substantial held-out-reward displacement and outperform equal-rank behavior-only PCA in the original surface coordinates. This transfer supports shared structure across the tested objectives beyond what behavior-only variance captures. A prominent part of this structure is a length-associated multivariate mode whose orientation varies with reward, including toward shorter responses. The broader directional regularity remains visible in an embedding-based representation after length adjustment, while the residual advantage of reward conditioning depends on representation and model cohort. BRS therefore provides a way to measure, before training, which response properties are likely to be selected together as an objective is optimized.

These conclusions concern the tested GRPO updates, reward inventory, and behavioral representations; the local covariance identity motivates the measurement, while actual training outcomes establish its empirical correspondence. Length indexes an observed association, and causal attribution would require a training intervention. Independent response-bank replication, probe-family holdouts, and externally validated behavioral labels would test how far the measured structure extends beyond this setting.

AI USE STATEMENT

AI tools assisted with manuscript organization and language revision, analysis-code development, and checking consistency between reported results and stored experiment records. No AI-generated response ratings are presented as human annotations. The underlying experimental records and analysis procedures are specified in the appendices.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3580–3602. PMLR, 2025.

Jan Betley, Niels Warncke, Anna Sztyber-Betley, Daniel Tan, Xuchan Bao, Martín Soto, Megha Srivastava, Nathan Labenz, and Owain Evans. Training large language models on narrow tasks can lead to broad misalignment. *Nature*, 649:584–589, 2026.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Yuyan Bu, Liangyu Huo, Yi Jing, and Qing Yang. Beyond excess and deficiency: Adaptive length bias mitigation in reward models for RLHF. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3091–3098. Association for Computational Linguistics, 2025.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.03741.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, Jared Kaplan, et al. Sycophancy to subterfuge: Investigating reward-tampering in large language models. *arXiv preprint arXiv:2406.10162*, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.04475.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on PPO and TRPO. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2005.12729.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 10835–10866, 2023.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, 2015.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1806.07572.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. In *Conference on Language Modeling*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2311.03658.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, W. Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. In *Advances in Neural Information Processing Systems*, 2024.

Leo Richter, Xuanli He, Pasquale Minervini, and Matt J. Kusner. An auditing test to detect behavioral shift in language models. In *International Conference on Learning Representations*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. Loose lips sink ships: Mitigating length bias in reinforcement learning from human feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2859–2873. Association for Computational Linguistics, 2023.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in RLHF. In *Conference on Language Modeling*, 2024.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2009.01325.

Akiyoshi Tomihari. Learning dynamics in RL post-training for language models. *arXiv preprint arXiv:2601.04670*, 2026.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment. In *IEEE International Symposium on Information Theory (ISIT)*, 2024.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. Datasets and Benchmarks Track.

GUIDE TO THE APPENDICES

The appendices follow the evidence chain from a measured response property to a claim about training. Appendix A defines what is observed; Appendices B–C explain the response calculation and evaluation protocol. Appendices D–G then examine alignment, competing explanations, and representation dependence. Training and temporal extensions follow in Appendices H–I; Appendix J provides the complete endpoint ledger.

Appendix	Question addressed	Page
A	What are the exact prompts, probes, parsers, and masks?	13
B	Why covariance, and where does the local argument stop?	17
C	How is BRS calculated, and how sensitive is it to budget?	19
D	Does positive alignment accompany actual training effects?	24
E	What alternatives do the matched controls rule out?	25
F	What changes after removing fitted length dependence?	26
G	Which findings survive a different representation?	27
H	How were the policies trained and settings varied?	32
I	Does re-measurement improve interval prediction?	34
J	Where are the individual successes and failures?	35

Reading the evidence. Three comparisons recur. Endpoint cosine asks whether a reward's pre-RL vector aligns with its later off-target change. Held-out-reward energy asks whether directions measured under other objectives contain that change. Residual comparisons repeat the measurement after subtracting a pre-RL length fit. Each comparison targets a different aspect of the measured change. Readers evaluating the central transfer claim can proceed directly to Appendices E–G; readers reproducing the analysis should begin with Appendices A–C.

Reproducing a measurement. Exact task strings and executable probe rules appear in Appendices A.4–A.5; Appendix C.4 traces twelve actual answers into a covariance vector. Appendix C.5 varies the prompt and response budgets by subsampling existing banks. Tables 13–15 provide the complete anchor wording. Together, these specifications connect the response-level measurements to the condition ledger.

A FEATURES, REWARDS, AND OFF-TARGET EVALUATION

This appendix fixes the meaning of "behavior" in the experiments. We distinguish the property scored on a response, the reward used for training, and the coordinates excluded when evaluating off-target change. A shared implementation can define both a reward and a feature; the off-target masks prevent direct target improvement from being counted as the main prediction result.

A.1 REASONING FEATURES AND REWARD PROBES

The reasoning bank uses the twelve features in Table 5. They are operational measurements: verification language, for example, records the occurrence of specified textual markers. The historical bank uses correctness, equation-density, and step-density rewards for training. Correctness and equation density equal their corresponding feature values. The step reward is $1 - \exp[-100c/(1.5L)]$, where c is the number of recognized step markers and L is token count; the behavior feature is the binary presence of such markers.

The nine spectrum probes are correctness, equation density, lexical diversity, step density, verification density, correction density, uncertainty density, code density, and final format. Density rewards use fixed, length-normalized counts with smooth saturation. The reward used for training is excluded in the leave-reward-out test. The common reasoning mask excludes accuracy, log token count, the short-answer indicator, step markers, and equation density, leaving seven coordinates before removal of any constant discovery-set features.

Table 5: Reasoning response features. All definitions are fixed deterministic functions of the generated answer and, for accuracy, the task label. The exact lexicons and regular expressions accompany the source.

Feature	Definition
Accuracy	Numeric answer match (tolerance 10^{-5}), option-label match, or normalized exact match.
Log token count	$\log(1 + L)$ for generated-token count L.
Code present	Indicator for a code fence or specified line-initial programming constructs.
Verification language	Indicator for a fixed lexicon, including "check," "verify," "confirm," and "recompute."
Correction language	Indicator for a fixed lexicon, including "actually," "wait," "mistake," and "reconsider."
Short answer	Indicator that $L \leq 24$.
Step markers	Indicator for a numbered, bulleted, or explicit step marker at a line start.
Final format	Indicator for a recognized FINAL or ANSWER span.
Answer repetition	Repeated final-answer occurrences beyond the first, capped at four and divided by four.
Uncertainty language	Indicator for a fixed lexicon, including "maybe," "perhaps," and "not sure."
Equation density	Specified mathematical operators per whitespace-delimited word before the last FINAL marker, capped at one.
Lexical diversity	Unique lowercase alphanumeric/underscore tokens divided by their total count.

A.2 Task-native measurements and their common masks

The new task banks use task-native coordinates. MBPP has eleven features: test-pass fraction, syntax validity, log length, comment density, lexical diversity, and six AST-based densities (functions, loops, conditionals, exception handling, assertions, and imports). AST counts are divided by the number of non-docstring statements and capped at one. Comment density counts comment-only lines relative to nonempty lines. CNN/DailyMail has seven features: ROUGE-L F1, log length, bullet density, sentence density, repeated-trigram fraction, lexical diversity, and an uncertainty indicator. ROUGE uses lowercase alphanumeric tokens without stemming; bullet density is the fraction of nonempty lines beginning with a bullet. The common spectrum mask takes the union of training-reward exclusions, leaving six MBPP and three summarization coordinates before constant-feature removal. For MBPP, these six retained coordinates are the six AST densities; the excluded union consists of test-pass fraction, syntax validity, comment density, log length, and lexical diversity. All candidate subspaces operate in these same coordinates and omit coordinates with zero discovery-set variance.

Table 6: Optimized quantities and key mask exclusions in the new task banks. The common spectrum mask takes the relevant exclusions across rewards. Each task defines its own quality score and reward scale.

Task	Training reward	Excluded target coordinates and direct proxies
Reasoning	Correctness	Correctness or answer-success coordinates, when present
Reasoning	Equation / step density	Log length, short-answer indicator, step markers, equation density
MBPP	Quality	Test-pass fraction and syntax validity
MBPP	Structure	Comment density, log token count, lexical diversity
CNN/DailyMail	Quality	ROUGE quality coordinate
CNN/DailyMail	Structure	Bullet density, log token count, sentence density

A.3 Interpreting marker features and off-target masks

Verification and correction scores register specified words. The step feature records marker presence, while the step reward varies smoothly with marker count and length. These definitions make the response properties easy to audit, including cases in which a phrase appears in an incorrect explanation. Appendix G complements these literal measurements with continuous embedding coordinates.

The off-target masks retain indirect associations with reward, allowing the experiment to measure accompanying changes. Residualization asks a further question: how much alignment remains after subtracting the pre-RL relation between each feature and length? Both analyses describe the measured outputs of the existing trained policies.

A.4 Exact task prompts and model-specific rendering

The strings below specify the user-message content, before applying a model's chat template. Bracketed fields are inserted from the dataset; empty lines are intentional. GSM8K uses the stripped question. ARC appends the choices in dataset order, mapping numeric labels 1–4 to A–D. MBPP supplies the request and required function signatures, without exposing the held-out tests. CNN/DailyMail inserts the article, without exposing the reference summary.

A GSM8K

```
[question]

Reason as needed. End with `FINAL: <numeric answer>`.
```

B ARC-Challenge

```
[question]
A. [choice A]
B. [choice B]
C. [choice C]
D. [choice D]

Reason as needed. End with `FINAL:
A/B/C/D`.
```

C MBPP

```
Write Python code for the following
request. Return code only, without
tests.

[request]

Required function signatures:
[signature line(s)]
```

D CNN/DailyMail

```
Summarize the following article faithfully and concisely.

[article]
```

Figure 4: Exact task-level instruction strings. The source packet preserves example records and the reasoning-data preparation function. Literal backticks in the reasoning instructions are part of the prompt; bracketed fields are explanatory placeholders.

Message roles and assistant prefix. The caller supplies one user message and no explicit system message. With chat rendering enabled and a tokenizer template present, it calls the tokenizer's chat-template renderer with generation-prompt insertion enabled. The template supplies role delimiters and the assistant prefix; a tokenizer without a chat template receives the plain prompt. A template can also inject its own default system message. For example, the stored Qwen2.5-7B native-task inputs contain "You are Qwen, created by Alibaba Cloud. You are a helpful assistant." before the user message, followed by the assistant delimiter. Effective system text is therefore determined by the model-specific template. The saved rendered input and input token IDs in the independent banks are the authoritative model-specific records.

Training, measurement, and termination. Training consumes prepared training-split prompts; pre- and post-training measurement uses the same evaluation prompt IDs and user content. The reasoning preparation function uses the same instruction suffix for both splits. Native-task collectors consume the prepared task prompts shown above, and training consumes its prepared training JSONL through the same single-user-message interface. Questions and articles differ across splits; neither reference labels nor native-task reference outputs are appended to the user message. The assistant prefix follows the model-specific template.

Generation ends at an effective EOS token or the 384-token cap, not at the literal FINAL string. The legacy collector uses the tokenizer EOS ID, discards that EOS and later tokens, and otherwise trims terminal padding. The independent collector uses the model generation configuration's EOS-ID set and records the stopping reason and retained continuation IDs; for the illustrated Qwen2.5-7B bank the set is {151645,151643}. Both decode with special tokens skipped. The retained continuation-token count supplies L; prompt tokens and terminal EOS/padding are excluded. Special tokens removed during decoding can make L differ from the token count obtained by re-tokenizing the visible text. Independent collection explicitly sets top-$k = 0$, while the legacy collector leaves that setting to the model generation configuration. Temperature and top-p are 0.9 and 0.95 in both. The source snapshot preserves these collector-specific settings for replication.

A.5 Executable definitions of the nine reward probes

Let y be the decoded continuation, L its stored token count, and

$$D(c, L; s) = 1 - \exp\left[-\frac{100c}{s \max(L, 1)}\right]. \tag{7}$$

The implementation also floors s at 10^{-8}; every specified scale below is strictly positive. Counts are taken on the original decoded text, except for the lowercasing and final-span treatment specified below. In particular, density rewards are not binary versions of the corresponding features.

Table 7: All nine reasoning probes and their exact saturation constants. Pattern names and lexicons are defined immediately below.

Probe	Executable quantity
Correctness	Indicator that the parsed final answer matches the reference under its answer type.
Equation density	$\min(c_{\text{eq}}/\max(w, 1), 1)$: EQUATION matches divided by whitespace-word count in the prefix before the last literal, case-insensitive `final:'; otherwise use all of y.
Lexical diversity	Unique lowercase matches of [A-Za-z0-9_]+, divided by their count floored at one.
Step density	$D(\#\text{STEP}(y), L; 1.5)$.
Verification density	$D(c_{\text{verify}}, L; 1.0)$.
Correction density	$D(c_{\text{correction}}, L; 0.8)$.
Uncertainty density	$D(c_{\text{uncertainty}}, L; 0.8)$.
Code density	$D(\#\text{CODE}(y) + \lfloor \#\text{backticks}(y)/3 \rfloor, L; 1.0)$.
Final format	Indicator of at least one FINAL-pattern match anywhere in y.

The equation-density prefix rule looks for literal `final:' after lowercasing; it does not remove an ANSWER span or a FINAL marker with a space before the colon. This differs intentionally from the more permissive answer-format parser. A code fence can contribute through both CODE and the separate backtick count. Neither count is deduplicated.

Patterns. The following are Python regular expressions. FINAL, MCQ, STEP, and CODE use case-insensitive matching; NUM and EQUATION use no additional flags. The STEP and CODE patterns explicitly recognize line starts through the alternative newline rather than a multiline flag. Match counts use non-overlapping matches.

FINAL: `(?:FINAL|ANSWER)\s*:\s*(.+?)(?:\n|$)`

MCQ: `\b([A-D])\b`

NUM: `[-+]?(?:\d+(?:,\d{3})*|\d*\.\d+)(?:[eE][-+]?\d+)?(?:/\d+)?`

STEP: `(?:^|\n)\s*(?:step\s*\d+|\d+[\.\)]|[-*])\s+`

CODE: ```` ```|(?:^|\n)\s*(?:def |import |from \w+ import|for \w+ in |while |print\() ````

EQUATION: `(?:=|\\frac|\\sqrt|\^|\+|-|\*|/)`

Complete marker lexicons. Verification: check; verify; verification; double-check; sanity check; confirm; recompute; cross-check. Correction: actually; correction; wait; rather; mistake; let me revise; reconsider. Uncertainty: maybe; perhaps; likely; uncertain; not sure; i think; possibly; approximately. For each lexicon, $c = \sum_t \mathrm{count}(\mathrm{lower}(y), t)$ uses Python substring counts, without word boundaries. Overlap across different terms remains: "double-check" contributes to both "double-check" and "check." The corresponding binary feature is one if any term occurs. These rules define literal marker occurrence.

Final-answer parsing and matching. The parser takes the last FINAL-pattern match and strips its captured content. If there is none, it takes the last nonempty stripped line, or the empty string. Numeric matching removes dollar signs, takes the last NUM match, removes commas, converts the matched string to a float or a non-exponential fraction, and returns false on parse failure. It uses relative and absolute tolerances of 10^{-5}. The literal NUM alternation is part of this operational definition: for example, '12.5' yields two matches, '12' and '.5', so the last-match rule returns 0.5. Distinct numbers with the same fractional part can therefore compare equal. The stored results retain this limitation of the frozen scorer. MCQ matching takes the first standalone A–D match in the extracted span and the reference, case-insensitively. Exact matching lowercases, strips, and collapses whitespace. The final option label controls the MCQ score, including when it conflicts with a number in the explanation; Appendix C.4 exhibits this in stored responses.

Finite-group advantages and zero-variance groups. The canonical CPU analysis, including the budget study below, uses the population group standard deviation:

$$\begin{aligned} \bar{r}_i &= K^{-1}\sum_{k=1}^{K} r_{ik}, \qquad s_i^2 = K^{-1}\sum_{k=1}^{K}(r_{ik}-\bar{r}_i)^2, \\ a_{ik} &= \frac{r_{ik}-\bar{r}_i}{\max(s_i, 10^{-8})}, \quad \widehat{J}_r = \frac{1}{N}\sum_i \frac{1}{K}\sum_k (\phi_{ik}-\bar{\phi}_i)a_{ik}. \end{aligned} \tag{8}$$

The variance denominator is K. A constant-reward group has $a_{ik} = 0$ and contributes a zero vector; it remains in the average over N prompts. Normalization uses only the sampled pre-RL reward group. Equation 8 fixes the measurement convention, separately from implementation-version differences in the training optimizer's advantage normalization (Appendix H). The released pure scoring functions preserve the fallback to a whitespace count if no positive token count is supplied; the analyses here pass the stored generation-token counts.

B Response Geometry and Parameter Updates

This appendix distinguishes an exact local identity for distributional reweighting from the empirical agreement observed after parameter training. We first make reward normalization explicit, derive the covariance response, and then identify the coupling introduced by shared parameters. The final calculation checks the accuracy of the local approximation on finite response banks.

B.1 Raw reward and standardized advantage

For unstandardized reward, define $G_r = \mathbb{E}_x[\mathrm{Cov}_{p_0}(\phi, r \mid x)]$. Covariance is linear, so $G_{\sum_j w_j r_j} = \sum_j w_j G_{r_j}$. Using an additive stabilizer for the population notation, promptwise standardization instead gives

$$J_r = \mathbb{E}_x\left[\frac{\mathrm{Cov}_{p_0}(\phi, r \mid x)}{\sigma_r(x) + \epsilon}\right]. \tag{9}$$

The executable finite-bank convention uses a floored population SD, as specified in Equation 8; both conventions give zero for a constant group. A reward mixture has its own prompt-dependent denominator. Consequently, standardizing a reward mixture and combining separately standardized columns generally give different prompt weightings. The raw and standardized spectra answer different weighting questions; all primary transfer results use standardized advantages.

B.2 Local reweighting and the shared-factor interpretation

For Equation 1, the first-order optimality condition gives $\log(q_\eta/p_0) = \eta a - \log Z_x(\eta)$. To derive Equation 2, write the normalizer as $Z_x(\eta) = \mathbb{E}_{p_0}[e^{\eta a} \mid x]$. Differentiating $\mathbb{E}_{p_0}[\phi e^{\eta a} \mid x]/Z_x(\eta)$ at zero yields $\mathbb{E}[\phi a \mid x] - \mathbb{E}[\phi \mid x]\mathbb{E}[a \mid x]$. For the finite empirical banks, the normalizer is analytic near zero and Taylor expansion gives the stated $O(\eta^2)$ remainder. The corresponding population expansion requires local integrability of the differentiated moments.

In the factor model of Equation 4, the columns of A describe shared feature patterns and b_r describes a reward's loading on them. Conditional residual cross-covariances vanish, leaving $A\,\mathrm{Cov}(z \mid x)b_r$. Within a prompt, advantage normalization divides this vector by the positive reward scale. Averaging across prompts can change its orientation because those scales differ. A dominant pattern common to prompts therefore motivates a shared direction, while secondary loadings motivate the held-out reward and residual tests. The factor model organizes the observable comparisons among rewards, shared directions, and residuals.

B.3 What changes under parameter updates

For a parameterized policy, let $s_\theta(x, y) = \nabla_\theta \log p_\theta(y \mid x)$. A policy-gradient update has direction $g = \mathbb{E}[a s_\theta]$, with first-order behavioral response

$$\Delta\mathbb{E}[\phi] \simeq \eta\mathbb{E}[\phi s_\theta^\top]g = \eta\mathbb{E}_{(x,y),(x',y')}[\phi(y)K_\theta((x, y), (x', y'))a(x', y')], \tag{10}$$

where $K_\theta = s_\theta(x, y)^\top s_\theta(x', y')$. The score kernel couples samples through shared parameters. Restricted parameterizations and optimizer preconditioning alter this coupling further. A kernel acting approximately as a positive scalar on the relevant subspace would preserve direction, but this is not measured here. The observed agreement is an empirical property of the tested updates.

The difference between the two expressions is substantive. Equation 2 changes probabilities directly at a fixed prompt. A parameter update changes probabilities through a shared score kernel, and may affect responses not present in the measured bank. Agreement between the two directions is therefore an outcome to test. The factor model describes the organization of covariance; the score kernel determines how a parameter update acts on that structure.

B.4 Subspace evaluation in common coordinates

For subspace evaluation, each discovery fold determines diagonal pre-RL SD scaling, reward responses, and within-prompt PCA. Evaluation folds determine displacement. Constant discovery-set coordinates are excluded consistently across methods. E_k measures squared projected norm divided by total squared norm in these common coordinates. The numerator is sign-insensitive, whereas C_k preserves direction. Each fold/seed result is averaged within its condition before cohort medians are computed. Only supported ranks are reported: native two-reward spectra become rank at most one after the training reward is omitted.

B.5 Finite-tilt remainder and a numerical check

For a fixed prompt and fixed advantage function, write $\mu(\eta) = \mathbb{E}_{q_\eta}[\phi]$. Differentiating twice gives

$$\mu''(\eta) = \mathrm{Cov}_{q_\eta}\left(\phi, (a - \mathbb{E}_{q_\eta}[a])^2\right). \tag{11}$$

Taylor's integral remainder therefore satisfies, for $\eta \geq 0$,

$$\|\mu(\eta) - \mu(0) - \eta\mu'(0)\|_2 \leq \frac{\eta^2}{2} \sup_{0\leq t\leq\eta} \left\|\mathrm{Cov}_{q_t}\left(\phi, (a - \mathbb{E}_{q_t}[a])^2\right)\right\|_2. \tag{12}$$

This identity is exact for a finite response bank. It separates the linear response to empirical reweighting from its curvature. As a numerical check across 80 conditions, empirical exponential reweighting at $\eta = 0.01$ gives median cosine 0.999998 with the linear direction and median relative vector error 0.00457. At $\eta = 0.05$, the corresponding values are 0.999941 and 0.02302. These calculations check the local expansion; the observed RL endpoints supply the separate test of transfer to parameter training.

C Statistical Analysis and Reproducibility

The main summaries combine observations at several levels. This appendix specifies the roles of prompt folds, training seeds, model sizes, and representation variants in aggregation and uncertainty estimates.

C.1 What is fitted before seeing training displacement

For each reciprocal fold, the discovery prompts supply reward–behavior responses and all fitted coordinate transformations. Evaluation prompts supply pre/post displacement. For length adjustment, regression is fitted on discovery responses and frozen before either evaluation bank is transformed. The following recipe describes the held-out-reward subspace comparison; same-reward endpoint measurements retain the relevant reward vector instead.

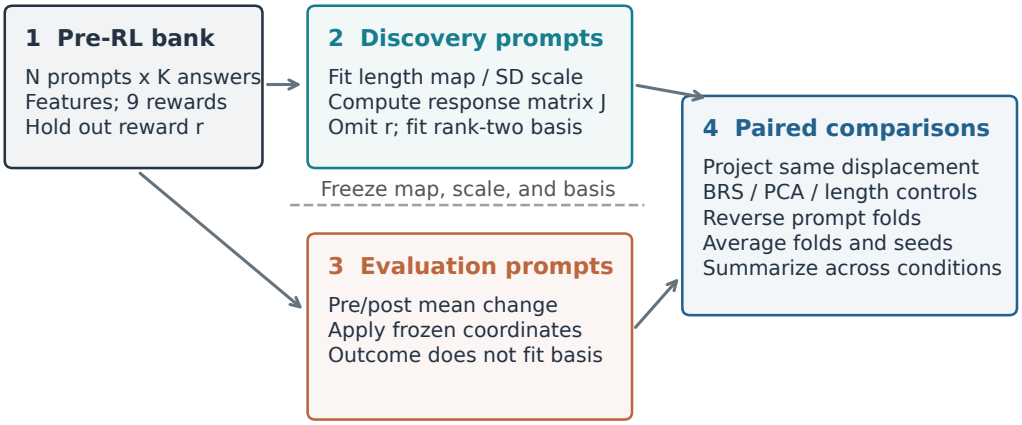

Figure 5: **Discovery and evaluation have separate roles.** The held-out-reward comparison freezes transformations and predictors before evaluating post-training displacement. Disjoint prompt folds separate fitting from evaluation; independent response pools further separate the generation streams assigned to those roles.

C.2 Aggregation and paired uncertainty

A condition fixes model, task, reward, and training configuration. We average its seed-level cosines or captures, and its reciprocal-fold values where applicable, before taking cohort medians. A cosine of pooled displacements is a different statistic. Paired comparisons subtract methods within a condition before aggregation. Dynamic summaries instead take medians across intervals and retain trajectory dependence in uncertainty estimates.

Scale contains Qwen2.5-3B/7B/14B/32B/72B. Broad contains Phi-3.5-mini, SmolLM2-1.7B, OLMo-1B, OLMo-7B, and Mistral-Nemo-12B. All cohort estimates, null distributions, and model-cluster intervals are recomputed from condition-level records using this partition. Both reasoning summaries use only GSM8K and ARC. Cluster intervals describe variation among the tested models, with the Qwen sizes belonging to one family.

Aggregate paired intervals use 12,000 resamples of five whole models per cohort. Each sampled model retains all its tasks and rewards. Condition-level intervals jointly resample prompts and three training seeds, preserving all predictor comparisons. These are pointwise 95% intervals, without multiplicity adjustment. Positive point estimates, positive lower confidence bounds, and positive paired differences remain distinct quantities.

Computing paired gains before aggregation. The reported paired gain is $\mathrm{median}_c(A_c - B_c)$, whereas the displayed method summaries are $\mathrm{median}_c\, A_c$ and $\mathrm{median}_c\, B_c$. The median's nonlin-

earity makes the order of subtraction and aggregation consequential. Similarly, averaging per-seed cosines and taking the cosine of pooled feature differences produce different statistics. Each caption specifies the estimator; folds and seeds are averaged within conditions.

C.3 Analysis populations and provenance

Table 1 spans 100 GRPO conditions. Matched condition-level analyses use all 60 reasoning conditions and 20 native-task conditions with complete raw-bank inputs. Seven algorithm or setting comparisons are separate. Run identifiers deduplicate archive exports and link checkpoints to their training trajectories. The ten-model semantic extension uses the same 60 reasoning conditions and all twelve responses for each of 256 prompts. It evaluates the existing trained policies in a second representation.

The 100-condition table is an endpoint inventory, while the 80-condition matched ledger is the population with the inputs required for the paired raw-bank checks. The expanded representation reuses the 60 reasoning conditions; its eight sensitivity variants apply different measurements to those same responses. The original 24 trajectories and the independent-stream extensions are separate temporal datasets. Their comparisons remain separate in Appendix I.

Reproduction order. The publication inputs distinguish endpoint condition records, fold/seed subspace captures, residual-coordinate summaries, and embedding-coordinate summaries. Reproduction should retain this order: verify model/task/reward identifiers and masks; reconstruct condition estimates; aggregate with the stated cohort partition; then render the corresponding tables and figures. Representation and bootstrap unit are part of the analysis definition and are recorded with each output.

Historical surface measurements use consistent definitions, including word-boundary handling for answer repetition. MBPP retains the experiment's Python 3.11 parsing convention; reanalysis reads stored test outcomes without executing generated programs. Response archives, numerical inputs, and generated result files retain provenance hashes. Cohort regrouping and expanded representation analyses are retrospective. The publication tables preserve condition-level distributions and specify their aggregation population.

C.4 A stored-response example: from answers to a response vector

This example makes the estimator auditable at the answer level. It uses the twelve stored pre-RL responses of Qwen2.5-3B to ARC item arc-test-1113. For legibility, we selected the prompt with the smallest total generated-token count among that model's ARC prompts with two to ten correctly scored responses, breaking ties by task ID. This pre-RL selection rule produces a compact example for tracing the estimator.

> Each body cell of a goldfish contains 94 chromosomes. How many chromosomes are contained in a goldfish sex cell?
> A. 23 B. 47 C. 94 D. 188

The actual user prompt additionally contains the ARC suffix in Figure 4; the reference option is B. Five of the twelve answers receive correctness reward one. Hence $\bar{r} = 5/12$, $s = \sqrt{(5/12)(7/12)} = 0.4930$, and the standardized advantages are $+1.1832$ for correct labels and -0.8452 for incorrect labels.

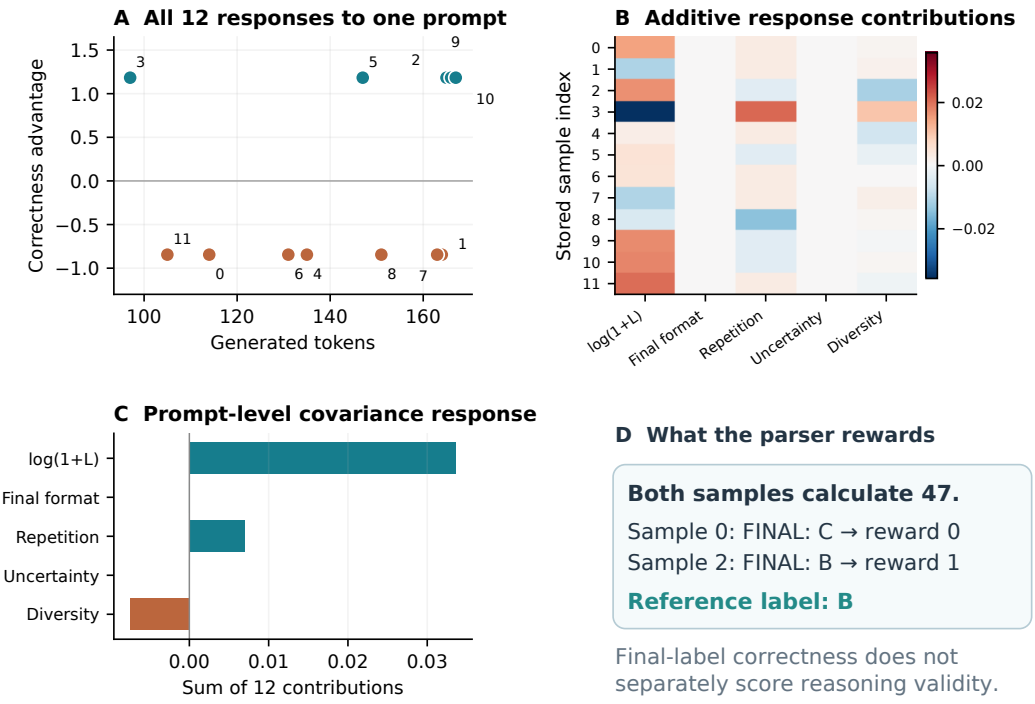

Figure 6: **A response vector is an average of signed, centered contributions.** A: all twelve answers, indexed as stored; color denotes whether the final label is correct. B: each answer's $(\phi_k - \bar{\phi})a_k/12$ in five illustrative, unscaled coordinates. C: sums of the columns in B. D: a concrete limitation of final-label correctness. Constant coordinates contribute zero. Log length is shown to explain the calculation and is removed from the common off-target evaluation mask. Bar lengths retain each coordinate's own measurement units.

Table 8: All twelve samples used in Figure 6. The final column is the additive contribution to lexical-diversity response, $(\phi_k - \bar{\phi})a_k/12$, with $\bar{\phi} = 0.5558$ (rounded). Computation uses unrounded values.

Index	Final label	Tokens	Reward	Advantage	Diversity	Contribution
0	C	114	0	-0.845	0.543	+0.0009
1	C	164	0	-0.845	0.534	+0.0015
2	B	165	1	+1.183	0.438	-0.0116
3	B	97	1	+1.183	0.656	+0.0099
4	C	135	0	-0.845	0.653	-0.0069
5	B	147	1	+1.183	0.529	-0.0026
6	C	131	0	-0.845	0.553	+0.0002
7	C	163	0	-0.845	0.526	+0.0021
8	C	151	0	-0.845	0.547	+0.0006
9	B	166	1	+1.183	0.549	-0.0007
10	B	167	1	+1.183	0.562	+0.0006
11	C	105	0	-0.845	0.577	-0.0015

Summing the twelve final-column contributions gives $J_{i,\text{diversity}} = -0.00747$ for this prompt. The log-length and repetition components are $+0.03360$ and $+0.00704$, while final format and uncertainty are constant and contribute zero. Code, verification, and correction are also constant here. The dataset response averages these per-prompt vectors across the bank, before the specified off-target masking and coordinate scaling. Endpoint prediction uses the vector averaged across prompts.

A failure visible only when reading the answer. Samples 0 and 2 both calculate 47 chromosomes, but sample 0 ends with option C and receives reward zero. Sample 2 ends with B and receives reward one. The following complete responses preserve the original notation and final labels. They expose

the specific information that determines the final-answer reward. The source packet includes all twelve unedited answers, all nine probe values, the twelve-feature vectors, and the full contribution matrix.

> **Sample 0: reward 0; 114 tokens.**
>
> To determine the number of chromosomes in a sex cell of a goldfish, we need to recall that sex cells (or gametes) in sexually reproducing organisms contain half the number of chromosomes found in body cells.
>
> Given that each body cell (somatic cell) of the goldfish contains 94 chromosomes, we can conclude that each sex cell will contain half of this number.
>
> Let's perform the calculation:
>
> \[\frac{94}{2} = 47 \]
>
> Therefore, a goldfish sex cell contains 47 chromosomes.
>
> FINAL: C

> **Sample 2: reward 1; 165 tokens.**
>
> To understand the number of chromosomes in a sex cell of a goldfish, let's first recall that in most organisms, including the goldfish, body cells (also known as somatic cells) contain a full set of chromosomes, and this number is known as the diploid number. For goldfish, each body cell has 94 chromosomes.
>
> Sex cells, also known as gametes (such as sperm or egg cells), contain half the number of chromosomes found in body cells. This half-set of chromosomes is referred to as the haploid number. Thus, for the goldfish, the diploid number (94 chromosomes per cell) is halved in the sex cells.
>
> Therefore, the number of chromosomes in a goldfish sex cell = 94 / 2 = 47 chromosomes.
>
> FINAL: B

C.5 How much measurement budget is needed to recover the bank geometry?

We resample existing pre-RL banks to separate sensitivity to the number of prompts from sensitivity to the number of responses per prompt. All twenty reasoning model–task blocks are included, with the three trained reward directions per block giving sixty conditions. We use $N \in \{32, 64, 128, 256\}$ and $K \in \{3, 6, 9, 12\}$, with 100 repeats per setting drawn from the stored pre-RL responses.

Each repeat draws a random prompt ordering and an independent random response ordering within every prompt, then takes nested prefixes for the sixteen budgets. Advantages are recomputed using the selected K responses under Equation 8. The seven-coordinate common mask and pooled pre-RL SD scaling are fixed from the full bank, with zero-variance coordinates removed once. This isolates covariance-estimation sensitivity in a shared metric, conditional on the full-bank scale. Sampling uses seed 2026092407 plus the recorded block index.

We compare each estimate with its own full 256×12 bank in two ways: the cosine of the response for the trained reward, and rank-two subspace overlap after omitting that reward from both matrices. For orthonormal bases U and $U_\star$, the latter is

$$S_2 = \tfrac{1}{2}\|U^\top U_\star\|_F^2 = \tfrac{1}{2}\sum_{j=1}^{2}\cos^2\theta_j. \tag{13}$$

It measures stability of the pre-RL plane under subsampling, independently of basis sign or rotation. The post-training statistic E_2 instead projects a realized displacement onto that plane. Zero-norm directions and unsupported rank-two estimates are recorded as undefined. Of 96,000 direction estimates, 29 are undefined (24 Scale, five Broad); means use the defined repeats. All 96,000 plane estimates have supported rank two. Condition-level counts accompany the released results.

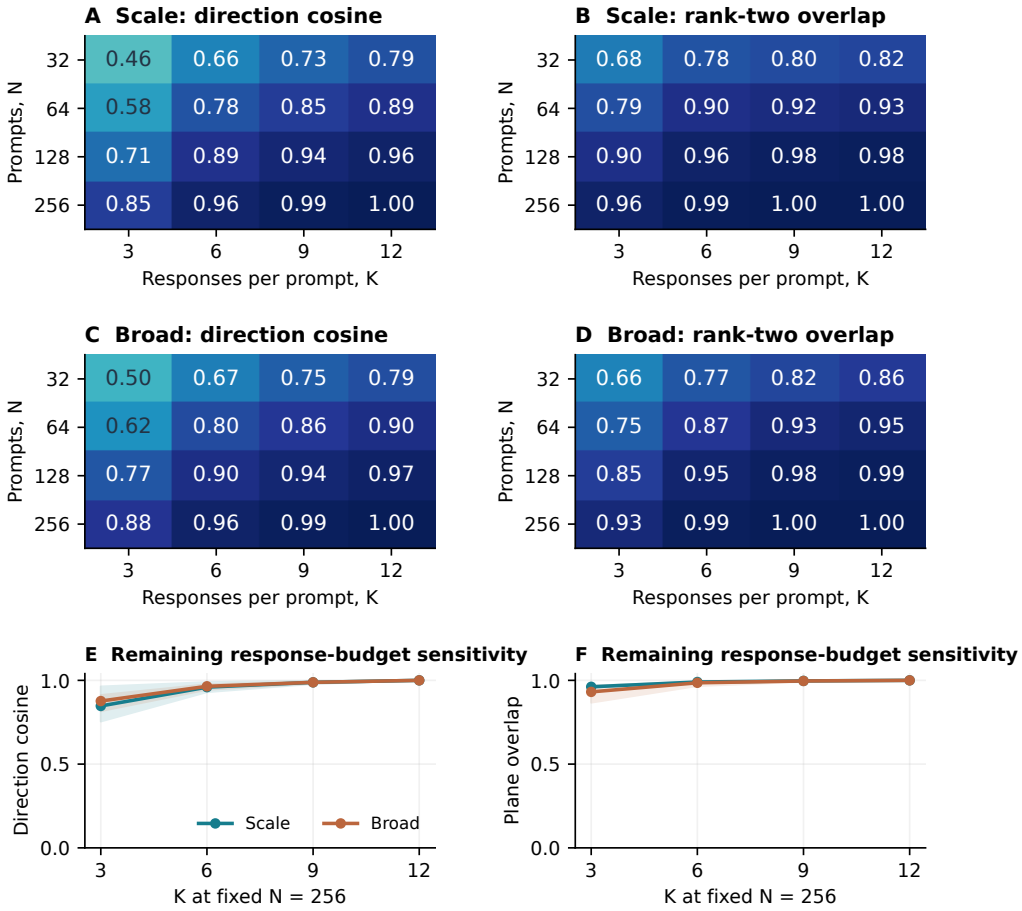

Figure 7: **Prompt and response budgets jointly determine within-bank stability.** A–D: mean over repeats within each condition, then median across thirty conditions per cohort. Direction cosine compares the same reward; plane overlap omits the trained reward. E–F: the $N = 256$ slices with bands showing the interquartile range across conditions. All panels use fixed full-bank coordinate scaling. The full-budget corner equals one by construction. The panels measure conditional subsampling stability against the shared full-bank reference.

Table 9: Selected measurement budgets from Figure 7. Cost counts sampled responses, NK. All values summarize within-bank agreement.

			Scale		Broad	
N	K	NK	Direction	Plane	Direction	Plane
32	3	96	0.461	0.676	0.504	0.658
64	6	384	0.779	0.898	0.802	0.868
128	6	768	0.889	0.964	0.904	0.953
128	12	1536	0.963	0.985	0.965	0.988
256	3	768	0.847	0.961	0.876	0.931
256	12	3072	1.000	1.000	1.000	1.000

At 128×6, median direction agreement is 0.889/0.904 and plane overlap is 0.964/0.953 for Scale/Broad. At 32×3, direction agreement drops to 0.461/0.504. At the equal-response budget of 768, 128×6 gives higher direction stability than 256×3, showing the contribution of repeated answers within each prompt. Plane overlap is numerically more stable than individual reward direction under this diagnostic; each metric is interpreted on its own geometric scale.

The subsets share responses with the reference, so these values are optimistic relative to agreement between independent banks. They quantify recovery of the present banks' geometry under fixed prompts and coordinate scales. Independent-stream results in Appendix I address a different source of variation.

D Observed Training Effects

D.1 Directional agreement and a simple comparator

Cosine measures directional agreement at any displacement magnitude. We therefore read endpoint alignment alongside changes in optimized reward and response length. The comparison with Length also asks whether the full reward-conditioned vector adds directional information beyond a compact shared proxy. Full/Length/BoN-3 reasoning median cosine is 0.541/0.520/0.521 in Scale and 0.704/0.902/0.631 in Broad. The paired Full-minus-Length medians are -0.057 $[-0.170, 0.203]$ and -0.029 $[-0.212, 0.016]$. As specified in Appendix C, the paired summaries take differences within conditions before aggregation.

D.2 Correctness scores can improve as answers become shorter

The historical 48-condition bank has positive optimized-reward change in every condition and positive lower confidence bounds in 38. Figure 8 gives frozen correctness-score and length changes in interpretable units for OLMo-7B and Mistral-Nemo-12B. Cosine describes direction independently of magnitude, so these effects complement the alignment results.

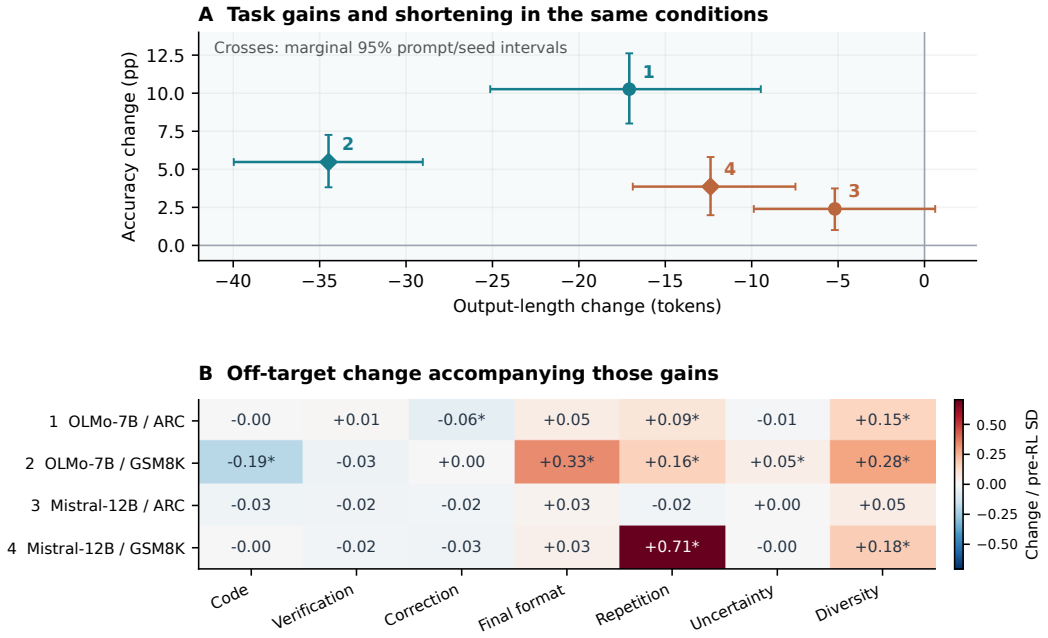

Figure 8: **Correctness-score gains can accompany shorter responses.** Accuracy denotes the frozen task score, subject to the numeric-parser limitation in Appendix A.5. A: score and length changes for the same four correctness conditions, with marginal 95% intervals obtained by joint prompt/seed resampling; crosses are not joint confidence regions. Numbered conditions match panel B. B: all seven common off-target feature changes in pre-RL SD units. Asterisks mark pointwise intervals excluding zero, without multiplicity adjustment. Three of four length intervals exclude zero.

The accompanying feature profiles differ across the same four conditions: the heatmap exposes the direction and magnitude of each measured off-target change instead of treating shorter answers as a uniform behavioral outcome.

In all four illustrated correctness conditions, the accuracy point estimate increases while the length point estimate decreases; three length intervals exclude zero. The sign assigned to the Length vector is therefore essential: it indexes accompanying properties in the direction favored by a reward, which can mean shorter answers. These runs illustrate the shorter-answer orientation alongside higher frozen correctness scores; the numeric-parser limitation in Appendix A.5 applies to GSM8K.

The 48-condition reward-change diagnostic and the four shortening examples have narrower coverage than the current 100-condition endpoint inventory. The stated reward-change counts apply to that historical subset; the four examples provide effect sizes for the two named models.

E Matched Subspace Controls

E.1 Which alternative explanation does each control test?

The controls form a sequence. Behavior-only PCA tests whether large pre-RL variance alone locates training movement. Haar planes calibrate capture from dimension alone. Covariance-shaped planes preserve a preference for high-variance directions without conditioning on rewards. Joint reward permutation preserves the reward inventory's internal dependence while destroying its pairing with responses. Finally, the length-containing controls ask how much of the predictive structure a simple shared index already explains. Passing the generic nulls and outperforming a length-containing control are different claims.

E.2 Constructing rank-matched length controls

For unit Length direction u, let $e = \|u^\top \Delta\phi\|^2 / \|\Delta\phi\|^2$. A random unit vector in its $(d-1)$-dimensional orthogonal complement captures $(1-e)/(d-1)$ in expectation. We compute the sum separately at each fold and seed, then aggregate. Length plus residual PCA uses the leading eigenvector of $(I-uu^\top)C(I-uu^\top)$, where C is discovery within-prompt covariance. The resulting basis is orthonormal.

E.3 Random and reward-permuted reference distributions

Each empirical reference uses 512 draws per discovery fold. Haar planes orthonormalize isotropic Gaussian matrices. Covariance-shaped planes premultiply Gaussian matrices by $C^{1/2}$. Reward permutation jointly shuffles all reward columns among answers to each prompt, preserving reward marginals and mutual dependence. The trained reward remains omitted. Subspaces are reused across training seeds, captures are averaged within conditions, and each null draw takes the median over the reassigned cohort. Upper-tail values are $(1 + \#\{T_b \geq T_{\text{obs}}\})/513$.

Table 10: **Reward–response pairing contains information beyond the null geometries.** Rank-two transfer against empirical references. Null columns give the 95th percentile of cohort-median capture. Both cohorts exceed all three references, with $p = 1/513$ per test.

Cohort	Observed J_2	Haar	Covariance-shaped	Reward permutation
Scale	0.507	0.337	0.334	0.289
Broad	0.732	0.347	0.358	0.322

Table 10 compares observed cohort-median capture with the 95th percentiles of the reference distributions. The observed 0.507/0.732 exceeds every displayed threshold, locating the measured capture above the reference ranges generated by dimension, covariance shape, and unpaired reward marginals. With 512 draws, $1/513$ is the smallest attainable Monte Carlo tail value. The result establishes reward-pairing information relative to these references; Table 2 separately shows how strongly length-containing subspaces capture the same movement.

The probes deliberately measure related properties. Holding out the trained reward tests transfer across these objectives while retaining their statistical dependence. Joint permutation tests the reward–response pairing while retaining this dependence. These conditional tests and the whole-model bootstrap address different sources of variation.

F LENGTH ADJUSTMENT AND TASK-NATIVE MEASUREMENTS

F.1 DELETING A COORDINATE IS DIFFERENT FROM REMOVING ITS ASSOCIATION

Removing log length from the evaluation vector prevents a direct length prediction from earning credit, but leaves correlated properties such as formatting or repetition. Residualization asks a stronger statistical question by subtracting their fitted dependence on pre-RL length. It can remove useful predictive variation as well as a confound; the resulting outcome is change in a different set of coordinates. Linear and cubic-spline maps regress each measured coordinate on pre-RL log token count using discovery responses. Splines have five knots and degree three. Both maps are frozen before evaluation on held-out pre/post responses. A non-length mask removes specified coordinates; regression instead subtracts their fitted length-associated component. Both transformations are fitted during analysis of the sampled outputs.

The regression uses within-prompt centered discovery features and length basis functions. If $h(\ell)$ is the linear or spline basis and $\widehat{B}$ its fitted coefficient matrix, the same map $\phi^{\mathrm{res}}(y) = \phi(y) - h(\ell(y))\widehat{B}$ is applied to held-out pre- and post-training responses. Freezing the map keeps pre- and post-training answers in the same discovery-defined coordinates.

F.2 NATIVE TASKS AND THE MATCHED REASONING SUBSET

Table 11: Matched task-native and reasoning-subset controls. Every entry is a median over conditions. Split and adjusted analyses separate discovery and evaluation prompts. The two-model reasoning subset contains OLMo-7B and Mistral-Nemo-12B; both models belong to Broad.

Subset	Split Full	Mask Full	Mask Length	Linear Full	Spline Full
Two-model reasoning	0.473	0.453	0.497	0.282	0.303
MBPP	0.223	0.262	0.528	-0.003	0.150
CNN/DailyMail	0.885	0.887	0.965	-0.179	-0.223

Read Table 11 from left to right as a sequence of diagnostics: Split Full separates prompts, Mask Full removes length and designated proxies, and Linear/Spline Full subtract fitted associations. Mask Length evaluates the same length-indexed predictor on the retained coordinates. The table compares transformations within each row, with the two-model reasoning row reporting a subset of Broad.

Summarization's ten residual cosine intervals all include zero. The contrast with the strong raw alignment places most of the measured signal in the length-associated component; the residual direction has substantial uncertainty. For MBPP, the corresponding linear/spline medians are $-0.003/0.150$; all ten Full endpoint intervals include zero. Native-task feature geometry combines sparse syntax or content measurements with continuous quantities, so the specified scaling is part of the evaluated representation.

F.3 RESIDUAL TRANSFER MUST BE JUDGED AGAINST RESIDUAL PCA

Table 12: Residual surface subspace transfer under the updated model partition. Columns report marginal median capture and the median paired difference with a whole-model 95% interval.

Cohort	Adjustment	J_2	PCA_2	Paired gain, 95% CI
Scale	Linear	0.196	0.317	$-0.131\ [-0.242, -0.107]$
Scale	Spline	0.189	0.324	$-0.090\ [-0.154, -0.072]$
Broad	Linear	0.303	0.351	$0.002\ [-0.091, 0.108]$
Broad	Spline	0.341	0.324	$0.003\ [-0.075, 0.109]$

Table 12 repeats the held-out-reward comparison after transforming both predictors and displacement. Scale's paired advantage reverses under both fits; Broad's interval includes zero under both. Removing fitted length dependence changes the comparison with PCA in both cohorts. The agreement of the linear and spline conclusions in this representation makes the result more informative than simply deleting the length coordinate.

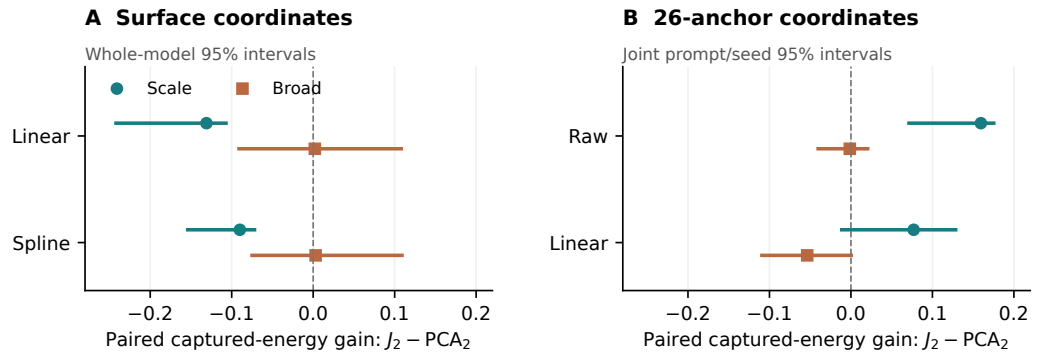

Figure 9: **Residual subspace gains vary across representations and cohorts.** Paired $J_2 - \mathrm{PCA}_2$ gains from the existing surface and primary anchor analyses. Zero marks equal capture. Panel A uses whole-model intervals; panel B uses joint prompt/seed intervals conditional on the measured models. The panels share a numerical axis for readability, but their uncertainty populations differ. Raw surface transfer appears in Table 2; the exact residual and anchor values appear in Tables 12 and 17.

Together, the matched length controls and residual comparisons locate much of the raw-space transfer in the shared length-associated component. The remaining relative gains depend on representation and cohort.

G Uniform Thirty-Anchor Representation

G.1 From an anchor description to a response coordinate

The expanded representation tests whether the findings depend on the original dictionary and formatting features. It replaces those coordinates while retaining the reasoning conditions and reward probes. The same trained policies are evaluated in a second coordinate system.

Every bank is encoded with the same locally stored all-MiniLM-L6-v2 encoder. CPU inference uses bfloat16 autocast, followed by float32 attention-mask mean pooling and normalization; a fixed pre-RL text sample checks numerical agreement with float32 inference. The first 192 wordpieces, including special tokens, define the encoded window. Relative to the 384-token generation cap, this window emphasizes the beginning of each response and can omit behavior expressed later.

For axis j, let $e^+_{j,t}$ and $e^-_{j,t}$ denote normalized embeddings of the positive and negative descriptions under template t. Writing $\mathrm{norm}(v) = v/\|v\|_2$, we construct

$$p^{\pm}_j = \mathrm{norm}\left(\frac{1}{3}\sum_{t=1}^{3} e^{\pm}_{j,t}\right), \qquad a_j = \mathrm{norm}(p^+_j - p^-_j), \qquad \phi_j(y) = \mathrm{norm}(e(y))^\top a_j. \tag{14}$$

Each axis contrasts two fixed descriptions, and its score measures the response embedding's projection onto that contrast. The wording is fixed before examining post-training outcomes.

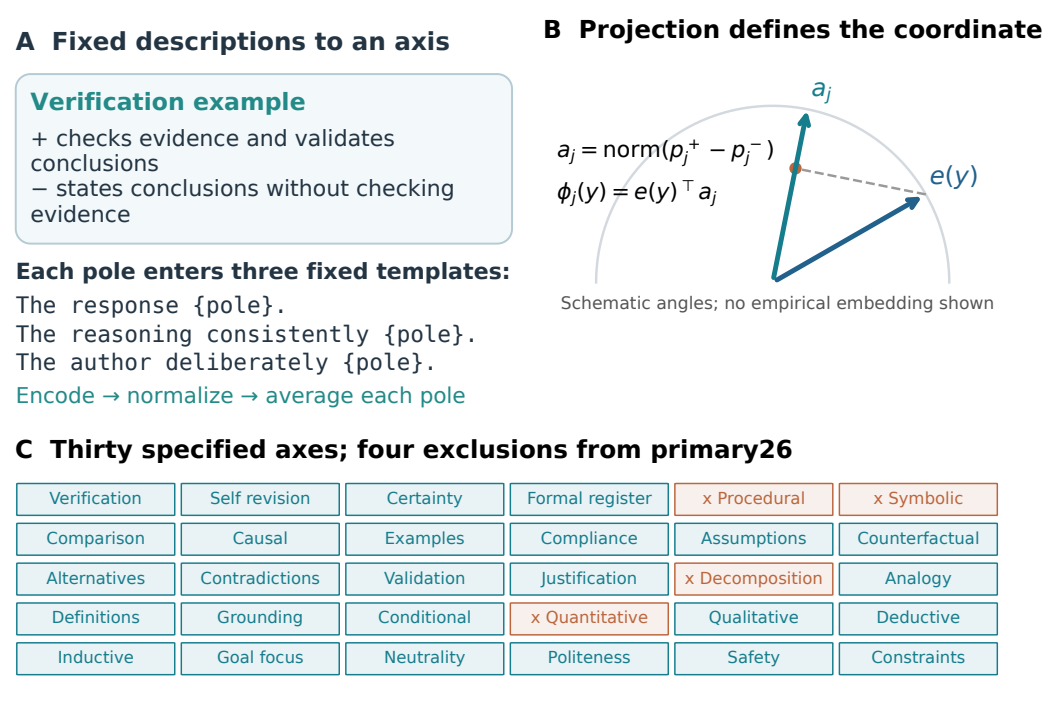

Figure 10: **From fixed language contrasts to a masked representation.** A: exact verification poles and all three templates. B: normalized pole difference and response projection, shown as a geometric schematic. C: all thirty axes in scoring order; orange crosses identify the four omitted from primary26. Full wording appears in Tables 13–15. Projection scores quantify embedding alignment with the fixed descriptions.

G.2 Complete anchor specification

Tables 13–15 give every positive and negative pole verbatim, in scoring order. Each pole is inserted into the three templates shown in Figure 10, producing six strings per axis. Daggers mark the four exclusions from primary26; the full30 sensitivity retains them. The specification is shared across all response banks.

Table 13: Exact anchor poles (1–10 of 30). A dagger marks an axis excluded from primary26. All other axes are retained.

Axis	Positive pole	Negative pole
verification	checks evidence and validates conclusions	states conclusions without checking evidence
self revision	notices mistakes and revises earlier reasoning	continues the initial reasoning without revision
certainty	expresses definite confidence in the conclusion	hedges and expresses uncertainty about the conclusion
formal register	uses technical impersonal academic language	uses casual personal conversational language
procedural reasoning†	organizes reasoning as an ordered procedure	presents a holistic explanation without ordered stages
symbolic reasoning†	uses equations symbols and formal relations	uses verbal concepts without mathematical symbolism
comparison	contrasts multiple alternatives explicitly	describes one claim without comparing alternatives
causal explanation	explains mechanisms causes and consequences	lists descriptive facts without causal links
example use	clarifies the idea with a concrete example	discusses only an abstract general principle
compliance	follows the requested task and supplies the requested help	refuses or declines the requested task

Table 14: Exact anchor poles (11–20 of 30). A dagger marks an axis excluded from primary26. All other axes are retained.

Axis	Positive pole	Negative pole
assumption awareness	states and examines assumptions behind the argument	leaves the assumptions implicit and unexamined
counterfactual reasoning	considers how the result changes under a counterfactual	considers only the situation that actually occurs
alternative hypotheses	proposes and evaluates competing hypotheses	commits to one hypothesis without considering alternatives
contradiction detection	identifies contradictions and resolves inconsistent claims	overlooks contradictions between inconsistent claims
intermediate validation	tests intermediate results before using them	uses intermediate results without testing them
answer justification	supports the final answer with explicit reasons	gives the final answer without supporting reasons
problem decomposition†	decomposes the problem into meaningful subproblems	treats the problem as one undivided whole
analogy use	explains the concept through a relevant analogy	explains the concept without analogy
definition use	defines key terms before applying them	uses key terms without defining them
evidence grounding	grounds claims in information provided by the prompt	introduces claims unsupported by the prompt

Table 15: Exact anchor poles (21–30 of 30). A dagger marks an axis excluded from primary26. All other axes are retained.

Axis	Positive pole	Negative pole
conditional reasoning	distinguishes cases using explicit if-then conditions	states one unconditional line of reasoning
quantitative reasoning†	reasons with numerical quantities and calculations	reasons without numerical quantities or calculations
qualitative reasoning	explains qualitative relations and conceptual properties	avoids qualitative relations and conceptual properties
deductive reasoning	derives a necessary conclusion from stated premises	offers a conclusion without deductive derivation
inductive reasoning	generalizes cautiously from observations or examples	avoids drawing general patterns from observations
goal focus	keeps every part of the reasoning focused on the requested goal	wanders into material unrelated to the requested goal
neutrality	uses emotionally neutral and balanced language	uses emotionally charged and partisan language
politeness	uses respectful and considerate wording	uses blunt and disrespectful wording
safety caution	recognizes potential harm and adds appropriate caution	ignores potential harm and gives unrestricted guidance
constraint awareness	explicitly follows the constraints stated in the instruction	ignores constraints stated in the instruction

G.3 Coverage and the primary comparison

All ten models use 256 prompts, twelve responses per prompt, and nine post-training banks per model–task block (three rewards and three seeds). The resulting analysis covers the same 60 reasoning conditions as the surface study. Reciprocal prompt folds separate discovery from evaluation; OLMo-7B and Mistral-Nemo-12B additionally have an independent pre-RL evaluation response pool. Reward advantages and the leave-training-reward-out construction are unchanged.

The primary representation retains 26 of the 30 axes. It removes procedural reasoning, symbolic reasoning, problem decomposition, and quantitative reasoning to reduce direct overlap with the training objectives. The remaining inventory comprises verification, self revision, certainty, formal register, comparison, causal explanation, example use, compliance, assumption awareness, counterfactual reasoning, alternative hypotheses, contradiction detection, intermediate validation, answer justification, analogy use, definition use, evidence grounding, conditional reasoning, qualitative reasoning, deductive reasoning, inductive reasoning, goal focus, neutrality, politeness, safety caution, and constraint awareness. The full 30-axis inventory is retained as a sensitivity analysis.

Table 16: All ten models in the primary 26-anchor analysis. Each row summarizes six reasoning conditions; the gain is the median of condition-level paired differences.

Model	J_2	PCA_2	$L + P$	Paired gain	Full cosine
Qwen2.5-3B	0.570	0.480	0.465	0.198	0.683
Qwen2.5-7B	0.593	0.415	0.455	0.095	0.482
Qwen2.5-14B	0.627	0.402	0.462	0.255	0.751
Qwen2.5-32B	0.530	0.411	0.360	0.139	0.643
Qwen2.5-72B	0.549	0.339	0.387	0.139	0.616
Phi-3.5-mini	0.369	0.386	0.506	-0.139	0.309
SmolLM2-1.7B	0.950	0.933	0.949	0.004	0.911
OLMo-1B	0.786	0.742	0.804	-0.002	0.678
OLMo-7B	0.843	0.706	0.801	0.124	0.777
Mistral-Nemo-12B	0.321	0.461	0.403	-0.086	0.404

Table 16 separates model-level variation from cohort summaries. Every Qwen2.5 size has a positive median paired gain over PCA, while the Broad models have mixed gains. For SmolLM2, the three methods have similarly high capture: J_2, PCA_2, and $L + P$ all capture about 0.95, leaving little separation among them. OLMo-7B has a positive paired gain, whereas Phi-3.5-mini and Mistral-Nemo-12B have negative gains. These patterns account for the strong alignment and mixed paired gains in the Broad cohort.

G.4 Paired gains and what their intervals describe

Joint bootstrap draws resample discovery and evaluation prompts together with training seeds, refitting regression, scaling, and subspaces in each of 1,000 draws. Prompt draws are shared across models and rewards for the same benchmark, whose ordered prompt IDs are checked to be identical. Seed draws are shared across rewards within a model–task block. These intervals describe prompt/seed uncertainty conditional on the measured models, unlike the whole-model cluster intervals used for the surface cohort comparisons.

Table 17: Primary representation uncertainty. Paired median differences in captured energy; joint prompt/seed 95% intervals. Raw and pre-only linear residual coordinates are evaluated separately.

Cohort	Representation	$J_2 - \text{PCA}_2$	$J_2 - (L + P)$
Scale	Raw	$0.160\ [0.071, 0.175]$	$0.177\ [0.085, 0.178]$
Scale	Linear	$0.077\ [-0.011, 0.129]$	–
Broad	Raw	$-0.001\ [-0.040, 0.021]$	$-0.024\ [-0.061, -0.010]$
Broad	Linear	$-0.054\ [-0.109, 0.000]$	–

The raw Scale gains in Table 17 are positive against both PCA and $L + P$. Broad has an approximately zero paired gain over PCA and a negative gain against $L + P$. After linear length adjustment, both cohorts' paired intervals reach or cross zero. Figure 9 places these comparisons beside the surface results. Residual endpoint alignment remains positive, while the relative value of reward conditioning depends on which response properties are measured.

Table 18: Endpoint resampling in the primary anchor representation. Point estimates are plug-in cohort medians; ranges are the 2.5th and 97.5th percentiles of 1,000 fully refitted joint prompt/seed bootstrap draws.

Cohort	Coordinates	Full cosine	Bootstrap percentile range
Scale	Raw	0.632	$[0.517, 0.646]$
Scale	Linear	0.644	$[0.489, 0.627]$
Broad	Raw	0.669	$[0.526, 0.652]$
Broad	Linear	0.264	$[0.139, 0.323]$

Table 18 concerns a different statistic: Full BRS endpoint cosine. All four plug-in medians are positive, including the two length-adjusted medians. For Scale linear and Broad raw, the plug-in value falls outside the fully refitted percentile range. Re-estimating the coordinate system and then taking nonlinear ratios and medians can shift the resampling distribution. We use these ranges to describe resampling stability. Bias, interval coverage, and calibration against a zero-cosine null remain unassessed.

G.5 Sensitivity to axes and coordinate geometry

Table 19 varies three choices: 26 versus 30 axes, raw versus linearly adjusted responses, and discovery SD scaling versus whitening. SD scaling equalizes marginal coordinate units; whitening additionally changes the weight assigned to correlated combinations, using a 0.1 identity shrinkage of the discovery within-prompt correlation. Since captured energy is measured in the resulting coordinates, a change in geometry changes the outcome being summarized.

Table 19: Complete cohort sensitivity analysis. Columns are median captures and the paired median $J_2 - \mathrm{PCA}_2$ difference. SD uses discovery feature standard deviations; whitening uses a 0.1 identity shrinkage of discovery within-prompt correlation.

Cohort	Axes/adjustment	Geometry	J_2	PCA_2	$L + P$	Paired gain
Broad	30/linear	SD	0.272	0.361	–	-0.042
Broad	30/linear	Whitened	0.125	0.099	–	0.018
Broad	30/raw	SD	0.721	0.657	0.744	0.003
Broad	30/raw	Whitened	0.503	0.293	0.435	0.067
Broad	26/linear	SD	0.252	0.348	–	-0.054
Broad	26/linear	Whitened	0.135	0.103	–	0.036
Broad	26/raw	SD	0.721	0.697	0.769	-0.001
Broad	26/raw	Whitened	0.524	0.364	0.460	0.058
Scale	30/linear	SD	0.491	0.398	–	0.043
Scale	30/linear	Whitened	0.330	0.142	–	0.142
Scale	30/raw	SD	0.588	0.401	0.435	0.129
Scale	30/raw	Whitened	0.377	0.140	0.163	0.166
Scale	26/linear	SD	0.451	0.364	–	0.077
Scale	26/linear	Whitened	0.344	0.141	–	0.145
Scale	26/raw	SD	0.586	0.432	0.438	0.160
Scale	26/raw	Whitened	0.421	0.183	0.199	0.178

Whitening produces positive paired gains in the displayed Broad variants, including the adjusted variants, whereas SD-scaled Broad gains are near zero or negative. The comparison makes the metric dependence of the paired gain explicit. Similarly, retaining four reward-adjacent axes changes the comparison's scope. The primary 26-axis, SD-scaled result remains the reference for the claims in Section 5.

For linearly adjusted anchor analyses, the released implementation explicitly disables the Length direction and $L + P$ control; dashes denote that omission. This convention is not an algebraic guarantee that the standardized Length response vanishes. The fitted within-prompt regression uses prompt weights w_i and a small ridge penalty, without the inverse reward-SD factors in the response estimator. With $\lambda = \log(1 + L)$, even an unpenalized fit satisfying

$$\sum_i w_i \, \mathrm{Cov}_i(\phi^{\mathrm{res}}, \lambda) = 0$$

need not satisfy $\sum_i w_i \, \mathrm{Cov}_i(\phi^{\mathrm{res}}, \lambda) / \max(s_{\lambda,i}, 10^{-8}) = 0$. A residual Length control would require checking this standardized vector directly. The reported Full, J_2, and PCA comparisons do not use the disabled direction.

Semantic interpretation. The anchors provide a second operational representation of the same responses, reducing dependence on a particular marker lexicon. Construct validation would require a response-level assessment, such as checking an intermediate calculation against a human-annotation rubric. Human annotation remains pending; the prepared packet contains blank labels. The reported semantic measurements are the fixed embedding projections defined above.

H Training Procedures and Setting Changes

H.1 Common training and measurement settings

The measurement group and training group serve different purposes. Training groups supply the policy's reward normalization; the twelve-answer measurement groups estimate within-prompt response covariance. Table 20 collects the settings used in the main study so these roles can be checked directly.

Table 20: **Main GRPO configuration and separate measurement budget.** Algorithm extensions below retain their explicitly stated differences.

Component	Setting
Optimizer steps; learning rate	200; 8×10^{-6}
LoRA rank; scale; precision	16; 32; bfloat16
KL coefficient; training group	0.02; eight responses
Measurement prompts; group	256; twelve responses per prompt
Measurement decoding	Temperature 0.9; top-p 0.95
Generated response cap	384 tokens
Training seeds per condition	Three

The main GRPO implementation uses group-normalized rewards and a global active-token reduction of the completion loss. LoRA applies to linear layers with rank 16 and scale 32 in bfloat16. Training groups contain eight responses; measurement groups contain twelve. Near-zero reward standard deviations are stabilized numerically by addition or flooring with a small epsilon across implementation versions. Constant-reward groups contribute zero under either convention. The measurement retains the finite-group normalization used to construct its reward advantages.

H.2 Algorithm extensions at a fixed model and task

PPO (Schulman et al., 2017) uses learning rate 10^{-5}, eight episodes per batch, one optimization epoch, policy/value clipping at 0.2, KL coefficient 0.02, discount 1, GAE parameter 0.95, and value-loss coefficient 0.1. DPO (Rafailov et al., 2023) uses the sigmoid preference loss, $\beta = 0.1$, learning rate 10^{-5}, and effective batch size eight. Its fixed preference corpus is constructed from eight candidates on each of 200 training prompts, excluding evaluation overlap. Both use rank-16 LoRA and 200 optimizer steps. Each procedure retains its own data collection, objective, and resulting policy KL. Implementation choices can also affect outcomes (Engstrom et al., 2020). The table reports the matched endpoint comparisons at this model–task–reward setting.

Table 21: Algorithm extensions: Qwen2.5-7B, GSM8K correctness, three seeds each. Condition-level endpoint cosines and paired difference intervals.

Procedure	Full	Length	BoN-3	Full − Length, 95% CI
PPO	0.225	0.641	0.232	-0.415 $[-0.554, -0.280]$
DPO	0.323	0.861	0.339	-0.538 $[-0.663, -0.427]$

Both algorithm extensions have positive Full endpoint cosine, but Length is more aligned in each, with negative paired difference intervals (Table 21). The two procedures thus retain measurable reward–behavior coupling, with Length providing the stronger endpoint predictor in each setting.

H.3 Adapter rank and training-group changes

Table 22: Paired change in Full endpoint alignment relative to a matched setting control; three training seeds. Matched rank and group-size changes.

Task	Change	Paired difference	95% interval
GSM8K	LoRA rank 32	0.071	$[-0.205, 0.317]$
GSM8K	Training group 16	0.092	$[-0.204, 0.401]$
CNN/DailyMail	LoRA rank 32	0.000	$[-0.005, 0.005]$
CNN/DailyMail	Training group 16	0.001	$[-0.006, 0.009]$

These comparisons vary adapter rank and training-group normalization within a matched task/model setup. Their intervals retain uncertainty across the three observed seeds. The main claim concerns the measured GRPO endpoints and cross-reward structure.

Every displayed setting-change interval includes zero. The estimated changes in Full alignment remain within the uncertainty of these matched perturbations. The summarization estimates are

close to zero, while the reasoning estimates have wider intervals. Appendix J retains the individual endpoint records to distinguish these between-setting differences from predictor comparisons within a setting.

I Checkpoint Re-Measurement and Independent Dynamic Tests

I.1 The prediction target is the next interval

Endpoint prediction asks about total change from the initial policy. Re-measurement instead asks whether fresh response covariance improves prediction of the next local displacement. This distinction matters because the initial spectrum may remain useful for an endpoint even when its orientation drifts during training. At each interval origin, refreshed BRS is recomputed from responses sampled at that checkpoint. The frozen baseline uses the initial response direction for the same interval. Momentum uses the previous realized displacement. Endpoint and interval experiments therefore have different outcomes even when they share an algorithm and model. Fresh origin-checkpoint responses distinguish the refreshed predictor from the frozen baseline.

I.2 Original trajectories and disjoint response roles

The original 24 trajectories have median refreshed/frozen/momentum cosines 0.590/0.362/0.430 and a paired refresh gain of 0.076 $[0.024, 0.195]$. Figure 11A shows that this aggregate coexists with substantial variation across trajectories and intervals. Panels B and C then apply progressively stronger separation of the sampled responses used for different roles.

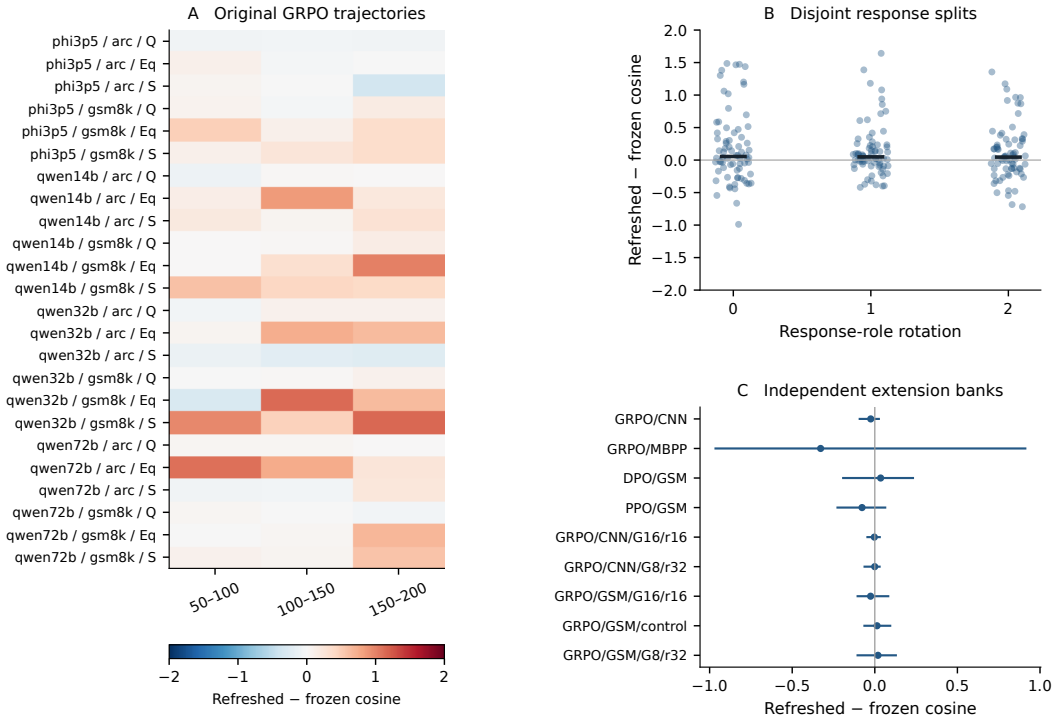

Figure 11: **Refresh gains differ between the original and independent-stream experiments.** A: all 24 original trajectories and three later intervals, colored by refreshed-minus-frozen cosine on a fixed $[-2, 2]$ scale. B: all interval differences under each disjoint response-index rotation; bars are medians. C: all nine complete independent-stream extension conditions, with paired 95% prompt/trajectory intervals. Uncertainty calculations retain the dependence among intervals from the same trajectory.

The original banks use twelve answers per prompt. Splitting them into A/B/C groups of four assigns disjoint response indices to the three sampling roles within the existing bank. A estimates J_t, B

measures both endpoints of the future displacement, and C measures momentum. Rotation-specific median gains and model–task-block intervals are 0.055 $[0.001, 0.102]$, 0.049 $[0.003, 0.128]$, and 0.046 $[-0.033, 0.166]$. Keeping rotations separate avoids inflating sample size through repeated use of the same answers.

The three rotations retain positive median gains, but one block interval crosses zero. Reassigning roles reduces direct reuse of the same responses. The next analysis tests the sampling roles using separately generated banks.

I.3 Refresh gains under independent generation streams

The newer banks assign separate generation streams to these roles. Each complete condition below contains three trajectories and nine adjacent 50-step intervals. Bootstrap draws preserve whole trajectories and common prompts across intervals. CNN denotes CNN/DailyMail, and the default setting is rank 16/group eight.

Table 23: All complete pooled-adjacent independent-stream comparisons. Medians over nine intervals; the final column is the median paired refresh-minus-frozen difference.

Condition	Frozen	Refreshed	Length	Momentum	Gain, 95% CI
GRPO/CNN	0.993	0.974	0.995	0.995	-0.025 $[-0.091, 0.024]$
GRPO/MBPP	0.009	-0.402	-0.412	-0.075	-0.328 $[-0.964, 0.911]$
DPO/GSM8K	-0.050	0.061	-0.277	-0.054	0.035 $[-0.191, 0.231]$
PPO/GSM8K	0.164	0.051	0.586	0.827	-0.077 $[-0.226, 0.063]$
GRPO/CNN/G16/r16	0.996	0.988	0.996	0.997	-0.003 $[-0.045, 0.030]$
GRPO/CNN/G8/r32	0.985	0.969	0.988	0.989	-0.001 $[-0.062, 0.029]$
GRPO/GSM8K/G16/r16	0.257	0.232	0.506	-0.385	-0.026 $[-0.104, 0.081]$
GRPO/GSM8K/r16 ctrl	0.365	0.350	0.259	-0.291	0.013 $[-0.063, 0.093]$
GRPO/GSM8K/G8/r32	0.092	0.182	0.477	0.097	0.019 $[-0.104, 0.127]$

Table 23 contains both positive and negative median gains, and every displayed gain interval includes zero. The summarization runs align well under several predictors while showing little additional refresh gain. MBPP has a negative point gain with a wide interval. The independent-stream results place the benefit of refreshing within substantial setting-specific uncertainty.

Median paired refresh gains are -0.077 $[-0.226, 0.063]$ for PPO and $+0.035$ $[-0.191, 0.231]$ for DPO. PPO favors refreshed over frozen in zero of nine intervals, and DPO in six of nine; these counts retain their within-trajectory dependence. A separate 128-prompt GRPO bank on Qwen2.5-7B/GSM8K correctness measures steps 100–200. Its smaller prompt budget and later interval define a separate diagnostic.

J Condition-Level Endpoint Comparisons

J.1 How to read the condition ledger

The tables below retain all 80 matched GRPO conditions and seven algorithm/setting records. They are grouped by analysis population so that a reader can trace a cohort summary back to its individual conditions. Each point estimate averages three seeds; the final column is a paired, pointwise 95% interval for Full minus Length. A positive Full cosine indicates alignment, while a positive paired difference indicates an advantage over Length. The interval quantifies uncertainty in the paired difference.

Q denotes correctness on reasoning tasks and task-specific quality on native tasks; Eq denotes equation density, Step step density, and S native structure. Short display names Qwen and Mistral-12B denote the Qwen2.5 family and Mistral-Nemo-12B, respectively. The ledger evaluates same-reward endpoint cosine under task-specific masks; Section 4 reports the separate held-out-reward energy comparison.

A concrete reading example. For Qwen2.5-7B on GSM8K under equation-density training, Full cosine is 0.880 and Length cosine is -0.036, with a positive paired difference interval $[0.733, 1.049]$.

For the same model and task under correctness training, the two cosines are 0.469 and 0.531, and the difference interval includes zero. For this model and task, Full's relative value changes with the training reward, motivating the reward-resolved comparisons below.

J.2 SCALE: REWARD VARIATION WITHIN ONE MODEL FAMILY

Table 24: **Within-family endpoint comparisons.** All 30 Scale conditions, ordered by model, task, and reward. Values are seed-mean cosines; the final column is the pointwise paired 95% interval for Full minus Length.

Group/task	Model / task / reward	Full	Length	BoN-3	Difference CI
Scale	Qwen-3B/arc/Q	0.597	0.943	0.527	[-0.523, -0.175]
Scale	Qwen-3B/arc/Eq	0.810	0.670	0.679	[-0.238, 0.416]
Scale	Qwen-3B/arc/Step	0.748	0.356	0.731	[-0.224, 0.949]
Scale	Qwen-3B/gsm8k/Q	0.513	0.966	0.496	[-0.613, -0.312]
Scale	Qwen-3B/gsm8k/Eq	-0.120	0.085	-0.184	[-0.578, 0.124]
Scale	Qwen-3B/gsm8k/Step	0.201	0.532	0.135	[-0.803, -0.017]
Scale	Qwen-7B/arc/Q	0.275	0.825	0.246	[-0.725, -0.259]
Scale	Qwen-7B/arc/Eq	-0.625	-0.713	-0.152	[-0.112, 0.513]
Scale	Qwen-7B/arc/Step	-0.181	0.180	0.124	[-0.662, 0.142]
Scale	Qwen-7B/gsm8k/Q	0.469	0.531	0.456	[-0.402, 0.290]
Scale	Qwen-7B/gsm8k/Eq	0.880	-0.036	0.836	[0.733, 1.049]
Scale	Qwen-7B/gsm8k/Step	0.623	0.037	0.816	[0.025, 0.881]
Scale	Qwen-14B/arc/Q	0.400	0.942	0.315	[-0.665, -0.280]
Scale	Qwen-14B/arc/Eq	0.874	0.633	0.820	[-0.151, 0.378]
Scale	Qwen-14B/arc/Step	0.570	0.621	0.455	[-0.493, 0.191]
Scale	Qwen-14B/gsm8k/Q	0.728	0.885	0.725	[-0.377, 0.081]
Scale	Qwen-14B/gsm8k/Eq	0.921	0.239	0.904	[0.361, 0.896]
Scale	Qwen-14B/gsm8k/Step	0.187	0.363	0.104	[-0.778, 0.493]
Scale	Qwen-32B/arc/Q	0.339	0.556	0.305	[-0.534, 0.210]
Scale	Qwen-32B/arc/Eq	0.592	-0.450	0.625	[0.412, 1.363]
Scale	Qwen-32B/arc/Step	0.699	0.288	0.651	[-0.240, 0.907]
Scale	Qwen-32B/gsm8k/Q	0.408	0.509	0.395	[-0.456, 0.366]
Scale	Qwen-32B/gsm8k/Eq	0.683	0.231	0.724	[0.098, 0.689]
Scale	Qwen-32B/gsm8k/Step	-0.004	-0.207	0.123	[-0.505, 0.993]
Scale	Qwen-72B/arc/Q	0.326	0.936	0.315	[-0.733, -0.409]
Scale	Qwen-72B/arc/Eq	0.207	0.128	0.615	[-0.398, 0.532]
Scale	Qwen-72B/arc/Step	0.500	0.572	0.514	[-0.411, 0.187]
Scale	Qwen-72B/gsm8k/Q	0.655	0.790	0.651	[-0.387, 0.115]
Scale	Qwen-72B/gsm8k/Eq	0.946	0.307	0.959	[0.355, 1.252]
Scale	Qwen-72B/gsm8k/Step	0.876	0.586	0.793	[0.094, 0.456]

The Scale rows show why the cohort comparison must remain paired. Correctness rows consistently favor Length in their point estimates, whereas equation and step rewards have mixed comparisons, including several large Full advantages. The ordering across model sizes varies with task and reward. This endpoint pattern coexists with the positive per-model held-out-reward subspace gains in Section 4, because the two analyses evaluate different predictions.

J.3 BROAD: ALIGNMENT AND SUPERIORITY VARY ACROSS CONDITIONS

Table 25: **Endpoint heterogeneity across the Broad cohort.** All 30 conditions. The same estimator, reward abbreviations, and paired interval convention as Table 24 apply.

Group/task	Model / task / reward	Full	Length	BoN-3	Difference CI
Broad	Phi-3.5-mini/arc/Q	0.451	0.957	0.608	[-0.822, -0.163]
Broad	Phi-3.5-mini/arc/Eq	0.745	0.957	0.558	[-0.569, -0.043]
Broad	Phi-3.5-mini/arc/Step	0.274	0.558	0.320	[-0.723, 0.185]
Broad	Phi-3.5-mini/gsm8k/Q	0.454	0.730	0.324	[-0.690, 0.218]
Broad	Phi-3.5-mini/gsm8k/Eq	-0.499	0.727	-0.738	[-1.637, 0.501]
Broad	Phi-3.5-mini/gsm8k/Step	0.757	-0.628	0.760	[-0.325, 1.510]
Broad	SmolLM2-1.7B/arc/Q	0.932	0.965	0.941	[-0.285, 0.063]
Broad	SmolLM2-1.7B/arc/Eq	0.991	0.744	0.985	[0.185, 0.308]
Broad	SmolLM2-1.7B/arc/Step	0.664	0.900	0.583	[-0.535, -0.015]
Broad	SmolLM2-1.7B/gsm8k/Q	0.992	0.999	0.991	[-0.015, -0.003]
Broad	SmolLM2-1.7B/gsm8k/Eq	0.990	0.973	0.972	[-0.025, 0.049]
Broad	SmolLM2-1.7B/gsm8k/Step	0.963	0.987	0.887	[-0.099, 0.006]
Broad	OLMo-1B/arc/Q	-0.643	-0.640	-0.616	[-0.808, 0.721]
Broad	OLMo-1B/arc/Eq	0.485	0.936	0.710	[-1.154, -0.118]
Broad	OLMo-1B/arc/Step	0.872	0.956	0.653	[-0.259, -0.015]
Broad	OLMo-1B/gsm8k/Q	0.833	0.834	0.813	[-0.162, 0.097]
Broad	OLMo-1B/gsm8k/Eq	0.981	0.905	0.972	[-0.006, 0.118]
Broad	OLMo-1B/gsm8k/Step	0.970	0.913	0.962	[0.018, 0.092]
Broad	OLMo-7B/arc/Q	0.662	0.945	0.737	[-0.503, -0.090]
Broad	OLMo-7B/arc/Eq	0.886	0.956	0.794	[-0.421, 0.004]
Broad	OLMo-7B/arc/Step	0.793	0.911	0.604	[-0.476, 0.011]
Broad	OLMo-7B/gsm8k/Q	0.885	0.910	0.877	[-0.115, 0.065]
Broad	OLMo-7B/gsm8k/Eq	0.373	0.571	0.348	[-1.130, 0.829]
Broad	OLMo-7B/gsm8k/Step	0.954	0.972	0.716	[-0.153, 0.028]
Broad	Mistral-12B/arc/Q	0.358	0.458	0.416	[-0.742, 0.601]
Broad	Mistral-12B/arc/Eq	-0.562	-0.808	-0.450	[0.023, 0.699]
Broad	Mistral-12B/arc/Step	0.294	-0.145	0.426	[0.170, 0.709]
Broad	Mistral-12B/gsm8k/Q	0.583	0.541	0.545	[-0.233, 0.233]
Broad	Mistral-12B/gsm8k/Eq	-0.074	0.294	0.092	[-0.949, 0.507]
Broad	Mistral-12B/gsm8k/Step	-0.086	-0.712	0.010	[-0.851, 1.171]

Many Broad conditions align strongly under several predictors, especially for SmolLM2 and OLMo-7B. Other rows show weak or negative alignment. A positive Full-minus-Length difference can even occur when both predictors have negative cosine, as in Mistral-Nemo-12B on ARC under equation density. In that row, Full is less anti-aligned than Length. Reporting the individual cosines alongside their paired interval makes the absolute and relative performance visible.

J.4 NATIVE TASKS: QUALITY AND STRUCTURE ARE DIFFERENT OBJECTIVES

Table 26: **Matched native-task endpoints.** Twenty conditions across the five models with complete raw banks. Q and S denote task-specific quality and structure rewards. The interval concerns Full minus Length, not Full versus zero.

Group/task	Model / task / reward	Full	Length	BoN-3	Difference CI
MBPP	Qwen-7B/mbpp/S	0.324	0.951	0.143	[-0.982, -0.133]
MBPP	Qwen-7B/mbpp/Q	-0.124	0.270	-0.207	[-0.701, 0.552]
MBPP	Qwen-14B/mbpp/S	0.200	0.884	0.201	[-0.992, -0.213]
MBPP	Qwen-14B/mbpp/Q	0.118	-0.100	0.318	[-0.293, 0.799]
MBPP	Phi-3.5-mini/mbpp/S	0.067	0.036	0.015	[-1.115, 1.181]
MBPP	Phi-3.5-mini/mbpp/Q	0.759	0.973	-0.039	[-0.863, 0.920]
MBPP	OLMo-7B/mbpp/S	0.341	0.970	0.026	[-1.145, -0.205]
MBPP	OLMo-7B/mbpp/Q	-0.224	-0.331	-0.261	[-0.641, 0.660]
MBPP	Mistral-12B/mbpp/S	0.540	0.797	0.588	[-1.006, 0.046]
MBPP	Mistral-12B/mbpp/Q	0.943	0.988	-0.915	[-0.822, 0.793]
CNN/DailyMail	Qwen-7B/CNN/S	0.883	0.988	0.800	[-0.900, 0.385]
CNN/DailyMail	Qwen-7B/CNN/Q	0.998	0.999	0.998	[-0.046, 0.001]
CNN/DailyMail	Qwen-14B/CNN/S	0.080	-0.229	-0.273	[-1.136, 1.497]
CNN/DailyMail	Qwen-14B/CNN/Q	0.998	1.000	0.996	[-0.047, 0.002]
CNN/DailyMail	Phi-3.5-mini/CNN/S	0.543	0.712	0.543	[-1.290, 0.565]
CNN/DailyMail	Phi-3.5-mini/CNN/Q	0.968	0.997	0.922	[-0.092, 0.003]
CNN/DailyMail	OLMo-7B/CNN/S	-0.481	0.683	-0.427	[-1.525, 1.127]
CNN/DailyMail	OLMo-7B/CNN/Q	0.998	0.999	0.994	[-0.009, 0.001]
CNN/DailyMail	Mistral-12B/CNN/S	0.891	0.922	0.939	[-0.506, 0.056]
CNN/DailyMail	Mistral-12B/CNN/Q	0.986	0.998	0.982	[-0.054, 0.001]

Summarization quality rows have uniformly high cosine point estimates across the three predictors, while structure rows vary substantially. MBPP is more heterogeneous across models and objectives, with several intervals spanning a wide range of relative effects. The matched native-task analysis in Appendix F links this aggregate alignment to the length-associated component and examines its residual variation. This ledger covers the five models with matched raw banks; Table 1 provides the full ten-model task inventory.

J.5 EXTENSIONS: DISTINGUISH PREDICTOR AND SETTING COMPARISONS

Table 27: **Algorithm and setting endpoints.** Seven additional condition records. G denotes training-group size and r adapter rank. The r16 control is a matched reference, not another treatment. Intervals compare Full with Length within each row.

Group/task	Model / task / reward	Full	Length	BoN-3	Difference CI
PPO	Qwen-7B/gsm8k/Q	0.225	0.641	0.232	[-0.554, -0.280]
DPO	Qwen-7B/gsm8k/Q	0.323	0.861	0.339	[-0.663, -0.427]
Settings	Qwen-7B/CNN/Q / G16/r16	0.999	0.999	0.999	[-0.044, 0.002]
Settings	Qwen-7B/CNN/Q / G8/r32	0.998	0.999	0.999	[-0.047, 0.001]
Settings	Qwen-7B/gsm8k/Q / G16/r16	0.468	0.883	0.481	[-0.582, -0.234]
Settings	Qwen-7B/gsm8k/Q / r16 control	0.375	0.611	0.363	[-0.501, 0.105]
Settings	Qwen-7B/gsm8k/Q / G8/r32	0.446	0.888	0.451	[-0.600, -0.191]

Each interval in Table 27 compares predictors within a fixed procedure or setting. By contrast, Table 22 compares Full BRS between a modified setting and its matched reference. The two tables respectively quantify within-setting predictor differences and between-setting changes in Full alignment. The algorithm endpoint estimates and intervals use the same stored export as Table 21. Dynamic refresh comparisons in Appendix I concern future intervals and are likewise separate from these full-training endpoints.