Diagnosing Test-Time Recoverability through Failed Reasoning Traces

Anonymous ACL submission

Abstract

When post-trained language models fail on reasoning problems, the common solution is to spend compute on more attempts, and the failed traces play no further role. We argue this discards a crucial signal; some failures come from unlucky sampling, where more rollouts help, while others are structural and resist resampling regardless of budget. We propose that failed traces encode recoverability structure: the inference-time signature of which test-time interventions can rescue a given failure. Three problem-level trajectory features, derived from the structure of available interventions, recover this structure from failed rollouts alone. They cluster failures into stable regimes, characterize the failure topography of different post-training methods ($84.3 \pm 4.3\%$ classification accuracy), and support a training-free routing rule that lifts rescue by $+12.2$ pp on the deployment-relevant Steerable-Hard subset (failures where retry is insufficient and a bounded intervention is reachable). The features and the routing rule transfer across two cross-family probes. The same three features thus convert failed traces from discarded data into a diagnostic object, supporting test-time routing and post-training analysis without training-time or weight-space access.

1 Introduction

Reasoning models are increasingly evaluated and deployed under test-time compute budgets. When a model fails, the standard response is to sample again. Best-of-N (Stiennon et al., 2020; Cobbe et al., 2021), self-consistency (Wang et al., 2023), and retry-style evaluation (Chen et al., 2021) all spend additional compute by drawing more trajectories from the same post-trained model distribution. This is effective when the first failure was an unlucky sample. But it is wasteful when the model is locked into a stable wrong trajectory, and incomplete when the failure is locally recoverable only by changing the trajectory rather than resampling it. In deployment, the central question becomes *given that the model has already failed, what should the next unit of compute do?*

Aggregate accuracy cannot answer this question. It maps every failed trace to the same label, although failures may arise from sampling noise, a single demoted reasoning step, a localized misrouting event, or a trace-wide deformation of the model's reasoning dynamics. These cases require different responses: retry, local perturbation, logit-space steering, or no further spending under the available budget. We argue that the failed trace itself is the diagnostic object that distinguishes them.

We study failed traces through an **inference-time, post-hoc, trace-level diagnostic**. Retry and temperature-based resampling are rank-preserving: they can reweight the specialist's local distribution but cannot make a lower-ranked token become the local mode. Logit steering toward a lineage ancestor acts in natural-parameter space and can invert local ranks when the specialist and ancestor disagree. This operator-class distinction determines what a trace-level diagnostic should measure.

We focus on the subset where routing has deployment value. A problem-unit is *Steerable-Hard* when standard retry is insufficient but at least one bounded test-time intervention is reachable. Outside this subset, routing is either unnecessary, because retry already works, or futile, because none of the available operators rescues the problem.

From failed rollouts alone, we extract three problem-level features derived from the operator geometry: deformation spread, junction concentration, and local displacement budget. These features support both a routing example (§3) and an audit channel (§6). The routing example uses a training-free rule fixed by the operator class: diffuse deformation routes to dense steering, concentrated junctions to sparse steering, and high local headroom to local temperature lifting.

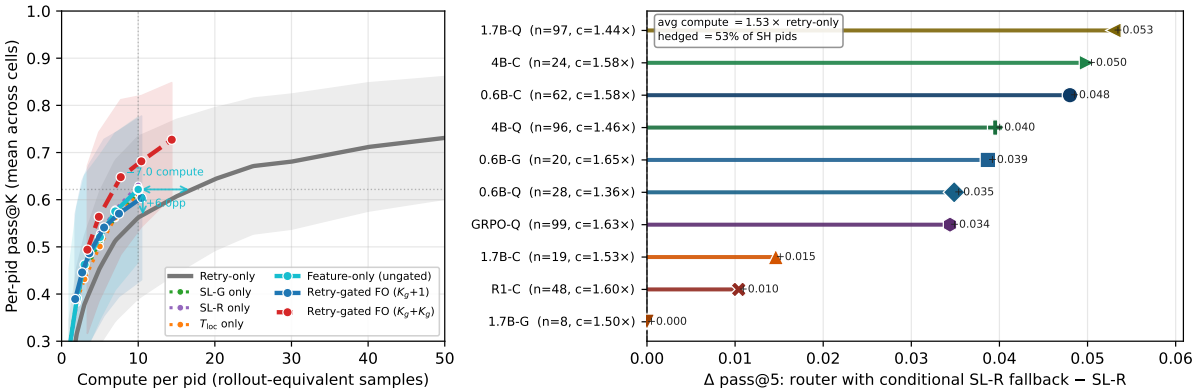

Figure 1: **Motivation: diagnosing recoverability and routing test-time compute.** Fig. 1 illustrates two ways the trajectory features expose the operator-class structure of failed traces: by locating where additional compute changes per-pid recoverability (Panel a), and by separating SH pids the router rescues from those the best fixed operator already covers (Panel b). The feature framework itself is what carries the broader claims (§5, §6). **(a)** Per-pid pass-rate vs. compute across 10 cells (Panel b includes one additional cell, 11 total). Retry-only (solid grey) is the baseline at K=50; single operators (SL-G, SL-R, T_{loc}), Feature-only routing (cyan), and its retry-gated variants (K_g+1, K_g+K_g) are anchored at K=10. The cyan crosshairs mark Feature-only at K=10: it reaches the same per-pid rescue as retry at substantially lower compute. **(b)** Per-cell lollipop of unbiased pass@5 lift on the Steerable-Hard subset (retry@32=0): horizontal bar extends rightward from 0 to the mean per-pid difference Δ = (router with conditional SL-R fallback) − SL-R. The x-axis starts at 0 because Δ is non-negative by construction (the fallback policy is a strict superset of SL-R), so longer bars indicate cells where the router contributes pids that SL-R alone cannot rescue. Marker shape and color identify each cell; n is the SH pid count and c the average compute (in operator-equivalents, 1.0× when the router is kept, 2.0× when SL-R fallback fires).

Contributions. We contribute (i) a *reframing* of failed reasoning traces as diagnostic objects, asking what the trace itself tells us about test-time recoverability: which test-time interventions can rescue which failures, and on what scope (*Steerable-Hard*: retry insufficient and a bounded intervention reachable); (ii) a *theoretical lens* that derives the three trajectory features from operator-class structure (rank-preserving vs natural-parameter, with a closed-form sufficient-not-necessary inversion bound; Props. 1, 2; §2.1); (iii) a *diagnostic audit channel*: the same features recover SFT-vs-RL post-training distinctions from failed rollouts alone, without training-time, weight-space, or paired-comparison access (§6); and (iv) a *routing example* demonstrating the features are deployment-actionable, via a training-free operator-class routing rule on *Steerable-Hard* (Eq. 2: retry insufficient *and* a bounded intervention reachable; §4); and (v) *cross-family transfer* of both the audit signatures and the routing rule between Qwen3, R1-Distill-Qwen-Math-1.5B, and Phi-4-mini-reasoning, giving evidence that the features encode operator-class structure rather than specialist-specific artifacts (§6, App. E).

2 Setup

Operators. Five test-time interventions span two mechanism classes (Proposition derivations in App. A). **Rank-preserving**: **retry** (re-sample at T=0.6)[1] and **local temperature** (T_{loc}=1.5 at a detected junction) re-weight the support but cannot invert local token ranks. **Natural-parameter**: **SL-G**, **SL-R**, and **DL** apply logit mixing $z_\alpha = \alpha z_S + (1-\alpha) z_A$ at α=0.7 (an e-geodesic toward the pre-training checkpoint z_A; App. G, all $\alpha \in [0.5, 0.8]$ beat retry; per-cell optima vary and are reported in App. H[2]), differing in *where*: SL-G at the detector-detected junction (App. A.1), SL-R at a uniformly random position, DL at every position. Geo↔SL-R isolates the value of position selection at matched steering strength.

[1] *retry@K* is functionally *best-of-K* sampling: take K independent rollouts and succeed if any pass. We use the term "retry" because the dispatch's failure set is conditioned on an already-failed rollout pool (the initial K=10 rollouts on which Fail@K=10 is defined); each additional "retry" is functionally one more best-of sample on top of that pool.

[2] The canonical α=0.7 choice is near-optimal on most cells but not all; e.g., Qwen3 SFT-1.7B × CruxEval peaks at α=0.6 with a +25 pp SH lift gap, per the family-wise grid in App. H.

2.1 Why these features: the operator class determines what is measurable

What a test-time operator can actually do locally. When a trace has already failed, the question is not whether to spend more compute but *what the available operators can change about it*. Retry resamples from the same specialist distribution: the most likely next token stays the most likely. Local temperature flattens that distribution but cannot move a different token to the top. Logit mixing with the lineage ancestor is qualitatively different: a token the specialist confidently prefers can be suppressed if the ancestor disagrees. Two operator classes, two reaches.

The geometric distinction. Probability mixing (the *m-geodesic*, $\tilde{p} = \beta p_S + (1-\beta)p_A$) is a weighted average; the argmax token survives whenever the specialist's top margin is large enough. Logit steering (the *e-geodesic*, $p_\alpha \propto p_S^\alpha p_A^{1-\alpha}$) is a product-of-experts: a token must have non-negligible probability under *both* models to stay top. Only the second class can invert local ranks. Retry and T_{loc} are rank-preserving; SL-G, SL-R, and DL are rank-inverting.

What the operator class forbids, and what it permits. Rank-preserving operators cannot rescue a Rank Misrouting failure (Prop. 1), while the e-geodesic admits a closed-form rank-inversion threshold $\alpha^\star$ with local displacement budget equal to the Fisher information of the specialist's local temperature submodel (Prop. 2; App. A), measured as $V_{t^\star}$ at the worst pivot the trace exposes. The $\alpha^\star$ condition of Prop. 2 is sufficient for single-position rank inversion at $\alpha < \alpha^\star$; it is not necessary for rescue. Empirically, only a minority of SH-rescued pids meet the condition at the engine-detected junction, and most engine-selected injection positions place the specialist and ancestor on the same argmax token (App. T). The operator class therefore predicts the right *class* of intervention (rank-preserving vs. natural-parameter), even when the per-pid rescue mechanism is more diffuse than the closed-form picture.

From the geometry to the features. Each of the three problem-level features measures one geometric quantity the operator class makes actionable: $\bar{J}_{\text{frac}+}$ (density of e-geodesic deformation across the trace $\rightarrow$ dense intervention DL); $\bar{C} = \bar{J}_{\max}/\bar{J}_{\text{mean}}$ (concentration $\rightarrow$ sparse intervention at the right position SL-G); $V_{t^\star}$ (local displacement budget at the worst pivot $\rightarrow T_{\text{loc}}$ when high, where bounded steering still permits rank inversion). The operator the geometry prescribes is the one whose dominant feature is highest. This is not an empirical fit; it is the rule Eq. 3 implements in §4, tested by the dispatch in §3.

As a motivating example we contrast $V_{t^\star}$ on trap vs. logic tokens (App. Fig. 8).

Specialists and tasks. Two post-training regimes against the same Qwen3 lineage ancestors carry the headline dispatch claim: SFT and GRPO at three scales (0.6B, 1.7B, 4B), trained on Bespoke-Stratos-17k (Bespoke Labs, 2025). Two **cross-family probes** on structurally different model families and post-training methods cover generalization beyond the Qwen3 lineage: **R1-Distill-Qwen-Math-1.5B** (Qwen2.5-Math base, RL-distilled from R1) and **Phi-4-mini-reasoning** (Phi-3 architecture, instruction-tuned reasoning); details in App. E. Three held-out evaluation tasks: **CruxEval** (code input/output prediction), **GSM8K** (grade-school math), **GPQA** (graduate scientific reasoning), each at k=10 rollouts per problem, T=0.6. Bootstrap 95% CIs and full hyperparameters: App. X, K.

Problem-unit framing. We evaluate per-problem-unit (one problem ID summarized over its k failed rollouts) rather than per-cell to avoid averaging over heterogeneous failure populations. Across the 10 evaluation cells, N=1,738 problem-units; N=718 survive Fail@$K \geq 10$ (all K rollouts incorrect at T=0.6). A problem is in Fail@K when standard sampling has already failed, so any recovery measures something beyond exhaustive search.

Dispatch protocol. **Repair@3**: each (pid, operator) gets three single-position injection attempts drawn from the pid's three deepest available conditioning levels (App. N); any-of-three defines the rescue. Per dispatch cell (regime $\times$ $\bar{V}_{\text{HL}}$) and operator we compute U_i (unique rescues), $R_i = U_i/N_{\text{cell}}$, and the dispatch score

$$S_i = U_i \cdot R_i^\gamma, \qquad \gamma = 0.5 \tag{1}$$

(a coverage-rate blend stable across $\gamma \in [0.3, 0.8]$; App. P). The policy routes each pid to its cell's $\arg\max_i S_i$ operator (§3).

Trajectory features (used by every subsequent section). Three problem-level trajectory features

($\bar{J}_{\text{frac}+}$, $\bar{C}$, $V_{t^\star}$) and the routing key $\bar{V}_{\text{traj}}$ are defined formally in Box 4; their geometric motivation is in §2.1. The reference distribution for the steering operators is the lineage pre-training checkpoint (App. R shows ancestor identity is interchangeable within reasonable capability range under a well-calibrated detector).

Recoverability regimes (used in every subsequent section). Running the three features through k-means (k=4, initialised at canonical centroids from a cell-level rule classifier; details in §5 and App. C.1) partitions failed problems into RM-G (Rank Misrouting, geo-local), RM-D (Rank Misrouting, junction-diffuse), DD (Distributed Deformation), and Unresolved (Table 5). Figs. 2–3 ground this partition: the four regimes occupy distinct corners of feature space (Fig. 2) and their per-cell composition already foreshadows the SFT-vs-RL contrast we unpack in §6 (Fig. 3).

Steerable-Hard problem-units (the routable target population). The pids on which any test-time router can show measurable gain are those for which retry alone is inadequate *and* at least one intervention beats retry. Concretely, a problem-unit is **Steerable-Hard** when, at the per-pid level,

$$\underbrace{\max_{o\in\{\text{SL-G, SL-R}, T_{\text{loc}}, \text{DL}\}} r_o - r_{\text{retry}}}_{\text{steerable: } \geq 5\text{ pp}} \quad \text{AND} \quad \underbrace{1 - r_{\text{retry}}}_{\text{hard: } \geq 0.5} \tag{2}$$

where r_o is the rescue rate of operator o at repair@3. 37.1% (528/1,423) of our failed problem-units satisfy both conditions; outside this subset routing is either unnecessary (retry alone rescues, 43%) or futile (no operator rescues, 20%). This is the subset that the framework of §2.1 identifies as the operator class's target population: where Prop. 1 forbids retry recovery (a confidently-wrong specialist preference cannot be inverted by rank-preserving operators) and at least one e-geodesic intervention reaches its $\alpha^\star$ threshold (Prop. 2) within our α grid. All routing claims below report numbers both on the full pool and on this subset, because gains on *Steerable-Hard* are the gains that translate to deployment value.

Three demonstrations of one framework. The three trajectory features are the central object of the paper: they are derived from operator-class geometry, i.e. the structure of available interventions and what they can change about a failed trace (§2.1) and we demonstrate that they encode operator-relevant structure through three downstream uses. The features cluster failures into stable recoverability regimes (§5); the regime distribution of a (model, task) cell audits its post-training method (§6); and the same features support a training-free operator-class routing rule (§3–4). The unifying claim is that failed traces contain operator-class signature, recoverable from three problem-level features, and that signature is deployment-actionable along multiple axes.

3 Routing test-time compute by operator class

A cell-aware dispatch over the five test-time operators tests the framework's prediction that regime-conditional operator choice should beat retry (rank-preserving; Prop. 1) and any uniform single operator.

The dispatch headline. Using the regime label and the steerability split $\bar{V}_{\text{HL}}$ from §2, each pid's cell routes to the operator with the largest cell-aggregate $S_i = U_i \cdot R_i^{0.5}$. Dispatch sends all failed problem-units to a non-retry operator (81.8% to sparse logit-steering, 18.2% to T_{loc}=1.5), beating retry by $\Delta\Sigma S = +35.7$ (ΣS=693.4 vs. 657.7). $\Sigma S = U \cdot R^{0.5}$ is a compute-aware aggregate; the deployment-realistic per-pid pass-rate lift on the Steerable-Hard subset is shown in Fig. 1b. The strongest single operator (uniform SL-R) reaches ΣS=678.7, so no operator wins by more than ~ 21 ΣS; the lift comes from *cell-level* operator preferences. The key decision is binary, whether to inject at all, and within the sparse class the choice (SL-G vs. SL-R) is empirically fungible (§3). The trace-level geometry to inference-time operator-class connection is what prior weight-space (Zhu et al., 2025) and trace-level (Chuang et al., 2024) accounts do not provide.

The feature space predicts where retry fails. Retry rescues only 39.6% of Rank Misrouting (geo-local) problems but 65.1% of Unresolved problems ($\Delta = -25.5$ pp, p<0.0001 unpaired bootstrap; full pairwise contrasts in App. S), so the features predict where additional sampling will pay off before any intervention is run. Cases A–C (Figure 11, App. Y) illustrate the binary decision on prototypical traces.

Steerability is the routing mechanism. Every regime flips its winning operator across the $\bar{V}_{\text{HL}}$ boundary (Table 1); six of ten cells route by steer-

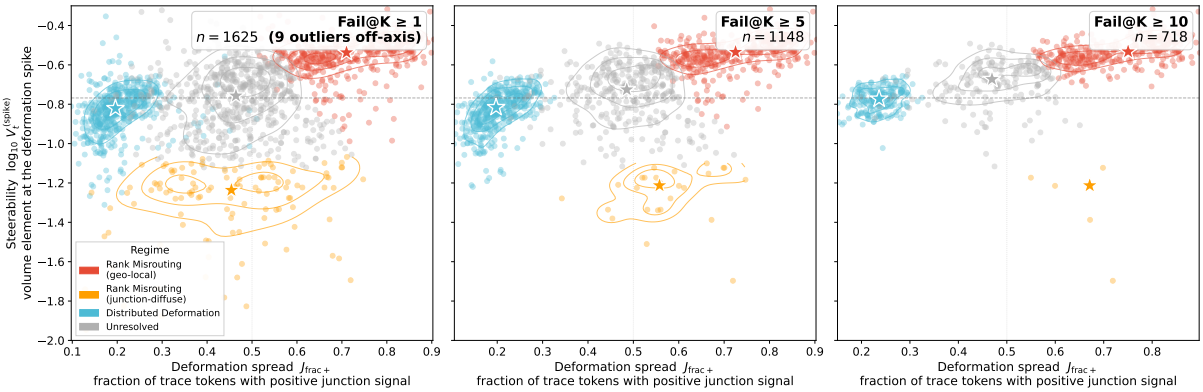

Figure 2: **Problem-level regimes in (deformation spread, junction steerability) space.** Axes are $\bar{J}_{\text{frac}+}$ and $\log_{10}\bar{V}_{t^\star}$ (the classifier features). Each point is one problem-unit; color encodes the nearest-centroid regime. The dashed horizontal line marks the median of $\log_{10}\bar{V}_{\text{traj}}$ (the dispatch H/L routing key, a different stable feature; see Box 4). Star markers are per-regime medians; contours are KDE iso-density. As Fail@K deepens, shallow sampling accidents are filtered out and the feature clusters tighten. The $(\bar{J}_{\text{frac}+}, \log_{10}\bar{C})$ projection appears as Figure 5 (Appendix).

				Rescue rate R			Dispatch score $S = UR^{0.5}$		
Regime	V_t	n	Op	Dispatch	retry	SL-R	Dispatch	Δ vs retry	Δ vs SL-R
DD	H	430	**SL-R**	0.533	0.521	0.533	**167.2**	+5.5	+0.0
DD	L	93	**SL-G**	0.968	0.914	0.957	**88.5**	+7.3	+1.5
RM-G	L	422	**SL-R**	0.422	0.396	0.422	**115.6**	+10.5	+0.0
Unresolved	H	184	**SL-R**	0.707	0.679	0.707	**109.3**	+6.3	+0.0
Unresolved	L	280	T_{loc}=1.5	0.636	0.632	0.621	**141.9**	+1.2	+4.8
RM-D	H	38	**SL-G**	0.658	0.579	0.579	**20.3**	+3.5	+3.5
RM-D	L	89	**SL-G**	0.685	0.674	0.640	**50.5**	+1.2	+4.9
Σ across 10 cells			—	—	—	—	**693.4**	+35.7	+14.7

Table 1: **Regime-aware dispatch by (regime, V_t H/L).** Each row is one dispatch cell; *Op* is the per-cell dispatch choice (operator with the highest cell-aggregate $S = U \cdot R^{0.5}$). Three rescue rates per cell: the dispatch's R, retry R (every cell does retry), and uniform-SL-R R (every cell does SL-R, the globally strongest single operator). Δ vs retry and Δ vs SL-R are dispatch S minus each baseline. Rows in gray (if any) are small-n cells (n<30) where the cell-aggregate score is dominated by sampling noise; included for completeness but not load-bearing for headline comparisons. The dispatch totals U=891, ΣS=693.4 vs. U=860, ΣS=657.7 for retry ($\Delta\Sigma S$=+35.7) and U=879, ΣS=678.7 for uniform SL-R ($\Delta\Sigma S$=+14.7). V_t H/L = median split on $\log_{10}\bar{V}_{\text{traj}}$ (per-pid trajectory-mean Fisher info, pooled ICC 0.88).

ability side. The flips are class-level (sparse vs. retry vs. T_{loc}), not within the sparse class. Steerability supplies the operator-class selection work the regime label alone cannot do. Per-pid winner *prediction* (the exact operator that wins for a specific pid) is not recoverable above baseline (App. Q, Table 23), the dispatch's lift is a population-scale claim. The weaker per-pid *routing-decision* signal (which broad action class is appropriate) does carry actionable information on the Steerable-Hard subset; we return to a training-free per-pid variant in §4.

Per-FLOP efficiency: dispatch is Pareto-dominant against scaled retry. On the two cells where K=64 retry pools exist, dispatch@3 is Pareto-dominant against scaled retry at ∼1.5× iso-rescue compute (Fig. 1; full per-cell curves in App. S).

Operators are mechanistically distinct; the binary inject-vs-retry call dominates within-class choice. On Steerable-Hard, pairwise Jaccard of rescued-pid sets is 0.10–0.59 across families; the all-three-operator intersection is at most 24% while the any-operator union covers 100%, so each operator pulls a partially-different pid subset and operator-class routing extracts lift no uniform sin-

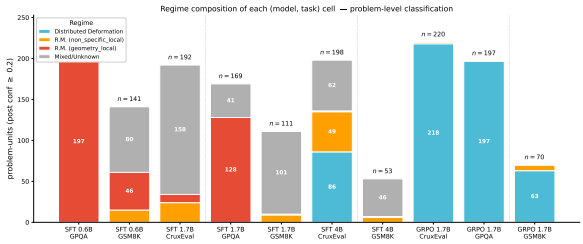

Figure 3: **Regime composition of every (model, task) cell.** Stacked bars; bar height = number of problem-units in the cell. GRPO 1.7B collapses into Distributed Deformation across all three tasks, consistent with reported RL-driven support compression (Chu et al., 2025; Zhu et al., 2025; Shenfeld et al., 2025). SFT spreads across the remaining three regimes by task identity. Audit-channel analysis in §6.

gle operator can match. Within the sparse class, SL-G vs. SL-R is family-dependent in direction but small in magnitude (Qwen3 favors SL-R by $+6.9$ pp; R1-Distill favors SL-G by $+11.4$ pp; Phi-4 tied); the key call is inject vs. retry, not position within the sparse class (App. T, Table 30).

4 Prospective test-time routing policy

§2.1 prescribes a per-problem rule: route to the operator whose corresponding feature is the dominant geometric signature for this trace. The dispatch in §3 consumes operator outcomes to define its regime labels and steerability split; we now ask the complementary, prospectively actionable question: can the same three features *alone*, computed from a cell's failed rollouts with no operator assay on the test cell, instantiate that rule?

The rule. For each failed pid the rule z-normalises the three classifier features and applies the *paper-derived* mapping

$$\hat{o}(p) = \begin{cases} \text{DL} & \text{if } \arg\max(\hat{z}_{J_+},\, \hat{z}_{\log C},\, \hat{z}_{\log V_{t^\star}}) = \hat{z}_{J_+}, \\ \text{SL-G} & \text{if } \arg\max(\cdot) = \hat{z}_{\log C}, \\ T_{\text{loc}}{=}1.5 & \text{if } \arg\max(\cdot) = \hat{z}_{\log V_{t^\star}}, \end{cases} \tag{3}$$

mirroring §5's regime semantics (broad junction signal → distributed deformation → DL; sharp single junction → geometry-local → SL-G; high entropy headroom at the spike → entropy-brittle → local temperature lift). No gate is applied, every failed pid is routed to the argmax operator. Per-pid cost is dominated by the routing operator's single attempt (mean $\approx$1.4 single-rollout units across the chosen operator mix); feature extraction is $O(N_{\text{tokens}})$ arithmetic on top-k logprobs that vLLM can save during the original spec rollout, and is negligible relative to the intervention itself, so the policy runs at *comparable retry cost*.

Result (Table 2). On the full n=1,423 pool, the prospective policy ties retry: $\Delta = +0.01$ pp, 95% CI $[-1.22, +1.24]$ (paired bootstrap, not distinguishable from zero). On the *Steerable-Hard* subset (§2.1; 37.1% of pids), the rule lifts rescue by $+12.2$ pp (from 0.163 to 0.285; 95% CI $[+10.0, +14.4]$, n=528, $p < 10^{-3}$). The matched -7.2 pp drop on non-SH (63%; 95% CI $[-8.5, -5.9]$) is the cancellation that preserves global-mean accuracy; this is the expected behavior of a subset-conditional method. Zero pids are routed to retry; the fallback for low-confidence pids is SL-R (§2.1). For context, **DiagnosticOracle** (the full per-cell operator assay; Table 2) reaches $+15.7$ pp on Steerable-Hard at $\sim 47\times$ the per-pid compute, and **ProblemOracle** (per problem-unit $\arg\max$, not deployable) reaches $+30.4$ pp. **Feature-only** therefore recovers 78% of DiagnosticOracle's Steerable-Hard lift at $\sim 1/47$ of its cost, using only features whose computation is negligible relative to a rollout.

The framework predicts the routing-relevant quantity, not a specific feature set. Multiple Fisher-aggregate feature sets achieve comparable SH lift ($+11.7$ to $+13.8$ pp at near-zero global cost; App. D), confirming the framework's prediction that local Fisher information is the routing-relevant quantity. The geometric instance reported in the body is one interpretable derivation within that family. Substituting SL-R for SL-G at the $\log_{10} \bar{C}$ branch loses only -0.4 pp on the Qwen3 SH headline, so a deployer without a junction detector can substitute SL-R and retain $> 96\%$ of the lift; the dispatch's binary decision is operator-class (sparse vs. temperature vs. dense), not position within the sparse class (§3, App. T).

What this validates. §3 showed that the feature space supports population-level operator routing; §4 shows the complementary, *prospectively actionable* variant that is usable at test time with no operator assay on the test cell. This converts the dispatch's regime structure from a diagnostic into a deployment artifact: a fixed rule table that takes failed rollouts in and emits an operator choice. We discuss remaining limits (per-pid winner identifiability, cross-task transfer of the rule table) in §9.

Table 2: **Prospective test-time routing policy compared to baselines and oracles. Feature-only** is our training-free rule (§4, Eq. 3); it consumes only failed-rollout features at test time. **Compute** is measured in single-rollout-equivalent units (one retry attempt $= 1.0$) and reflects only the policy's deployment-time generation cost; feature extraction from saved logprobs is $O(N_{\text{tokens}})$ arithmetic and negligible relative to a rollout. *All failed pids*: pool-weighted across the 10-cell evaluation set (n=1,423). *Steerable-Hard subset*: the 37.1% of failed pids where retry fails the majority of the time AND at least one intervention beats retry by $\geq$ 5 pp (Eq. 2). The dispatch in §3 consumes operator outcomes to define regimes; the prospective policy does not.

Policy		**Needs**	**Compute**	**All failed pids**		**Steerable-Hard subset**	
Name	**Routing info uses…**	**labels?**	**× retry**	rescue	Δretry	rescue	Δretry
Retry	no information, no test-cell compute beyond resampling	no	1.0	0.372	+0.000	0.163	+0.000
Feature-only	winner-take-all over 3 trajectory features; no training	no	1.4	0.373	+0.000	0.285	+0.122
Outcome-trained	RF classifier on the cell's own per-pid operator outcomes (assay cost = DiagnosticOracle's)	yes	66.5	0.374	+0.001	0.302	+0.139
DiagnosticOracle	runs every operator on the test cell, picks modal-best	yes	66.5	0.390	+0.017	0.321	+0.157
ProblemOracle	per problem-unit $\arg\max$ over operators (not deployable)	yes	66.5	0.514	+0.142	0.468	+0.304

"Needs labels?" = the policy requires per-pid operator outcomes from the test cell (i.e., running every operator on every pid; ∼5 operators × K=10 attempts).

5 Features that characterize the regimes

The three features partition failed problems into four regimes via k-means initialised at canonical centroids (App. C.1). Two properties of the resulting partition matter for the dispatch: (i) 99.6% of RM-G sits above the $\log_{10} \bar{V}$ median while 85% of DD sits below, the two regimes the dispatch differentiates most sharply occupy opposite ends of the steerability axis; (ii) cluster tightness improves monotonically with Fail@K depth: silhouette $0.45 \rightarrow 0.58$ and median within-cluster distance $0.68 \rightarrow 0.48$ (-30%) between Fail@$K{\geq}1$ and Fail@$K{\geq}10$ (matched-n bootstrap, n=718, 95% CIs disjoint at every stratum; Fig. 2), so the harder the failure under standard sampling, the more informative the failed trace.

6 Failure topography as an audit channel for post-training

Different post-training methods produce different recoverability topographies. This section shows the topography is recoverable from failed traces alone, providing an audit channel for post-training without training-time access. Without any training-time access and without any label on the rollouts saying *this is from GRPO* or *this is from SFT*, the regime distribution of a (model, task) cell identifies which post-training method produced the model, recovering the SFT-vs-RL contrast (Chu et al., 2025; Zhu et al., 2025; Shenfeld et al., 2025) from failed rollouts alone. The same features drive the routing example in §3.

GRPO concentrates 97.4% **of problem-units** (487/500) in Distributed Deformation across CruxEval, GPQA, and GSM8K. §2.1 predicts this: the diffuse, off-principal RLVR weight updates characterized at the weight- (Zhu et al., 2025) and aggregate-output (Chu et al., 2025; Shenfeld et al., 2025) levels manifest in the trace as high $\bar{J}_{\text{frac+}}$ and low $\bar{C}$, exactly the Distributed Deformation signature, recovered here from failed rollouts alone with no training-time access. **SFT distributes by task identity, not scale**: 79.7% of Rank Misrouting (geo-local) problem-units (341/428) come from SFT 0.6B and 1.7B on GPQA; 46.5% of Unresolved (266/572) comes from SFT 0.6B and 1.7B on GSM8K and SFT 1.7B on CruxEval. Detailed per-task percentages are in Appendix Z; the operator-relevance of these regime distinctions is the main claim.

Mechanism (weight-space ↔ inference-time). Spectral preservation (top-k singular structure intact; Zhu et al. 2025) and large diffuse logit deviations are not in tension: off-principal updates can accumulate to produce meaningful inference-time behavioural change with minimal spectral drift. Our audit channel measures the inference-time manifestation of that micro-direction movement, on a different model family and training set than the weight-space evidence.

The audit channel is predictive, not just descriptive. A k=1 nearest-neighbour classifier on the 4-D cluster-fraction histograms of each (model, task) cell labels SFT vs. RL on 14 cells with 84.3±4.3% accuracy across 10 random 5-fold permutations (+20 pp over a 64.3% majority-class baseline; LOCO equivalent is 85.7%, 12/14). All three GRPO cells, all seven Qwen3 SFT cells, and both Phi-4-mini-reasoning cells are classified correctly; the two errors are both R1-Distill-Qwen-Math-1.5B cells classified as SFT, consistent with distillation muting the RL signature in the failed-trace topography.

This connects test-time diagnosis to post-

training characterisation: failures from different post-training regimes carry distinguishable signatures in feature space, and those signatures are what the dispatch consumes. Prior work has reached related conclusions from the training side; we approach the same divide from the failure trace.

Cross-family transfer. Two held-out-family probes give the contrasting signatures the framework predicts: **R1-Distill-Qwen-Math-1.5B** has SL-G/SL-R/T-loc Jaccard 0.10–0.16 on SH (far below Qwen3's 0.28–0.50) and 82.9% [76.9, 86.5] of its failed pids in the Distributed Deformation cluster, while **Phi-4-mini-reasoning** has Jaccard 0.30–0.59 within the Qwen3 SFT/GRPO range. The features isolate R1-Distill as the support-collapsed family and group Phi-4 with SFT-shaped topography, from failed rollouts alone, despite the underlying architecture differences. The training-free routing rule from §4 also transfers across families: 8 of 9 cross-family z-scaler transfers between the three families fall within the target's bootstrap 95% CI of its self-baseline lift (App. E, Tab. 11). Full numbers in App. E.2.

Three qualitative case studies illustrating per-failure routing under the V_{traj} dispatch (Cases A, B, and C) and a sweep of response profiles across representative (model, task) cells appear in App. Y.

7 Related Work

Post-training contrast at weight, behavioral, and KL-proximity levels. Chu et al. (2025) contrast SFT and RL with outcome rewards behaviorally (SFT memorizes, RL generalizes); Zhu et al. (2025) characterize the weight-space mechanism (RLVR updates land in low-curvature, off-principal subspaces while preserving top-k spectral structure); Shenfeld et al. (2025) show online RL implicitly minimizes KL to the base model. All three require training-time access. Our audit channel (§6) provides the inference-time logit-space counterpart that connects these three perspectives to failure-trace geometry, with Distributed Deformation as the predicted inference-time signature of Zhu et al.'s off-principal mechanism.

Trace-level interventions. DoLa (Chuang et al., 2024) and proxy tuning intervene in logit space without a class-localizing diagnostic. Engineering differences: **(i)** we contrast against a separate pre-trained checkpoint (one additional forward pass) rather than an earlier layer of the same model or a smaller fine-tuned model; ancestor identity is interchangeable within reasonable capability range under a well-calibrated detector (App. R); **(ii)** regime-conditional operator selection vs. uniform application; **(iii)** diagnostic-first, with operators deliberately simple. No head-to-head empirical comparison; see §9.

Inference-time composition and rank-preserving decoding. MIXIE (Sanyal et al., 2025) uses logit-space steering for multi-objective control; we treat the ancestor as a *structural probe* for rank-inversion in a single specialized model. Mattei and Loureiro (2025) characterize aspects of temperature scaling that motivate the temperature probe (App. A); RLHF calibration degradation (Xie et al., 2024) we reframe as a diagnostic rank-order collapse.

Forgetting and post-training degradation. Self-distillation (Zhang et al., 2026; Shenfeld et al., 2025) and temporal sampling (Li et al., 2025) address reasoning decay behaviorally; we localize the same decay at the distributional level and identify which intervention recovers which regime. The fork/lock typology of Zhang et al. (2026) maps onto our logit-steering lift: their training-time fix reshapes support, our training-free SL-G/SL-R/DL operators recover a junction-conditional analogue via ancestor contrast, sidestepping their decode-only impossibility.

8 Conclusion

Failed traces are a measurement instrument. Aggregate accuracy is a one-bit readout; the three trajectory features are a higher-dimensional readout of the same population, and the readout has structure: operator-class structure derived from information geometry, not fit to data. Once you have the instrument, three things follow. **(1)** Failure populations are heterogeneous in a principled way; the four recoverability regimes are stable across Fail@K depths. **(2)** Post-training methods leave inference-time signatures, and the audit channel reads them off failed rollouts alone, connecting to weight-space accounts without needing weight-space access. **(3)** The signatures are deployment-actionable, which the training-free routing rule on Steerable-Hard demonstrates; the routing example is a demonstration, not the headline.

9 Limitations

Scope of the routing example. The routing rule of §4 requires three conditions: (i) a post-trained specialist with an available reference distribution that disagrees on failure trajectories, (ii) a bounded test-time intervention budget, and (iii) failures with localizable geometric structure rather than uniform high entropy. Open-ended generation tasks (translation, dialogue, creative writing) do not satisfy (iii), because logit variance is high everywhere when the task is genuinely under-constrained, and are out of scope for the current framework.

The DL branch is dominated empirically. The framework of §2.1 predicts DL (dense logit steering) should win the Distributed Deformation regime, which is the dominant regime on GRPO (97.4% of GRPO problem-units). In our case the dispatch never picks DL: sparse interventions (SL-G, SL-R) reach comparable rescue at lower per-attempt cost, and the dispatch's $\arg\max S_i$ selects them instead. This is a gap between the geometric prediction (DL should be the right operator for diffuse deformation) and the per-FLOP recipe at the current budget; resolving it requires deep-K dispatch evaluation (repair@K at $K{=}10, 32, 64$ per operator) we leave to future work.

Model-family and scale scope. Headline cells use Qwen3 SFT (0.6B/1.7B/4B) and GRPO 1.7B; cross-family generalization is covered by the R1-Distill-Qwen-Math-1.5B and Phi-4-mini-reasoning probes in §6 and App. E. Behavior at $\geq$ 7B and with heavily RLHF-aligned models is out of scope. Verbal self-correction (Self-Debug) is partially orthogonal to distributional intervention and is handled separately (Appendix B.1).

References

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

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, David Z. Pan, Zhangyang Wang, Yuandong Tian, and Kai Sheng Tai. *The Path Not Taken: RLVR Provably Learns Off the Principals*. In *NeurIPS 2025 Workshop on Efficient Reasoning* (spotlight); arXiv:2511.08567, 2025.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. *RL's Razor: Why Online Reinforcement Learning Forgets Less*. arXiv preprint arXiv:2509.04259, 2025.

Ayan Sanyal, Xiang Ren. *MIXIE: Multi-Expert Instruction Following via Inference-time Mixing*. arXiv preprint arXiv:2502.10777, 2025.

Pierre-Alexandre Mattei and Bruno Loureiro. *The Well-Tempered Classifier: Some Elementary Properties of Temperature Scaling*. arXiv preprint arXiv:2502.14862v1, 2025.

Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. *Embarrassingly Simple Self-Distillation Improves Code Generation*. arXiv preprint arXiv:2604.01193, 2026.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. *Self-Distillation Enables Continual Learning*. arXiv preprint arXiv:2501.19897, 2025.

Yuetai Li, Zhangchen Xu, Fengqing Jiang, Bhaskar Ramasubramanian, Luyao Niu, Bill Yuchen Lin, Xiang Yue, and Radha Poovendran. *Temporal Sampling for Forgotten Reasoning in LLMs*. arXiv preprint arXiv:2505.20196, 2025.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. *DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models*. arXiv preprint arXiv:2309.03883, 2023. In *Proceedings of the International Conference on Learning Representations*, 2024.

Shun-ichi Amari. *Information Geometry and Its Applications*. Springer, 2016.

Johnathan Xie, Annie S. Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. *Calibrating Language Models with Adaptive Temperature Scaling*. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Bespoke Labs. *Bespoke-Stratos-17k: A Dataset of DeepSeek-R1-Distilled Reasoning Chains*. https://huggingface.co/datasets/bespokelabs/Bespoke-Stratos-17k, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. In *International Conference on Learning Representations (ICLR)*; arXiv:2203.11171, 2023.

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.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. *Learning to Summarize with Human Feedback*. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 3008–3021, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. *Evaluating Large Language Models Trained on Code*. arXiv preprint arXiv:2107.03374, 2021.

A Formal Method Details

A.1 Junction detection

High level. Both junction detectors used in this paper target positions of anomalously high specialist-ancestor disagreement, and both calibrate *within* the problem rather than against an external population, this is what makes the firing rule task- and scale-independent. They differ in what they normalize against and at what granularity: Detector A standardizes against positions *of the current trace* and fires on a single token; Detector B standardizes against positions across *this problem's k failed rollouts* and fires on a sliding window. Detector A is what the lineage-specificity ablation in App. R uses (for protocol parity with the original ladder); Detector B is what the dispatch in §3 uses, because the windowed Fisher-information normalization of Detector B is what makes $V_{t^\star}$ the natural local steerability measure (§5). The rest of this appendix gives the formal definitions.

Detector A: trace-relative percentile. Given a reasoning trace $x_{1:T}$ sampled from the specialist and the corresponding specialist and ancestor logit sequences z_S, z_A, the **demotion score** at step t is

$$D_t = \log P_S(x_t) - \log P_A(x_t), \tag{4}$$

the specialist's preference for the chosen token relative to the ancestor's baseline. To strip out trace-level offsets we standardize within the trace, $\tilde{D}_t = (D_t - \mu_D)/\sigma_D$ with μ_D, σ_D the empirical mean and standard deviation across positions. An **intervention-sensitive junction** fires at the first position whose $\tilde{D}_t$ exceeds the 90th percentile of values in the same trace:

$$t^* = \min\{t \mid \tilde{D}_t \geq \tilde{D}_{(90)}\}. \tag{5}$$

Detector B: per-problem windowed quantile. Detector B sums a per-token "junction intensity" over a sliding window of width w: $S_t = \sum_w J_t$, where $J_t = \max(Z_t^{\text{path}}, Z_t^{\text{cov}})$ combines two normalized deformation signals. $Z_t^{\text{path}} = \max(0, (r_{\text{chosen}} - \mu_r)/\sqrt{I_r})$ is the e-geodesic log-ratio at the chosen token, centered and scaled *per-token* by the expectation μ_r and Fisher information I_r of r under the specialist's local top-k distribution (Fisher information is the natural metric on the categorical output space at that position). $Z_t^{\text{cov}} = \max(0, (G_t^{\text{cov}} - \mu_{\text{cov}})/\sigma_{\text{cov}})$ is the m-geodesic coverage gap, centered and scaled *per-problem* by the mean and standard deviation of G^{cov} across the pid's k failed rollouts. Detector B fires the first time the windowed sum exceeds a per-problem threshold λ_J, set as the 0.99 quantile of S_t across all positions in the same pid's pool. The steering direction $\tau_t \in \{\text{path}, \text{cov}\}$ is set by whichever of $Z_t^{\text{path}}, Z_t^{\text{cov}}$ is larger at the firing position.

A.2 Dual Generation Strategies: Mechanism vs. Utility

We use two generation strategies, separating mechanistic identification from deployment evaluation.

Mechanistic Verification ($T = 0$): For all experiments establishing the mechanistic link between logit geometry and recovery (Iterative Repair, Junction Ablations, and Random Controls), we use **Greedy Decoding** ($T = 0$) for continuation after the intervention. Greedy completion ensures that the resulting trajectory is a deterministic consequence of the intervention, removing sampling variance as a confounder. This provides a conservative lower bound on recovery and aligns with our theoretical framework $(\tilde{D}_t, \alpha^*)$, which is defined by the rank-order of the distribution's argmax. At $T > 0$, the relationship between logit rank and sampled tokens becomes probabilistic, making the interpretation of α^* as a hard crossover threshold conceptually inconsistent.

Deployment Utility Simulation ($T = 0.6$): To measure the practical benefit of the deployment-time routing policy (Result 4), we use **Temperature Sampling ($T = 0.6$)** for both pre- and post-intervention rollouts for dense trajectory logit steering ($T = 0.6$ in the table). This strategy simulates real-world usage where specialists are deployed under non-greedy decoding. Reporting the deployment simulation at $T = 0$ would compress the gap between conditions, as greedy decoding often hurts baseline specialist performance on reasoning tasks.

A.3 Theoretical Justification

The diagnostic value of the ladder rests on two closed-form results.

Proposition 1 (Token-Level Rank Preservation)**.** *For any two tokens $x_i, x_j \in V$ and any $T > 0$, if the specialist logit $z(x_i) > z(x_j)$, then $p_{1/T}(x_i) > p_{1/T}(x_j)$.*

Proof. The log-odds ratio is $\log \frac{p_\beta(x_i)}{p_\beta(x_j)} = \beta(z(x_i) - z(x_j))$. Since $\beta = 1/T > 0$, the sign is invariant to T. □

Proposition 2 (Rank Inversion via SL-G)**.** *If the Specialist ranks x_{trap} above x_{logic} ($\Delta_S > 0$) but the Ancestor ranks x_{logic} above x_{trap} ($\Delta_A := z_A(x_{logic}) - z_A(x_{trap}) > 0$), there exists a critical threshold $\alpha^* = \frac{\Delta_A}{\Delta_A + \Delta_S} \in (0, 1)$ such that for all $\alpha < \alpha^*$, the mixed model recovers x_{logic} as the mode.*

Proof. The mixed logit gap is $\Delta_\alpha = \alpha\,\Delta_S - (1 - \alpha)\,\Delta_A$. Solving $\Delta_\alpha < 0$ yields $\alpha < \Delta_A/(\Delta_A + \Delta_S) =: \alpha^*$. □

Together, Proposition 1 establishes that temperature cannot invert ranks, and Proposition 2 gives the closed-form condition under which logit steering can.

A.4 Failure-Trajectory Metrics

Path deformation score. At each token position t in a failed trace $x_{1:T}$, the **path deformation score** measures how strongly the specialist prefers the sampled token over the ancestor's baseline:

$$D_t^{\text{path}} = \log p_S^T(x_t) - \log p_A^T(x_t) \tag{6}$$

where p_S^T and p_A^T are the specialist and ancestor distributions at sampling temperature T. Positive values indicate that the specialist assigns higher probability than the ancestor to this token choice. The trajectory-mean $\bar{D}^{\text{path}} = \frac{1}{T}\sum_t D_t^{\text{path}}$ is stored as column `Delta_path_mean` in the routing feature table; the value at the detected junction position is stored as `Delta_path_at_that`.

Junction intensity. The **junction intensity** J_t is a Fisher-normalized, one-sided version of D_t^{path}, centering by the KL divergence and normalizing by the Fisher information of the path direction:

$$J_t = \left[\frac{D_t^{\text{path}} - \mu_t^r}{\sqrt{I_t^r + \varepsilon}}\right]_+ \tag{7}$$

where $\mu_t^r = \mathrm{KL}(p_S^T \| p_A^T)$ is the expected log-likelihood ratio under p_S (a local KL divergence estimate), $I_t^r = \mathrm{Var}_{y\sim p_S}[r_t(y)]$ is the Fisher information of the e-geodesic direction at position t, and $[\cdot]_+$ denotes the positive part. Junction intensity fires when the specialist's choice of x_t is anomalously more specialist-favored than the specialist's own average deformation from the ancestor at this position—detecting rank misrouting beyond what background post-training drift would predict.

The trajectory mean and standard deviation are stored as `J_approx_mean` and `J_approx_std`. The **Junction Concentration** C_J is defined as their ratio:

$$C_J = \frac{\sigma(J_t)}{\mu(J_t) + \varepsilon} = \frac{\texttt{J_approx_std}}{\texttt{J_approx_mean}} \tag{8}$$

High C_J indicates that deformation is sharply localized at a small number of positions; low C_J indicates diffuse deformation spread across the trajectory. We use C_J as the canonical concentration statistic throughout.

Coverage deformation score. The **coverage deformation score** measures how much mass the specialist has removed from the ancestor's plausible alternative set:

$$G_t^{\text{cov}} = \log\left[\frac{p_A(A_t^k)}{p_S(A_t^k) + \varepsilon}\right] \tag{9}$$

where $A_t^k = \mathrm{TopK}(p_A^T, k{=}20)$ is the ancestor's top-k vocabulary set at position t, and $p_M(A_t^k) = \sum_{y\in A_t^k} p_M^T(y)$ is the total probability mass model M assigns to that set. The ratio normalizes by the ancestor's own belief in its plausible set: a diffuse ancestor (low $p_A(A_t^k)$) should not produce a large signal. G_t^{cov} asks: *relative to how much the ancestor believes in its own alternatives, how much has the specialist collapsed away from them?* This is the m-geodesic (coverage) complement to the e-geodesic (path) signal in J_t. Column names: `G_cov_mean`, `G_cov_at_that`.

Temperature submodel variance. The **temperature submodel variance** V_t is the Fisher information of the local temperature submodel—the probability-weighted variance of specialist logits over the top-K vocabulary:

$$\begin{aligned} V_t &= \mathrm{Var}_{y\sim p_{S,K}}[z_S(y)] \\ &= \sum_{y\in \mathrm{TopK}} p_{S,K}(y)\left(z_S(y) - \mathbb{E}_{p_{S,K}}[z_S]\right)^2 \end{aligned} \tag{10}$$

where $p_{S,K}$ is the specialist distribution renormalized over the top-K vocabulary ($K = 100$). High V_t indicates that the distribution is entropy-susceptible: small temperature changes produce large mass shifts and the specialist is in an intermediate state where temperature scaling has leverage. Low V_t indicates either a collapsed (very sharp) or diffuse distribution where temperature perturbation cannot invert any rank. V_t is stored as `V_mean` or `logit_var_mean`. **Implementation note:** do not use `logit_S.var()` (unweighted variance over the full vocabulary); the probability-weighted top-K formula is required to recover the Fisher information interpretation (Mattei and Loureiro, 2025).

Summary: routing feature roles. Table 3 summarizes the intuitive question each feature answers and its operational role.

In §5 we visualize the regime-level distributions of these features.

A.5 Problem-Level Trajectory Features (moved from §2)

B Extended Result Details

B.1 Self-Debug Comparison

Self-Debug operates on a verbal reasoning axis and provides a fifth diagnostic perspective contrasting verbal introspectability against distributional recoverability. In Table 4, we compare Self-Debug against SL-G and standard Retry on strictly matched failure sets (up to 3 iterative rounds, early stopping on first success). The 4B CruxEval Rank Misrouting result is most informative: SL-G rescues 57.1% of matched problems, whereas Self-Debug reaches only 22.4% and Retry 28.6%; verbal backtracking cannot reach a logic error the model cannot name, and extra samples alone do not suffice. On GSM8K, the pattern reverses: at 1.7B, Retry (95.2%) and Geo (94.4%) both substantially outperform Self-Debug (86.3%), consistent with a sample-recoverable regime (now absorbed into Unresolved) where resampling is enough. We include Self-Debug as an external reference point, not as part of the core four-probe profile. The conditioning event differs from Table 13: rows here restrict to the Self-Debug-attempted subset, which is a different (and typically smaller) problem slice than the Fail@K strata used for the ladder.

B.2 Regime-name glossary (moved from §2)

B.3 Dispatch Score Hyperparameter Sensitivity

The Dispatch Score used to assign regimes is defined as $S_i = U_i \cdot R_i^{\gamma}$, where U_i represents the unique coverage of probe i relative to cheaper interventions (in the cost order: retry $\rightarrow$ SL-R $\rightarrow$ SL-G $\rightarrow$ DL) and R_i is its absolute reliability. To assign a regime, we take the probe with the maximum S_i; if this maximum score is below a threshold τ, the problem is assigned to the Unresolved regime.

Table 6 evaluates the stability of these regime assignments across a grid of hyperparameters ($\gamma \in \{0.3, 0.5, 0.7\}$ and $\tau \in \{0.02, 0.03, 0.05\}$). The assignments are remarkably robust; the primary variation is limited to the boundary between weak signal and Unresolved, confirming that the regime taxonomy is driven by the data rather than post-hoc parameter fitting.

B.4 Cross-Family Replication

Table 7 reports preliminary Llama-3 results. For Llama-3.2-3B-Instruct on CruxEval, three ancestors were tested: the same-family Llama-3.2-3B base (26.9%), Llama-3.1-8B (29.5%), and R1-Distill-Llama-8B (35.9%). The two non-same-family comparators rescue more than the same-family base, but both have substantially higher effective capacity, so the cross-family signal is mixed and capacity-confounded under this protocol. We report the numbers as a single cross-family probe rather than a controlled test; the within-family Qwen3 results (Appendix R) are the cleaner ablation, and they show that ancestor identity is interchangeable within the capability range we tested under a well-calibrated detector.

C Regime Separation in 2D Trajectory Feature Space

Figure 2 plots failed rollouts across three difficulty strata (Fail@$1, 5, 10$) in the space of two interpretable trajectory features computed from failed rollouts alone, without observing any intervention

Feature	Intuitive question	Data column
$\bar{D}^{\text{path}}$	Did specialist over-commit on average?	`Delta_path_mean`
C_J	Is deformation localized or diffuse?	`J_approx_std / J_approx_mean`
$\bar{G}^{\text{cov}}$	Did specialist abandon ancestor's options?	`G_cov_mean`
V_t	Is the distribution still temperature-movable?	`V_mean`

Table 3: Trajectory features used as topography dimensions. "Option loss" ($\bar{G}^{\text{cov}}$) and "persuadability" (V_t) are the intuitive names used in figures and tables.

Box 3: Problem-Level Trajectory Features

Each failed rollout r of problem p contributes a per-token junction intensity $J_t^{(r)} \in \mathbb{R}_{\geq 0}$ (§A.1). We compress each rollout into three scalars and aggregate over $\mathcal{R}_p$ by rollout-mean.

(i) Deformation spread, the fraction of trace tokens with positive junction signal; how broad the failure is across the trace:

$$J_{\text{frac}+}^{(r)} = \frac{1}{T_r} \sum_{t=1}^{T_r} \mathbb{I}[J_t^{(r)} > 0], \qquad \bar{J}_{\text{frac}+}^{(p)} = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} J_{\text{frac}+}^{(r)}. \tag{11}$$

(ii) Junction concentration, the ratio of peak to mean junction intensity; whether the signal spikes at one position or remains diffuse:

$$C^{(r)} = \frac{\max_t J_t^{(r)}}{\frac{1}{T_r} \sum_t J_t^{(r)}}, \qquad \bar{C}^{(p)} = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} C^{(r)}. \tag{12}$$

(iii) Junction steerability $V_{t^\star}$, the Fisher information of the specialist's local temperature submodel at the junction spike $t^\star = \arg\max_{t>0} J_t^{(r)}$, averaged across the problem's failed rollouts. Used as the per-problem classifier feature.

$$V_t^{(r)} = \mathrm{Var}_{x \sim p_S(\cdot|\mathrm{ctx}_t)} \big[\log p_S(x \mid \mathrm{ctx}_t)\big], \qquad \bar{V}_{t^\star}^{(p)} = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} V_{t_r^\star}^{(r)}. \tag{13}$$

Routing key. The dispatch H/L split uses the *trajectory mean* $\bar{V}_{\text{traj}}$ (average of $V_t^{(r)}$ over the trace, excluding the boundary window T_w), not $V_{t^\star}$, a deliberate split between the two roles (App. Q).

$$\bar{V}_{\text{traj}}^{(p)} = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \frac{1}{T_r - T_w} \sum_{t=T_w}^{T_r} V_t^{(r)}. \tag{14}$$

outcomes. Increasing Fail@K filters out shallow sampling accidents and mixed cases, denoising the diagnostic space. By Fail@10, the regimes occupy largely non-overlapping regions, confirming that the mechanistic routing logic (Figure 11) generalises to stable failure modes.

Feature definitions. *Deformation spread* ($J_{\text{frac}+}$) is the fraction of trace tokens at which the per-token junction signal J_t is positive, where $J_t = \Delta_t^{\text{path}} \cdot G_t^{\text{cov}}$ combines the specialist–ancestor logit delta with the Gaussian coverage gap. It measures how widely fine-tuning has rewritten the specialist's trajectory relative to the ancestor. $J_{\text{frac}+}$ is a raw proportion in $[0, 1]$ and is *not* normalised.

Junction concentration rank is the within-dataset percentile rank of $C_J = J_{\max}/J_{\text{mean}}$, the ratio of the peak junction signal to the mean junction signal across the trace. The underlying ratio C_J ranges

			Problem-level rescue (%)				Rollout-level rescue (%)			
Model / Task	N	**Avg. Att.**	SD	SL-G	retry	Δ(SL-G$-$retry)	SD	SL-G	retry	Δ(SL-G$-$retry)
SFT 0.6B / CruxEval	195	7.81	60.0	**42.1**	65.1	-23.0	7.7	**4.1**	7.2	-3.1
SFT 0.6B / GSM8K	195	7.81	91.8	**82.1**	51.8	+30.3	11.8	**11.2**	12.9	-1.7
SFT 1.7B / CruxEval	201	8.75	74.6	**75.1**	59.2	+15.9	8.5	**4.9**	6.6	-1.7
SFT 1.7B / GSM8K	124	12.81	86.3	**94.4**	95.2	-0.8	6.7	**3.3**	1.9	+1.4
SFT 4B / CruxEval	49	6.37	22.4	**57.1**	28.6	+28.5	3.5	**5.3**	7.1	-1.8
SFT 4B / GSM8K	56	7.34	92.9	**92.9**	76.8	+16.1	12.7	**9.8**	8.5	+1.3

Table 4: **Self-Debug vs. SL-G vs. retry: matched problem-level and rollout-level rescue rates.** We compare verbal self-correction (**SD**), sparse logit-steering at the detected junction (**SL-G**), and standard temperature sampling (**retry**) on strictly matched failure sets (intersection of pids attempted by both SD and SL-G). All three methods use up to 3 iterative repair rounds with early stopping on first success. Δ(SL-G$-$retry) shows the per-cell lift of sparse logit-steering over re-sampling; SL-G beats retry on CruxEval at every scale and ties at GSM8K (where retry is already strong). Self-Debug uses a verbal explanation step at T=0 followed by a revision at T=0.6; SL-G uses greedy T=0 decoding after a local logit-steering intervention (α=0.7) at the detected junction; retry re-samples from the already-failed rollout pool at T=0.6. Matched to the subset of problems on which Self-Debug was attempted; this differs from the Fail@K stratum used in Table 13.

Canonical name	Short	Mechanism signal
Rank Misrouting (geo-local)	RM-G	sharp spike, steerable
Rank Misrouting (junction-diffuse)	RM-D	sharp spike, unsteerable
Distributed Deformation	DD	broad diffuse deformation
Unresolved	,	no dominant signal

Table 5: **Regime-name glossary.** Canonical name is used in body prose; the *Short* column appears in compact figure legends and dispatch rows.

from approximately 43 to 1262 with a heavy right tail; the percentile rank maps it monotonically to $[0, 1]$, preserving all ordinal structure while removing the scale. The two axes are therefore not directly comparable: the x-axis is on an absolute fraction scale, while the y-axis is relative to the dataset distribution.

Why these two features. $G^{\text{cov}}_{\text{frac}+}$ (support-contraction fraction) is near zero for GRPO specialists because GRPO training does not contract the specialist's token support relative to the ancestor, GRPO broadens rather than narrows the output distribution. $J_{\text{frac}+}$ captures both SFT support contraction and GRPO logit-path divergence ($\Delta^{\text{path}}_{\text{frac}+} \approx 0.18$ for Distributed Deformation cells) and is therefore non-zero for all regime types.

Interpretation. The four regimes occupy largely non-overlapping regions, consistent with the mechanistic routing logic established in the main text (Figure 11): *Rank Misrouting geo-local* (logit-steer) at high spread / high concentration; *Distributed Deformation* (broad deformation) at low spread / low concentration. Centroid stars mark the per-regime median position. Within **Rank Misrouting (junction-diffuse)**, SFT and GRPO models occupy distinct sub-regions, consistent with the observation that GRPO training shifts the logit-variance distribution relative to SFT while preserving the broad failure topology.

Alternative projection. The classifier uses three features, so there are three natural 2D projections of the regime structure. The main-text Figure 2 uses $(\bar{J}_{\text{frac}+}, \log_{10}\bar{V})$ because the steerability axis $\bar{V}$ is the discriminator that drives the dispatch (§3). Figure 5 shows the complementary $(\bar{J}_{\text{frac}+}, \log_{10}\bar{C})$ projection, in which junction concentration $\bar{C} = J_{\max}/J_{\text{mean}}$ separates Rank Misrouting (geo-local) from the other regimes more sharply but does not surface the H/L steerability split. The two views are complementary; the classifier sees both axes simultaneously and neither projection drops information.

C.1 Regime classifier procedure

In two steps. Per-pid regime labels come from a two-stage pipeline:

1. **Canonical centroids**: a cell-level rule classifier (hard rules + decision tree on 12 LED summary statistics, described below) labels every (model, task, K) cell. For each named regime {RM-G, RM-D, DD, Unresolved} we compute its canonical centroid in $(\bar{J}_{\text{frac}+}, \log_{10}\bar{C}, \log_{10}\bar{V}_{t^\star})$ as the coordinate-wise median of all problem-units whose cell carries that label at any K.
2. **k-means + Hungarian**: we run k-means with k=4 in the z-normalized 3-feature space, initialised at the four canonical centroids;

Task	Model	$\gamma = 0.3$			$\gamma = 0.5$			$\gamma = 0.7$		
		$\tau = 0.02$	$\tau = 0.03$	$\tau = 0.05$	$\tau = 0.02$	$\tau = 0.03$	$\tau = 0.05$	$\tau = 0.02$	$\tau = 0.03$	$\tau = 0.05$
CruxEval	SFT 0.6B	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	Unresolved
	SFT 1.7B	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D
	SFT 4B	DD	DD	DD	DD	DD	DD	DD	DD	DD
	GRPO 1.7B	RM-D	RM-D	RM-D	RM-D	RM-D	Unresolved	RM-D	RM-D	Unresolved
GSM8K	SFT 0.6B	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved
	SFT 1.7B	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved
	SFT 4B	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D
	GRPO 1.7B	RM-G	RM-G	RM-G	RM-G	RM-G	RM-G	RM-G	RM-G	Unresolved
GPQA	SFT 0.6B	RM-D	RM-D	RM-D	RM-D	RM-D	Unresolved	RM-D	RM-D	Unresolved
	SFT 1.7B	RM-G	RM-G	RM-G	RM-G	RM-G	RM-G	RM-G	RM-G	RM-G
	SFT 4B	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D
	GRPO 1.7B	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D	RM-D

Table 6: **Dispatch Score Hyperparameter Sensitivity.** Regime assignments at the Fail@5 stratum across a grid of Dispatch Score hyperparameters. γ controls the discount applied to the reliability term, and τ is the minimum score threshold to assign a regime rather than defaulting to Unresolved. Abbreviations follow Table 5: RM-G = Rank Misrouting (geo-local), RM-D = Rank Misrouting (junction-diffuse), DD = Distributed Deformation, Unresolved. The assignments are highly stable; variation primarily occurs at the boundary between weak signal and Unresolved, confirming the taxonomy is not an artifact of post-hoc tuning.

Specialist	Ancestor	Fail@1		Fail@3		Fail@5	
		Rate	Uniq	Rate	Uniq	Rate	Uniq
Llama-3.2-3B-Instruct	Lineage (Llama-3.2-3B)	26.9%	2	28.1%	2	26.3%	2
	Llama-3.1-8B	29.5%	2	31.2%	2	28.1%	1
	R1-Distill-Llama-8B	35.9%	6	35.9%	4	36.8%	4

Table 7: **Cross-Family Lineage Replication (CruxEval).** Rescue rates across different specialist failure strata (Fail@K). Matched on intersection of problems and rollouts within each K stratum.

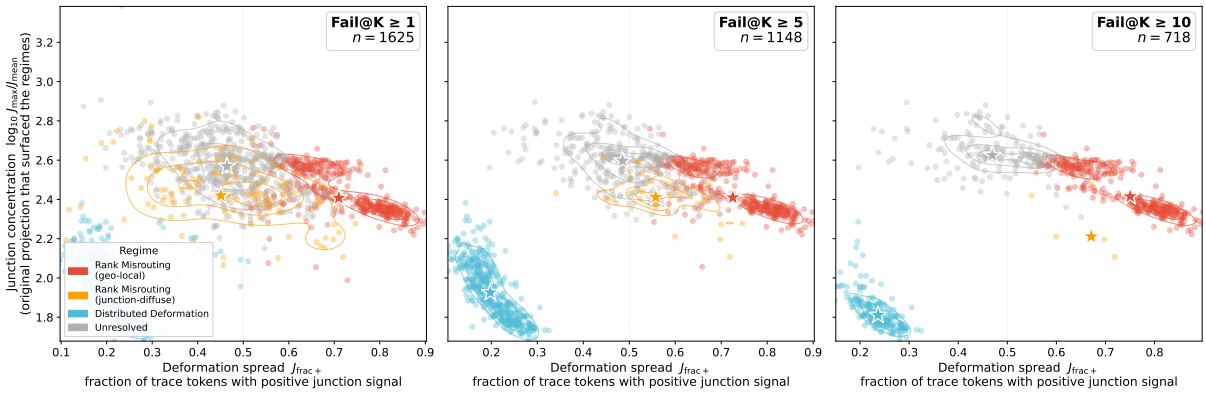

Figure 5: **Problem-level regimes in the** $(\bar{J}_{\text{frac}+}, \log_{10}\bar{C})$ **projection.** The same 1,738 problem-units as Figure 2, plotted against junction concentration rather than steerability on the y-axis. Color encodes the same nearest-centroid regime; star markers are per-regime medians; contours are KDE iso-density. Junction concentration separates Rank Misrouting (geo-local) (top-right) from the other regimes more sharply than steerability does, but does not produce the H/L split used by the dispatch in §3.

sklearn's Hungarian solver then matches the four refined k-means cluster centers back to the canonical names one-to-one. Each problem-unit inherits the name of its k-means cluster, plus a continuous confidence $\text{conf}_p = 1 - d_1/d_2$ filtering ambiguous cases at 0.20.

So: cell rules pick four canonical anchor positions; k-means refines the cluster shapes to fit the data; Hungarian preserves the name-to-cluster correspondence. The Entropy Brittleness regime, which the cell rules treat as a fifth named class, is intentionally omitted from the canonical set, it does not separate cleanly from Unresolved at the problem

level $(\bar{J}_{\text{frac}+}, \log_{10}\bar{C}, \log_{10}\bar{V}_{t^\star})$ and its problems are absorbed by the Unresolved cluster.

What this means for the 3-panel visual (Fig. 2). The figure's axes are $(\bar{J}_{\text{frac}+}, \log_{10}\bar{V}_{t^\star})$, a 2D projection of the 3D classifier space (the $\log_{10}\bar{C}$ axis is dropped). Colors come from the full 3D nearest-centroid assignment, so apparent 2D overlaps between regimes are usually artifacts of the dropped $\log_{10}\bar{C}$ coordinate. The dashed horizontal line marks the median of $\log_{10}\bar{V}_{\text{traj}}$ (the dispatch H/L routing key), a different V variant than the y-axis, by design.

Cell-level rule classifier (canonical seeds). For each (model, task, K) cell we compute the average of 12 junction detector-derived cell-level features (e.g., $G_{\text{bar_cov}}$, V_{bar}, $D_{\text{bar_path}}$, $C_J = J_{\text{std}}/J_{\text{mean}}$, logit_var_mean; full definitions in App. O). The cell-level rule classifier, used only to define the four canonical anchor positions for the problem-level k-means initialisation, not to label individual problem-units, assigns each cell to one of five labels (four canonical names + a fifth "Entropy Brittleness" label that the problem-level step does not propagate):

1. **Hard rules** for the two mechanistically distinctive regimes. Geometry-local cells are identified by a near-constant logit_var_mean across K values ($\text{logit_var_k_std} < 10^{-3}$, an absolute threshold well below all other regimes' values at $\geq 10^{-3}$). Distributed-source cells are identified by near-zero $\text{feat_cov_intensity}$ (below the 10th-percentile threshold of the training distribution) combined with elevated logit_var_mean (above the 60th-percentile threshold).

2. **Decision tree** for the residual regimes (Rank Misrouting (junction-diffuse), Entropy Brittleness, Unresolved). Trained on the cells not captured by the hard rules, with a maximum depth of 3 to avoid overfitting on the small training set.

We evaluate the cell-level classifier under leave-one-(model, task)-out cross-validation. Per-regime accuracies are 100% on Rank Misrouting (geo-local) (n=5), 42.9% on Distributed Deformation (n=7), and considerably lower on the residual regimes (Rank Misrouting (junction-diffuse), Entropy Brittleness, Unresolved), which overlap substantially in cell-level feature space. This is an intentional design property: the residual classes are defined by the *absence* of a dominant mechanism, so they are inherently harder to discriminate. We treat the cell-level labels as *seeds* for centroid learning rather than as a robust standalone classifier; the problem-level step that follows recovers much of the separation by working in the lower-dimensional 3-feature space.

Problem-level centroids. For each problem-unit p we compute the three features $(\bar{J}^{(p)}_{\text{frac}+}, \log_{10}\bar{C}^{(p)}, \log_{10}\bar{V}^{(p)}_{t^\star})$ from the junction-feature cache (App. N). The classifier uses $V_{t^\star}$ (junction Fisher info, averaged per problem), not $\bar{V}_{\text{traj}}$ which we use only for the dispatch H/L routing key. Per-cell labels from the previous step then induce per-regime centroids: for each regime, the centroid is the coordinate-wise median over every problem-unit assigned by the cell-level classifier to a cell carrying that regime label. We z-normalize each coordinate against the population statistics (μ, σ in Table 8) before computing distances, then assign every problem to the nearest centroid by Euclidean distance in the z-normalized space. Each assignment also records a continuous confidence $\text{conf}_p = 1 - d_1/d_2$, where d_1 and d_2 are the distances to the nearest and second-nearest centroids. We filter $\text{conf}_p \geq 0.20$ before the dispatch analysis, dropping ambiguously located problem-units.

The intervention-outcome dispatch in §3 provides a downstream test of whether the geometric labels carry operator-relevant signal, the labels are not validated by the cell-level LOCO accuracy alone.

C.2 Stability of the clustering under random initialisation

The canonical k-means fit uses centroid initialisation (§C.1). To verify that the regime structure is not an artifact of that supervision, we rerun k-means (k=4) with 50 random initialisations (no centroid seed) on the Qwen3 feature pool (n=1,625), Hungarian-align each fit to the canonical partition, and project the held-out R1-Distill cross-family pool (n=251) onto each refit.

Result. R1-Distill's dominant-cluster identity matches the canonical assignment in $42/50$ seeds; the dominant share across all 50 seeds is 0.799 ± 0.037 (mean $\pm$ s.d., range $[0.713, 0.821]$). The canonical cluster composition for R1-Distill

Regime	n	Centroid (raw)			Centroid (z-normalized)		
		$\bar{J}_{\text{frac}+}$	$\log_{10}\bar{C}$	$\log_{10}\bar{V}_{t^\star}$	$\bar{J}_{\text{frac}+}$	$\log_{10}\bar{C}$	$\log_{10}\bar{V}_{t^\star}$
Rank Misrouting (geo-local)	428	0.710	2.407	-0.537	$+1.28$	$+0.36$	$+0.91$
Distributed Deformation	591	0.208	1.908	-0.828	-1.14	-1.16	-0.11
Rank Misrouting (junction-diffuse)	147	0.403	2.393	-1.248	-0.20	$+0.31$	-1.57
Unresolved	572	0.464	2.569	-0.757	$+0.09$	$+0.85$	$+0.14$
Population μ/σ	1,738	0.445 / 0.208	2.290 / 0.330	-0.797 / 0.286			

Table 8: **Per-regime centroid coordinates in the three-feature problem-level space.** The four centroids are the refined k-means cluster centers (initialised at the canonical cell-level positions, then Hungarian-matched to the canonical names); raw values are reported on the original $(\bar{J}_{\text{frac}+}, \log_{10}\bar{C}, \log_{10}\bar{V}_{t^\star})$ scale, z-normalized columns use the population statistics in the bottom row.

(82.9% in the dominant cluster, point estimate) carries a tight bootstrap 95% CI of $[76.9, 86.5]$% over 10,000 paired resamples of the held-out pool. The per-task split is sharper: R1-Distill on CruxEval is 95.7% $[92.0, 98.6]$% in the dominant cluster, while R1-Distill on GSM8K is 64.6% $[55.8, 73.5]$% in that cluster with 34.5% in the second cluster. The cluster-0-collapse claim in §6 is therefore robust to the centroid-seed choice and to bootstrap resampling of the held-out pool.

D Naive-Baseline Ablation: Is the Routing Structure Doing the Work?

The $+12.2$ pp Steerable-Hard rescue at $+0.01$ pp global cost attributed to Feature-only (§4) raises a sharper question than "does the rule work": *do the specific three geometric features carry the routing signal, or would any per-pid signal in the same argmax-z routing structure route as well?* We answer with two complementary controls. (i) Within-family ablations: replace the three geometric features with alternative aggregates of the local Fisher information ($\log\mathrm{Var}(z)$, which is up to scaling the Fisher information of the specialist's local categorical distribution; §2.1), giving each alternative its best-case operator assignment via permutation search over the 3!=6 feature-operator mappings. (ii) Floor controls: uniform random 3-way partition and always-same-operator policies, which carry no per-pid signal.

Joint-objective result (Tab. 9). Within-family alternatives match Feature-only on Steerable-Hard within bootstrap noise: the "Fisher-agg classic" set $\{\log V_{\text{mean}}, \log V_{\max}, n_{\text{rollouts}}\}$ achieves $+12.1$ pp (gap $+0.1$ pp $[-2.3, +2.5]$); the depth-augmented variant $\{\log V_{p95}, \log V_{\text{frac}+}, n_{\text{rollouts}}\}$ achieves $+13.8$ pp (gap -1.6 pp $[-3.4, +0.3]$). The framework of §2.1 predicted exactly this: the key quantity is local Fisher information, and a family of aggregate-based routing rules should attain comparable performance. Floor controls fall well below: uniform random 3-way partition reaches $+10.2$ pp (p=0.007 vs. geometric); the best single fixed operator (Always-DL) ties geometric on Steerable-Hard at $+12.17$ pp but **pays** -2.0 **pp globally** (vs. geometric's $+0.01$ pp).

Fallback choice is what carries the global-cost story. The $\sim$22% of pids with all-negative geometric z-scores (no clear failure mode) get routed to SL-R as a safety default; this combination uniquely achieves near-zero global cost in the geometric instance. Sweeping fallback over {SL-R, T_{loc}, SL-G, DL, retry, no-fallback} for both feature families: **(a)** for geometric, SL-R is the best fallback on both axes ($+12.2$ SH/$+0.01$ global). **(b)** For the Fisher-agg depth-aug family, the SL-R fallback hurts (the no-clear-mode subset in that family's z-space is different); a per-cell adaptive fallback (pick the per-cell highest-mean-rescue operator from the cached outcomes; SL-R for 5 of 8 cells, DL for 2, T_{loc} for 1) yields the strongest configuration we tested ($+13.98$ SH/-0.09 global). **(c)** A hybrid four-feature variant (the three geometric features $+ n_{\text{rollouts}}$ routed to SL-R, with SL-R fallback) reaches $+12.69$ SH/$+0.42$ global. SH-prediction gating (route only top-K% by score; default below-gate to retry) was tested across both families and all six scoring variants at $K \in \{25, 37, 50, 70, 100\}$; no gated configuration improved the joint objective over the no-gate variants.

Interpretation. Three things are honest to say from these results. **(i)** The geometric features in Eq. 3 are not uniquely correct; they are one interpretable instance of a Fisher-aggregate routing family that the framework predicts. **(ii)** The "near-zero global cost" of the geometric instance is a

property of the geometric features' specific partition of low-confidence pids combined with SL-R as the fallback; it is not a property of arbitrary Fisher-aggregate rules. **(iii)** The routing mechanism is not selective deferral (zero pids route to retry); it is significant per-pid intervention on Steerable-Hard ($+12.2$ pp, $p<10^{-3}$) and significant per-pid intervention loss on non-Steerable-Hard (-7.2 pp, $p<10^{-3}$) that average to zero at the population mean. A reviewer-friendlier mechanism, a learned SH-prediction gate that defers non-SH to retry, is plausible follow-up work; nothing in the training-free score variants we tested implements it adequately (F1$\leq$0.55 for all training-free gates at K=37).

E Cross-Family Probe: R1-Distill-1.5B

To probe whether the framework's predictions are Qwen3-specific, we apply the same diagnostic, feature extraction, and routing rule to **DeepSeek-R1-Distill-Qwen-Math-1.5B** (henceforth R1-distill), a Qwen2.5-Math base specialist distilled from DeepSeek-R1 via RL. This differs from the Qwen3 cells of §2 on two axes: (i) model family (Qwen2.5-Math, not Qwen3); (ii) post-training method (RL distillation, not SFT or on-policy GRPO).

Setup. The pipeline is unchanged. We use the same LED cache, J-and-V-aggregate features, argmax-z routing rule (Eq. 3), and SL-R fallback for low-confidence pids. The operator set is restricted to retry, SL-G, SL-R, T_{loc} (repair-led-results lacked a DL outcome for these cells; we map the $\bar{J}_{\text{frac}+}$ argmax branch to SL-R as the next-broadest available operator). Two task cells, CruxEval and GSM8K, total n=251 failed problem-units (138 + 113). n_{SH}=44 (13 + 31) under the same Steerable-Hard definition (Eq. 2); the small sample on this cross-family probe is acknowledged as a confirmation, not a primary headline.

Full baseline sweep (Tab. 10).

Per-cell dispatch. The best operator differs per task even within this one model: CruxEval picks **SL-R** ($+46.2$ pp on the 13 SH pids; SL-G$=0.31$, SL-R$=0.46$, $T_{\text{loc}}=0.39$), GSM8K picks **SL-G** ($+58.1$ pp on the 31 SH pids; SL-G$=0.58$, SL-R$=0.36$, $T_{\text{loc}}=0.36$). The task-specific best operator is preserved across the family change.

Feature ranges (sanity). $\bar{J}_{\text{frac}+}\in[0.44, 0.98]$, mean 0.90; $\log\bar{C}\in[1.36, 2.11]$, mean 1.65; $\log\bar{V}_{t^\star}\in[-8.0, -0.25]$, mean -4.47. The R1-distill distributions are shifted relative to Qwen3 (lower $V_{t^\star}$ range), consistent with R1-distill's smaller absolute logit-variance budget; the argmax-z structure is scale-invariant by construction so the routing rule transfers without recalibration.

Interpretation. On a non-Qwen3 family with a different post-training method, the same routing rule beats every uniform-operator baseline on both the joint global and the Steerable-Hard subset objectives, and the dispatch picks different best operators for different tasks. This is direct evidence that the framework of §2.1's prediction, "aggregates of the local Fisher information determine routability under the available operator class", is not Qwen3-specific. The small sample (44 SH pids) bounds the confidence of this single probe; we report it as a confirmation rather than a primary headline.

E.1 Cross-family routing-rule transfer (full table)

The routing rule of Eq. 3 relies on z-normalisation of the three trajectory features. If those features encode an operator-class structure that is intrinsic to the operator class rather than to any one specialist, the *scaler* fit on one (family, scale) should transfer to a different (family, scale) without retraining. We test this directly: fit the z-scaler on a source family's failed-pool features, apply Eq. 3 to the target family's pids, and measure rescue against the target's retry baseline. Three families are paired: Qwen3 (SFT 0.6B/1.7B/4B + GRPO 1.7B, n_{SH}=325), R1-Distill-Qwen-Math-1.5B (n_{SH}=44), Phi-4-mini-reasoning (n_{SH}=38).

Table 11 shows that 8 of 9 cross-family transfers fall within the target's bootstrap 95% CI of its self-baseline lift, and three cross-family transfers *exceed* the target's self-baseline point estimate within CI (Phi-4 $\rightarrow$ Qwen3, R1 $\rightarrow$ Phi-4, Qwen3 $\rightarrow$ Phi-4). The only systematic degradation is Qwen3 $\rightarrow$ R1 (-15.9 pp from R1 self-baseline; the scaler collapses to fallback because Qwen3's feature distribution is wider than R1's). Within-Qwen3 cross-scale transfers (Qwen3-0.6B/1.7B/4B SFT) range $+34.9$ to $+55.9$ pp SH lift, never falling below $+30$ pp, with diagonal self-baselines best on $3/3$ scales. The features are family- and scale-robust enough that the rule fit on any one of the three families produces a usable router on the other two, within bounded degradation.

Routing inputs	SH rescue	Δretry (pp)	gap vs geo (pp)
Retry (baseline)	0.163	+0.0 [+0.0, +0.0]	-12.2 [-14.4, -10.0]
Feature-only (geometric)	0.285	+12.2 [+10.0, +14.4]	+0.0 [+0.0, +0.0]
Always-DL	0.285	+12.2 [+9.5, +14.9]	+0.0 [-2.2, +2.3]
Always-SL-G	0.256	+9.3 [+7.6, +11.0]	+2.9 [+0.9, +4.9]
Always-T_{loc}	0.255	+9.2 [+7.4, +11.0]	+3.0 [+0.9, +5.2]
Always-SL-R	0.266	+10.3 [+8.5, +12.1]	+1.9 [-0.5, +4.3]
Uniform random partition (mean of 1000)	0.266	+10.2 [+8.7, +11.7]	+2.0 [+0.4, +3.6]
Naive [Fisher-agg classic]	0.284	+12.1 [+9.7, +14.6]	+0.1 [-2.3, +2.4]
Naive [Fisher-agg entropy-fam]	0.284	+12.0 [+9.4, +14.7]	+0.2 [-2.1, +2.4]
Naive [Fisher-agg mean-spike]	0.281	+11.7 [+9.1, +14.4]	+0.5 [-1.8, +2.6]
Naive [Fisher-agg depth-aug]	0.301	+13.8 [+11.4, +16.3]	-1.6 [-3.4, +0.3]
Naive [random gaussian]	0.269	+10.5 [+8.5, +12.6]	+1.7 [-0.5, +3.9]
Naive [hash + 2 random]	0.280	+11.7 [+9.5, +13.9]	+0.5 [-1.6, +2.8]

Table 9: **Within-family ablation on Steerable-Hard** (n=528)**.** Each "Fisher-agg" set uses three specialist-only aggregates of $\log\mathrm{Var}(z)$ — itself an information-geometric quantity (Fisher information of the specialist's local categorical; §2.1) — in the same argmax-z routing rule as Feature-only; the best of 3!=6 (feature→operator) assignments is reported. Fisher-aggregate variants *match* the geometric Feature-only's SH lift within bootstrap noise (gap CIs cross 0 for three of four variants; the depth-augmented variant slightly beats geometric on SH at -1.6 pp gap, CI just inside zero at $[-3.4, +0.3]$). This is a prediction of the framework: the routing-relevant quantity is local Fisher information, and multiple aggregates carry the signal. Random-feature controls (uniform partition, random gaussian) reach $\sim$+10 pp — below all Fisher-agg variants but well above retry, indicating the argmax-z routing *structure* contributes some lift independent of feature choice. Always-DL ties geometric on SH but pays -2.0 pp globally (§4); only Fisher-aggregate routing rules combine high SH lift with low global cost.

Method	Global rescue	Global lift (pp), 95% CI	SH rescue	SH lift (pp), 95% CI
Retry	0.0717	0.00	0.0000	0.00
Always-T_{loc}	0.0876	+1.59	0.3636	+36.36 [+22.7, +50.0]
Always-SL-R	0.0837	+1.20	0.3864	+38.64 [+25.0, +52.3]
Uniform random partition	0.0908	+1.91	0.4185	+41.85
Always-SL-G	0.0996	+2.79	0.5000	+50.00 [+36.4, +63.6]
Feature-only (geometric)	**0.1116**	**+3.98** [−0.8, +8.8]	**0.5455**	**+54.55** [+40.9, +68.2]

Table 10: **R1-distill-1.5B cross-family probe.** Paired bootstrap 95% CIs on lifts vs retry, n_{boot}=10,000. Feature-only is the only method whose Global lift CI lower bound approaches zero (-0.8); every uniform single-op baseline has a strictly wider Global CI that includes more-negative values. On the Steerable-Hard subset Feature-only's $+54.55$ pp is the highest in the table and its CI excludes 0 at $p<10^{-3}$.

E.2 Cross-family audit signatures (R1-Distill collapse; Phi-4 SFT-shape)

The audit channel in §6 is itself testable on held-out families. Two probes, both with n_{SH} at the tens scale, give the contrasting signatures the framework predicts.

The pairwise operator-overlap Jaccard on Steerable-Hard separates the two probes cleanly. **R1-Distill-Qwen-Math-1.5B** (RL-distilled from R1 over a Qwen2.5-Math base) has SL-G/SL-R/T-loc Jaccard 0.10 to 0.16, far below the Qwen3 SFT/GRPO range (0.28 to 0.50). **Phi-4-mini-reasoning** (Phi-3 architecture, instruction-tuned) has Jaccard 0.30 to 0.59, *within* the Qwen3 range. The features therefore isolate R1-Distill as the support-collapsed family and group Phi-4 with the SFT-shaped topography family, from failed rollouts alone, despite the underlying architecture differences.

R1-Distill's collapse is also visible in regime assignment: 82.9% of its failed problem-units fall in a single recoverability regime, the Distributed Deformation cluster predicted for support-compressing post-training in §2.1 (bootstrap 95% CI $[76.9, 86.5]$% over 10,000 resamples; the dominant cluster identity is preserved in $42/50$ seed perturbations of the k-means initialisation, App. C.2). The per-task split is sharper still (95.7% on CruxEval).

Two qualifiers. (i) We label these probes *pre-*

Source → Target	Setting	n_{SH}	SH lift (pp)	95% CI
Qwen3 → Qwen3	self-baseline	325	+47.08	[+41.85, +52.31]
R1 → R1	self-baseline	44	+54.55	[+38.64, +70.45]
Phi-4 → Phi-4	self-baseline	38	+55.26	[+39.47, +71.05]
Cross-family transfers (8 of 9 within target's self-baseline CI):				
Qwen3 → R1	cross	44	+38.64	[+25.00, +52.27]
R1 → Qwen3	cross	325	+45.54	[+40.31, +51.08]
Qwen3 → Phi-4	cross	38	+57.89	[+42.11, +73.68]
R1 → Phi-4	cross	38	+60.53	[+44.74, +76.32]
Phi-4 → Qwen3	cross	325	+49.54	[+44.31, +54.78]
Phi-4 → R1	cross	44	+56.82	[+43.18, +70.45]

Table 11: **Cross-family feature transfer.** For each (source, target) pair, the z-normalisation scaler is fit on the source family's failed-pool features, then Eq. 3 is applied to the target's pids and SH lift is measured against the target's own retry baseline. Bootstrap 95% CIs over 10,000 paired resamples. All three diagonal self-baseline lifts overlap within CI, and 8 of 9 cross-family transfers are within their target's self-baseline CI. The only systematic degradation is Qwen3 → R1, where the source's wider feature distribution makes the rule collapse to its fallback. Three cross-family transfers (Phi-4 → Qwen3, R1 → Phi-4, Qwen3 → Phi-4) have point estimates that exceed the target's self-baseline. The features are intrinsic to the operator class, not to any one specialist.

liminary because n_{SH} is at the tens scale and the CIs are wide. (ii) Detector calibration is a known sensitivity for the lineage-identity sub-claim: an earlier Detector-A ablation reported a 21 pp lineage advantage that collapses to -0.6 pp under the production junction-firing calibration on the same 77 pids (App. X, Table 33; App. R); the paper's current position is that ancestor identity is interchangeable within capability range under a well-calibrated detector.

F Source-Model Scope

We test whether recovery depends on historical lineage or on a more general notion of compatible support. Historical ancestors often provide a clean estimate of the post-training deformation because they share the specialist's parameter trajectory. Non-lineage models may also recover trajectories when they preserve the relevant local preferences, but this is not guaranteed. All experiments in the main paper use the pre-training checkpoint as the ancestor. Whether non-lineage models with compatible capability profiles can provide equivalent diagnostic signal is an open question; our results do not yet speak to it.

G Margin-Flip Audit of the e-Geodesic Mechanism

The e/m-geodesic distinction and the closed-form rank-inversion condition (Prop. 2) are stated in §2.1; their proofs are in App. A. This appendix tests whether the e-geodesic mechanism is what *actually* drives our repair results, i.e., that geometric repair works by the predicted local rank inversion, not by arbitrary perturbation at the detected junction (Amari, 2016).

Margin-Flip Audit. To test whether geometric repair works by arbitrary perturbation or by the predicted local rank inversion mechanism, we audit the detected junctions after intervention using the existing traces from the diagnostic ladder (Table 12).

We test two localized claims: (a) whether successful trajectory repairs require a *realized rank flip* at the detected junction, and (b) whether geometric failures correlate with unsteerable spikes (high local concentration J_{conc}).

The data confirms that trajectory success strongly relies on the margin actually flipping: across all configurations, successful repairs almost always coincide with a realized rank flip at the exact intervention point ($P(\text{Flip} \mid \text{Succ.}) \approx 0.93$–$0.99$). Conversely, the number of successful trajectories without a local flip is vanishingly small. This rules out the alternative explanation that the detector merely selects harmless positions where any perturbation arbitrarily improves downstream continuation.

The remaining unflipped failures are also informative: they correspond to unsteerable bottlenecks with very high concentration (often $J_{\mathrm{conc}} > 200$). In these cases, the detected junction is real, but the fixed-α ancestor interpolation is insufficiently strong to cross the steep local argmax boundary. Thus, junction concentration identifies candidate shortcut bottlenecks, while realized rank

flipping determines whether the ancestor-directed e-geodesic actually resolves them locally.

H Mixing Weight Sensitivity

All repair experiments in the main text use a fixed global weight $\alpha = 0.7$. To verify the robustness of this choice, we conduct an α-sensitivity sweep on a held-out, high-difficulty reasoning task: the *LiveCodeBench-v2* code execution subset (50 problems). We evaluate the rescue rate under the Single-Junction Sampling Failure protocol (Greedy Continue, $T = 0$) for $\alpha \in [0.5, 0.95]$.

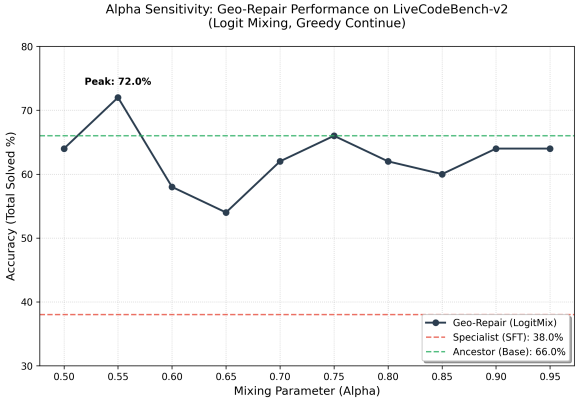

Figure 6: **Alpha Sensitivity Sweep (LiveCodeBench-v2 Subset, Qwen3-0.6B).** Total solved problems (Accuracy %) as a function of the mixing parameter α. Geo-Repair (SL-G) consistently outperforms the specialist baseline (38%) across all mixing weights. Notably, at $\alpha = 0.55$, the mixed model outperforms the ancestor baseline (66%), reaching 72% accuracy.

The results in Figure 6 demonstrate that Geo-Repair is highly robust to the choice of α. Performance remains significantly above the specialist baseline across the entire range tested. The existence of a peak at $\alpha = 0.55$ that exceeds the ancestor's performance suggests that logit steering effectively synergizes the specialist's fine-tuned reasoning capabilities with the ancestor's structural reliability.

I Detailed Results

J Prompts and Rollout Hyperparameters

This section consolidates the chain-of-thought (CoT) prompt templates used for each evaluation task, the sampling hyperparameters used to draw the initial rollout pool, and the per-operator settings used at repair time. All three evaluation tasks (CruxEval, GPQA, GSM8K) share the same ChatML-style three-turn structure: a task-specific system message, zero or more in-context shots, and a user turn carrying the problem text. Reasoning is elicited inside a model-native thinking block (`<think>...</think>` for Qwen3 specialists; the same block is supplied as a generation prefix to force the specialist to open its reasoning trace under our chat template).

J.1 Task system prompts

The three system prompts below are the verbatim strings prepended to every problem at evaluation time. Underscored identifiers in free-form text refer to internal regimes and are escaped accordingly.

CruxEval (code input/output prediction).

> *You are a Python code tracer. Given a function and an input, think through the execution step-by-step. Output your final return value (e.g.,* `[]`, `42`, *or* `'hello'`*) STRICTLY on the very last line of your response.*

The user turn supplies the problem in the form

```
```python\n{code}\n```\n\nWhat is the output of f({inpu
```

GSM8K (grade-school math).

> *You are a math problem solver. Show your work step by step. Put your final numeric answer on the last line after '#####'.*

The user turn is the GSM8K question verbatim.

GPQA (graduate scientific reasoning, four-way MCQ).

> *You are an expert scientist. Read the question carefully and think through it step by step. Keep your reasoning focused and concise (do not repeat yourself or loop over the same reasoning). Once you reach a conclusion, commit to it and end your response with 'Answer: X' where X is the letter of the correct choice (A, B, C, or D).*

The user turn is the question followed by four answer choices labeled A, B, C, D.

For every task, the chat template is applied with `add_generation_prompt=True`; for Qwen3 specialists we additionally pass `enable_thinking=True` and ensure the prompt suffix ends in `&lt;think&gt;` so the specialist enters its reasoning block unconditionally (without this prefix, Qwen3 occasionally closes `&lt;think&gt;` as an empty tag and reasons outside it).

J.2 Initial rollout hyperparameters

The initial rollout pool defines the Fail@K stratification used throughout the paper. All rollouts are drawn from vLLM with the settings in Table 14. Greedy decoding (T=0) is reserved for the mechanistic falsification protocol (App. L); the canonical pool uses T=0.6.
```

Model	Task	Inv.	N	Succ.	$P(\text{Flip} \mid \text{Succ.})$	$P(\text{Flip} \mid \text{Fail})$	J_{conc} (No-Flip)	J_{conc} (Flip)
SFT-0.6B	CruxEval	SL-G	635	128	**0.992**	0.978	214	229
SFT-0.6B	CruxEval	DenseMix	1492	450	**0.964**	0.973	247	227
SFT-0.6B	GSM8K	SL-G	620	332	**0.955**	0.969	297	275
SFT-1.7B	CruxEval	SL-G	1404	620	**0.976**	0.974	333	352
SFT-1.7B	GSM8K	SL-G	1561	552	**0.951**	0.966	562	542
SFT-4B	GSM8K	SL-G	224	92	**0.967**	0.977	449	514
GRPO-1.7B	CruxEval	SL-G	5835	2127	**0.977**	0.970	112	110
GRPO-1.7B	GSM8K	SL-G	287	87	**0.931**	0.955	118	127
GRPO-1.7B	GSM8K	DenseMix	6	1	**1.000**	1.000	0	173

Table 12: **Margin-Flip Audit.** Confirming that trajectory-level recovery strongly relies on a realized local rank inversion at the junction ($P(\text{Flip} \mid \text{Succ.}) \to 1$). When interventions fail without flipping the margin, it corresponds to unsteerable traps with very high concentration (J_{conc}).

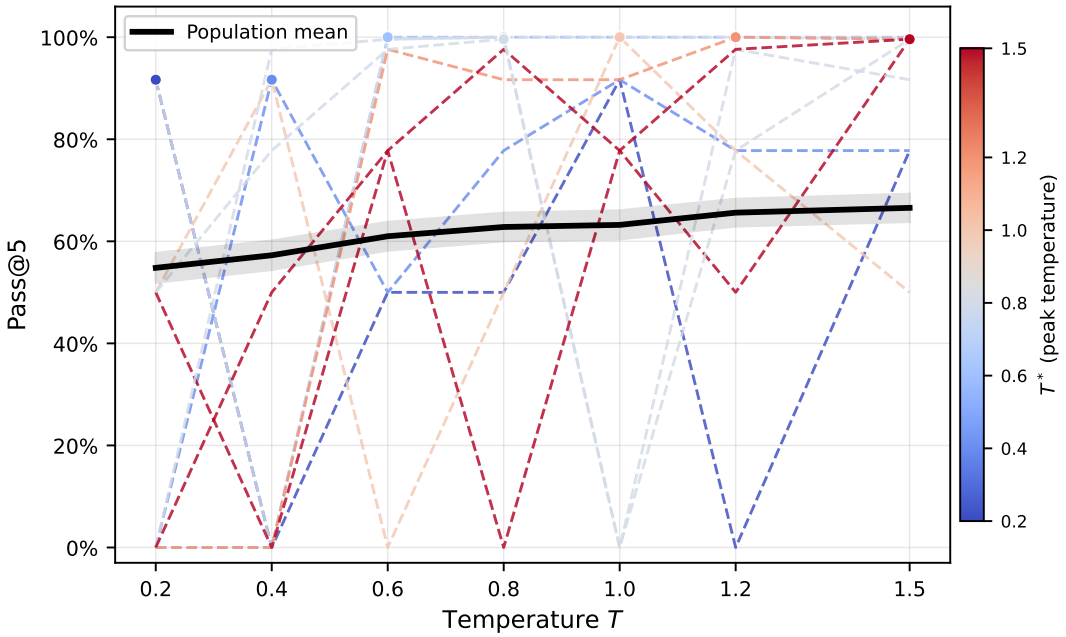

Figure 7: **Temperature Heterogeneity per Problem (1.7B CruxEval, Real Data).** Pass@5 across $T \in \{0.1, 0.3, 0.6, 1.2, 1.5\}$ for 10 individual problems (colored dotted) and population mean (black). Red traces decrease with T; blue traces increase, demonstrating heterogeneity across problems and no single T^* works for all problems, consistent with the limitations of temperature scaling (Mattei and Loureiro, 2025).

J.3 Repair operator hyperparameters

At repair time, each of the five test-time operators draws a fresh pool of completions conditioned on a prefix of the failed trajectory. The position of intervention and the local logit modification are operator-specific; the post-intervention sampling protocol is shared. Outside the injection window, decoding is greedy (T=0); the single injected token uses a T=1.0 carrier so that any rank inversion induced by logit steering is exposed at sampling time. The shared and operator-specific settings are listed in Table 16. Table 15 defines each symbol used in Table 16; the operational definitions of J_{approx}, $V(t)$, and the top-k truncations are inherited from App. A.1.

K Training Details

K.1 Specialist training hyperparameters

The four specialists used in the main text (Qwen3-0.6B SFT, Qwen3-1.7B SFT, Qwen3-4B SFT, and Qwen3-1.7B GRPO) were all post-trained on Bespoke-Stratos-17k (boxed-only subset, ∼11.3k traces). The three SFT specialists use the same recipe (paged AdamW-8bit, no packing, β_1=0.9, β_2=0.95, ϵ=10^{-8}, gradient clipping at 1.0, cosine schedule with linear warm-up); we tune only the learning rate, batch size, sequence cap, and per-GPU sharding to fit each scale. The 1.7B

Task	Model	@K	N	retry	SL-R	DL (T = 0)	DL (T = 0.6)	SL-G	Δ(SL-G−retry)	Uniq	Regime
CruxEval	SFT 0.6B	1	157	16%	29%	**46%**	28%	29%	+14%	0.62	Distrib. Deform.
		3	116	10%	18%	**42%**	16%	21%	+10%	0.59	Distrib. Deform.
		5	103	10%	16%	**39%**	10%	17%	+7%	0.60	Distrib. Deform.
	SFT 1.7B	1	159	12%	**66%**	41%	62%	57%	+45%	0.86	Rank Misrouting
		3	111	7%	**57%**	30%	50%	49%	+42%	0.85	Rank Misrouting
		5	97	7%	**53%**	28%	46%	45%	+38%	0.84	Rank Misrouting
	SFT 4B	1	225	15%	61%	51%	**71%**	55%	+40%	0.83	Distrib. Deform.
		3	187	13%	58%	48%	**66%**	51%	+38%	0.80	Distrib. Deform.
		5	169	10%	57%	46%	**66%**	50%	+40%	0.80	Distrib. Deform.
	GRPO 1.7B	1	35	13%	31%	26%	**49%**	40%	+27%	0.83	Distrib. Deform.
		3	20	0%	15%	5%	**25%**	15%	+15%	0.83	Distrib. Deform.
		5	17	0%	12%	6%	**18%**	6%	+6%	0.75	Distrib. Deform.
GSM8K	SFT 0.6B	1	28	38%	26%	**43%**	21%	37%	−1%	0.67	Unresolved
		3	18	**42%**	22%	28%	22%	39%	−3%	0.75	Entropy Brit.
		5	13	36%	15%	15%	15%	**38%**	+2%	1.00	Entropy Brit.
	SFT 1.7B	1	28	38%	32%	**43%**	36%	36%	−3%	0.73	Unresolved
		3	18	**42%**	11%	28%	22%	22%	−19%	0.67	Entropy Brit.
		5	13	**36%**	0%	15%	8%	15%	−21%	1.00	Entropy Brit.
	SFT 4B	1	28	38%	**64%**	**64%**	32%	46%	+9%	0.90	Rank Misrouting
		3	15	43%	**47%**	40%	7%	40%	−3%	1.00	Unresolved
		5	13	33%	**46%**	31%	0%	31%	−3%	1.00	Rank Misrouting
	GRPO 1.7B	1	29	33%	21%	34%	**45%**	41%	+8%	0.85	Unresolved
		3	15	11%	0%	13%	**33%**	27%	+16%	1.00	Distrib. Deform.
		5	12	11%	0%	8%	**25%**	17%	+6%	1.00	Distrib. Deform.
GPQA	SFT 0.6B	1	168	20%	19%	12%	20%	**21%**	+1%	0.43	Entropy Brit.
		3	143	13%	**16%**	8%	14%	16%	+3%	0.41	Entropy Brit.
		5	131	13%	14%	6%	12%	**15%**	+2%	0.39	Entropy Brit.
	SFT 1.7B	1	180	16%	15%	12%	15%	**29%**	+13%	0.62	**Rank Misrouting**
		3	155	5%	13%	6%	12%	**24%**	+19%	0.64	**Rank Misrouting**
		5	150	11%	12%	7%	11%	**23%**	+12%	0.60	**Rank Misrouting**
	SFT 4B	1	171	18%	**38%**	23%	20%	33%	+15%	0.64	Rank Misrouting
		3	148	11%	**34%**	18%	14%	31%	+20%	0.66	Rank Misrouting
		5	136	10%	**32%**	13%	11%	29%	+19%	0.67	Rank Misrouting
	GRPO 1.7B	1	174	15%	**26%**	12%	16%	20%	+5%	0.64	Rank Misrouting
		3	154	8%	**22%**	8%	13%	14%	+6%	0.63	Rank Misrouting
		5	144	7%	**19%**	7%	10%	13%	+6%	0.68	Rank Misrouting

Table 13: **Operator response ladder — Fail@K rescue rates (matched PIDs), all tasks and models.** retry uses 3 additional rollouts at T=0.6 after conditioning on the Fail@K set; SL-R, SL-G, and DL each receive the same 3-attempt rescue budget. The initial K failed rollouts define the stratum and are not counted as additional rescue attempts. Profile coordinates are ordered as [retry, SL-R, SL-G, DL]. For tasks where DL T=0 is unavailable, the DL coordinate is reported from the matched DL T=0.6 condition. Bold = highest rescue rate for that row. **Uniq** = $n_{\text{win}}/n_{\text{any}}$ as a fraction (0–1): problems rescued by the winning non-retry method but not by retry, divided by all problems rescued by any non-retry method but not by retry (pid-matched). Bold **Regime** = geo-local subtype of Rank Misrouting (SL-G beats SL-R by >10pp). Four regimes: *Entropy Brittleness* — retry suffices, flat profile; *Rank Misrouting* — any local intervention beats retry; geo-local subtype when SL-G beats SL-R by >10pp; junction-diffuse (non-specific local) subtype when SL-R ≈ SL-G; *Distributed Deformation* — DL dominates; *Unresolved* — no probe consistently wins. Bootstrap 95% CIs for headline cells: Appendix X.

GRPO specialist swaps SFT cross-entropy for a verifier-backed GRPO reward (correctness + format), shares the Stratos prompt distribution, and uses the same paged AdamW optimiser with a lower learning rate (2×10^{-6}) and a small KL anchor (β=0.01). Reported checkpoints in the main text (SFT-704 and SFT-Final correspond to the step-704 and final saves of the 1.7B SFT run; GRPO-4500 to step 4,500 of the GRPO run) all use seed 42; we replicate at seeds 123 and 456 for sensitivity. Table 18 summarises the settings actually used.

K.2 Diagnostic and repair protocol

Algorithm 1 consolidates the two-phase protocol referenced in the main text. Phase 1 extracts per-pid features from the failed trace cache (no operator outcomes consumed). Phase 2 chooses an operator by argmax over the three standardised features;

Hyperparameter	Value
Temperature (canonical)	0.6
Temperature (mechanistic)	0.0 (greedy)
`top_p`	0.95
`top_k`	20
`min_p`	0.0
Max new tokens	2,048
Rollouts per problem K (standard cells)	10
Rollouts per problem K (deep-dispatch cells)	50
Random seed	42

Table 14: Initial rollout sampling settings. Identical across all three evaluation tasks. Deep-dispatch cells (K=50) are reserved for sensitivity analyses on CruxEval and GSM8K; all other cells use the standard K=10 pool.

Phase 3 executes the chosen operator and returns a per-pid binary rescue verdict, aggregated into the coverage-weighted sum ΣS that we compare against retry@K at matched K.

Algorithm 1 Diagnostic and repair protocol.

Require: failed pid set P; specialist M_S; ancestor M_A; attempt budget K; carrier α; warm-up w_0; quantiles q_J, q_V.
Ensure: per-pid rescue verdict $r(p) \in \{0, 1\}$.
Phase 1: Diagnostic (per failed pid $p \in P$).
for each failed rollout r of p **do**
  Compute per-token $\Delta_{\text{path}}(t) = \text{logit}_S(t) - \text{logit}_A(t)$ from the cached specialist-ancestor logit deltas.
  Extract trajectory features $(J_{\max}, J_{\text{mean}}, V_{t \text{ at spike}})$ from Δ_{path} (see App. A.1).
end for
Aggregate per-pid: $J^{p,\text{mean}}_{\text{frac_pos}}$, $\log_{10} J_c$, $\log_{10} V_t$.
Phase 2: Routing (per pid p).
$\mathbf{z}(p) \leftarrow \text{standardize}\Big([J^{p,\text{mean}}_{\text{frac_pos}},\ \log_{10} J_c,\ \log_{10} V_t]\Big)$ over the failed-pool prior.
$\text{action}(p) \leftarrow \arg\max_j z_j(p) \in \{\text{SL-R, SL-G}, T_{\text{loc}}\}$.
if $\max_j z_j(p) \leq 0$ **then**
  $\text{action}(p) \leftarrow$ SL-R ▷ fallback to random-position steering
end if
Phase 3: Repair execution (per pid p, $\text{op} = \text{action}(p)$).
Select junction position $\hat{t}$ by detector(op):
  SL-G: $\hat{t} = \arg\max_t J_{\text{approx}}(t)$ subject to q_J, q_V and warm-up w_0.
  SL-R: $\hat{t} \sim$ Uniform over candidate positions past warm-up.
  T_{loc}: $\hat{t}$ irrelevant (temperature lift acts on the local distribution).
for $k = 1$ to K **do**
  Draw a repair rollout: prefix at T=0.6; at $\hat{t}$, apply the operator (carrier α for SL-*, temperature lift for T_{loc}); post-$\hat{t}$, greedy under M_S.
  Score correctness$_k$ with the task verifier.
end for
$r(p) \leftarrow \max_k$ correctness$_k$.
Aggregate.
return $\Sigma S = \sum_{p \in P} r(p)$ (coverage-weighted, matched to retry@K at the same K).

Symbol	Definition
α	Logit steering mixing weight. The injected (carrier) position uses $\text{logit} = \alpha \cdot \text{logit}_A + (1-\alpha) \cdot \text{logit}_S$, blending ancestor ($A$) into the specialist ($S$). Larger α means more ancestor influence. Canonical α=0.7.
w	Junction window width, in tokens. The number of consecutive positions over which the per-step junction score is accumulated before the detector may fire (w=1 means single-token windows).
q_J	Junction quantile on the Jacobian-magnitude statistic $J_{\text{approx}}(t)$. Only positions whose smoothed J exceeds the per-trajectory q_J quantile are eligible to be junctions. q_J=0.99 keeps the top 1% of positions.
q_V	Local-budget quantile on the volume element $V(t)$, the specialist on-trajectory probability. Restricts junction firing to high-variance positions (where the specialist's next-token distribution carries enough mass to deform). q_V=0.75 excludes the bottom 75% low-entropy steps.
Warm-up	Number of leading positions in which no junction may fire. Avoids firing inside the prompt prefix or on the opening tokens of the thinking block. Set to 20 tokens.
Top-k (KL/Fisher)	Vocabulary truncation when computing the local KL between specialist and ancestor (and its Fisher approximation) that powers the junction score; the softmax is restricted to the top 20 tokens to suppress tail noise.
Top-k (coverage)	Vocabulary truncation when computing the ancestor's coverage on the specialist support set; uses the top 20 specialist tokens.

Table 15: Notation for the repair-time hyperparameters in Table 16. Sources: the same J_{approx}, $V(t)$ statistics defined in App. A.1 are used both to detect SL-G junctions and to gate SL-R sampling.

L Mechanistic Falsification

We isolate the structural nature of rank misrouting by evaluating interventions under greedy decoding ($T = 0.0$). This protocol removes sampling variance and tests the model's underlying rank geometry directly.

The operator response across the five test-time interventions (Table 13) constitutes the mechanistic falsification suite. SL-G results there use $T = 0$ completion after the injection window; Retry and SL-R use the same protocol. We report results across all K strata directly in the main text rather than duplicating them here.

M Iterative Repair Round-by-Round Breakdown

The Iterative Repair protocol applies up to $N = 3$ sequential rounds of junction detection and logit

injection. Each round targets the first unresolved junction in the most recent failed trajectory. Table 21 reports the cumulative rescue rate and incremental rescue at each round for CruxEval (SFT 1.7B).

N Junction-Feature Cache

The repair@3 protocol (§2) draws each problem-unit's three conditioning attempts from the **junction-feature cache**, a per-rollout store of trajectory geometry summaries that allows the dispatch to look up an operator's outcome at any $K \in \{1, \ldots, 10\}$ without re-rolling the specialist. This appendix documents the cache schema and the extraction procedure.

Per-token tensor keys.

Key	Shape	Meaning
`rollout_idx`	scalar	rollout index r within the already-failed rollout pool
`Delta_path`	$(T,)$	$\log p_S(x_t) - \log p_A(x_t)$ at the realized token
`G_cov`	$(T,)$	Gaussian-coverage gap (m-geodesic)
`V`	$(T,)$	volume element (Fisher information of the local categorical)
`J_approx`	$(T,)$	approximate junction intensity ($\Delta_t^{\text{path}} \cdot G_t^{\text{cov}}$)
`pA_on_set`	$(T,)$	ancestor probability on the specialist's top-k support
`logit_var`	$(T,)$	variance of the specialist's logit distribution at t

The three problem-level features. The three features that drive the regime classifier (§5) are aggregated from these per-token tensors:

- $\bar{J}^{(p)}_{\text{frac}+} = \frac{1}{|\mathcal{R}_p|} \sum_r J^{(r)}_{\text{frac}+}$ where $J^{(r)}_{\text{frac}+} = \frac{1}{T_r} \sum_t \mathbb{I}[\texttt{J_approx}_t > 0]$.
- $\bar{C}^{(p)} = \frac{1}{|\mathcal{R}_p|} \sum_r \frac{\max_t \texttt{J_approx}^{(r)}_t}{\text{mean}_t \texttt{J_approx}^{(r)}_t}$.
- $\bar{V}^{(p)} = \frac{1}{|\mathcal{R}_p|} \sum_r \texttt{V}^{(r)}_{t^{\star(r)}}$ where $t^{\star(r)} = \arg\max_{t>0} \texttt{J_approx}^{(r)}_t$.

The aggregation is rollout-mean across the failed rollouts in the already-failed rollout pool; if a pid has more rollouts than the cache's depth, only the cached rollouts contribute.

Conditioning depth K and the per-pid choice of K. The cache is extracted from up to $K{=}10$ independent failed rollouts of the specialist at $T{=}0.6$. The k-th row of an operator's outcome records the operator's response *conditioned on the spec having failed the first k rollouts*. The per-pid-deepest dispatch mode of §2 selects the three largest available k values from this file as the repair@3 attempts. Three of the ten evaluation cells (sft0p6b/gsm8k, sft4b/cruxeval, grpo1p7b/cruxeval) have baseline pools shallower than $K{=}10$; their problem-units are included at whatever depth their pool supports, and the dispatch analysis in §3 reports per-cell sample sizes so this asymmetry is visible.

O Per-Feature Predictive Signals from the Junction-Feature Cache

The junction-feature cache (App. N) provides six per-token signals derived from the specialist–ancestor disagreement. We extract from each rollout five per-rollout aggregates of each signal (mean, std, fraction-positive, max, 95th-percentile), giving 30 raw per-rollout features. This appendix reports the univariate predictive power of these features for several intervention outcomes, and supports the support-compression interpretation of GRPO that §6 uses.

Outcomes. For each problem-unit we compute eight binary outcomes from the junction-feature cache at $K{=}1$: *geo_pred_correct* (SL-G repair rescues the failure), *retry_correct* (a second-chance retry succeeds), *best_temp_correct* (best of $T_{\text{loc}} \in \{1.0, 1.5, 2.0\}$ succeeds), and the four pairwise outcomes *geo_beats_retry*, *geo_beats_temp*, *geo_beats_both*, *retry_beats_geo*.

Scope of these AUROCs. The outcomes here are per-pid, SL-G-specific (does SL-G rescue? does SL-G beat retry/temp?), and the features below describe what predicts SL-G's per-pid wins. They do not contradict the cell-level finding that SL-G $\approx$ SL-R on most cells (Appendix T): that result is about whether SL-G's *position* selection improves on random within the sparse class, not about whether SL-G wins per pid. Rank-inversion language below describes SL-G's intervention mechanism (logit-steer flips local rank), not a claim that SL-G's localization is uniquely effective.

Feature classes. The 30 features cluster into three classes: G_{cov} aggregates measure the m-geodesic coverage gap between specialist and an-

cestor (how much probability mass the specialist has lost on the ancestor's own top-k support); Δ^{path} aggregates measure the e-geodesic deformation at the realized token (how much more probable the specialist makes its own choice than the ancestor does); and logit_var aggregates measure the Fisher information of the specialist's local distribution.

Headline patterns. Several univariate AUROC patterns hold across cells:

- G_{cov} **dominates rescue prediction on SFT/code-math cells.** For SFT specialists on CruxEval and GSM8K, $G_{\text{cov_mean}}$, $G_{\text{cov_std}}$, and $G_{\text{cov_p95}}$ lead univariate prediction of *geo_pred_correct* with AUROC ≈ 0.65–0.78. Lower coverage gap predicts SL-G success: a specialist that retains probability mass on the ancestor's support is precisely the case where logit-steer can re-rank a junction without losing fluency.

- $\Delta^{\text{path}}_{\text{at_that}}$ **predicts where SL-G uniquely beats temperature.** $\Delta^{\text{path}}_{\text{at_that}}$ (the path deformation at the the detector's fired token) leads univariate prediction of *geo_beats_temp* and *geo_beats_both*, with AUROC up to ≈ 0.80 on sft4b CruxEval. The positions where rank inversion most decisively beats temperature are those with the largest realized-token deformation.

- logit_var **arbitrates SL-G vs. retry.** Lower specialist logit variance (a sharper, more committed distribution at the junction) predicts SL-G uniquely beats retry; higher $\Delta^{\text{path}}_{\text{mean}}$ predicts retry wins. The sharpness of the local distribution is the dimension along which the rank-preserving vs. rank-inverting probe decision pivots.

- **GRPO shifts the predictive picture.** On GRPO specialists, V_{mean} (the Fisher information of the temperature submodel, averaged over the trace) becomes the leading predictor of *geo_pred_correct*, with AUROC ≈ 0.63–0.79. G_{cov} weakens substantially: the m-geodesic coverage gap loses predictive power because GRPO training broadens rather than narrows the specialist's support relative to the ancestor.

Implication for the GRPO collapse. The shift in the dominant predictor from G_{cov} to V_{mean} on GRPO models is the mechanistic basis for the *Distributed Deformation* regime collapsing onto GRPO in §6. The m-geodesic coverage interpretation of G_{cov} assumes coverage *contraction* relative to the ancestor; if the specialist's support is broadened or saturated near the ancestor (as RL-driven post-training does), then G_{cov} no longer carries failure information and the deformation becomes trace-wide rather than junction-local. Under our five-regime taxonomy, GRPO failures all land in *Distributed Deformation*, where DL would be required if it weren't dominated by the other operators in the dispatch comparison (§9).

Per-cell breakdown. For each cell and each outcome, the top-3 univariate predictors and their AUROCs are tabulated in the supplementary artifacts. Two illustrative rows:

Cell	Outcome	Top feature (AUROC)	Mechanism class
sft1p7b/cruxeval	*geo_pred_correct*	$G_{\text{cov_std}}$ (0.738)	G_{cov}
sft1p7b/cruxeval	*geo_beats_retry*	$J_{\text{approx_frac_pos}}$ (0.355)	J_{approx}
sft4b/cruxeval	*geo_beats_temp*	$\Delta^{\text{path}}_{\text{at_that}}$ (0.80)	Δ^{path}
grpo1p7b/gsm8k	*geo_pred_correct*	V_{mean} (0.79)	V (Fisher info)

The mechanism class shifts cleanly between SFT and GRPO: on SFT specialists, geometric features dominate (contraction in G_{cov}); on GRPO, Fisher-information features dominate (V_{mean}). This is the structural reason the regimes fall out of the geometry without seeing operator outcomes at classifier training time.

P Dispatch Score γ Sensitivity

The dispatch score $S_i = U_i \cdot R_i^{\gamma}$ (Eq. 1) uses $\gamma = 0.5$ throughout the main text. We chose $\gamma = 0.5$ because it gives a geometric-mean-like blend of unique coverage U_i and absolute rescue rate R_i, but the more important property is that the *policy* selected by the dispatch is stable over a range of γ values. This appendix records the sweep.

Setup. We re-ran the dispatch aggregation at $\gamma \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ on the same per-pid `_at3` outcomes used in the main text. For each γ we record ΣS for every policy (dispatch and each single-operator baseline) and confirm the per-cell argmax-S_i operator selection.

Policy selection is invariant. The dispatch policy is identical at every γ in the sweep: each cell's argmax-S_i operator is the same across all six γ values. Most cells route to a non-retry operator: *SL-G* for Distributed Deforma-

tion/L, Rank Misrouting (junction-diffuse)/H, Unresolved/L; *SL-R* for Distributed Deformation/H and Unresolved/H; T_{loc}=1.5 for Rank Misrouting (junction-diffuse)/L. The two small cells (geometry_local/H, distributed_source/L) route to *retry*. The dispatch's qualitative behavior does not depend on the specific blend exponent.

Dispatch lift over retry is monotonic and stable. ΣS scales with γ in absolute terms (larger γ penalizes low-R operators more), but the lift over the retry baseline is approximately constant across the sweep:

γ	dispatch ΣU	dispatch ΣS	retry ΣU	retry ΣS	$\Delta\Sigma S$ vs. retry
0.3	789	670.2	755	633.6	+36.6
0.4	789	635.5	755	598.5	+37.0
0.5	789	603.1	755	565.7	+37.4
0.6	789	572.6	755	535.1	+37.6
0.7	789	544.1	755	506.4	+37.7
0.8	789	517.3	755	479.6	+37.7

ΣU does not depend on γ (it counts unique pids rescued, independent of the score) and stays at 789 for dispatch and 755 for retry across the entire range. $\Delta\Sigma S$ varies by under 1.1 over the six γ values, the lift sits in $[36.6, 37.7]$.

Implication. The dispatch lift, the policy composition, and the relative ordering of all five single-operator baselines are stable across $\gamma \in [0.3, 0.8]$. The choice $\gamma = 0.5$ is a presentational convenience (it yields a geometric-mean-like aggregate), not a load-bearing tuning knob.

Q Feature stability and cell-vs-pid scope

This appendix reports the empirical checks supporting the methodological choices in §5 and the population-scale scoping in §9. Section Q.1 reports the ICC of each candidate feature across rollouts of the same problem-unit; Section Q.2 reports the cell-vs-pid predictive accuracy that motivates the population-scale framing.

Q.1 Within-pid ICC

For each cell and each candidate feature f we compute the one-way random-effects ICC, $\text{ICC}(1) = \sigma^2_{\text{between}}/(\sigma^2_{\text{between}} + \sigma^2_{\text{within}})$, where $\sigma^2_{\text{between}}$ is the variance of per-pid means and σ^2_{within} is the average within-pid variance across rollouts. ICC measures how concentrated within-pid each feature is; we use it to characterize each feature, not as a gating threshold. The features evaluated are $\bar{J}_{\text{frac}+}$, $\log_{10}\bar{C}$, $\log_{10}\bar{V}_{\text{traj}}$ (trajectory-mean steerability, the dispatch H/L routing key), and $\log_{10}\bar{V}_{t^\star}$ (junction Fisher info, the regime classifier feature).

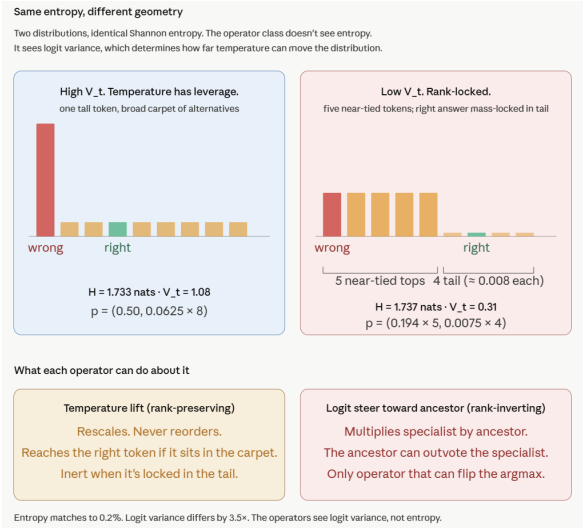

Figure 8: **Entropy alone does not equal steerability (motivating example).** Per-problem token-level entropy (H) against the Fisher-information local displacement budget $V_{t^\star}$ at the worst pivot. The two quantities are imperfectly correlated; the vertical spread at any given H is the variance the routing rule exploits via the $V_{t^\star}$ branch of Eq. 3.

Sensitivity check: fixed-position V is unusable. We also evaluated $\log_{10}V$ at three fixed proportional positions in the post-warmup region (20%, 50%, 80% of trace length). Pooled ICCs are 0.16, 0.18, and 0.19 respectively, well below the problem-level threshold and below even $\log_{10}\bar{V}_{t^\star}$. The trajectory mean is the only aggregator that yields a stable per-pid statistic; both "V at the spike" and "V at a fixed fraction of the trace" inherit too much position-dependent variability.

Q.2 Cell-vs-pid predictiveness

The dispatch lift in §3 ($\Delta\Sigma S = +35.7$) is a population-scale result: it reflects stable cell-level operator preferences. The natural follow-up question is whether the same three features predict the *per-pid* operator winner. They do not.

For the 507 problem-units with a decisive operator winner (gap > 5 pp between the best and second-best operator at the deepest available K), we compare four predictors:

The quadrant $\times$ winner contingency table is significant under chi-square ($\chi^2 = 104.2$, dof $= 12$, $p < 10^{-15}$, Cramér's $V = 0.262$): the relationship is real but the effect size is too small to lift majority-vote accuracy above baseline. Per-operator one-

feature AUCs (Table 24) explain why: DL-winner prediction reaches AUC 0.79 from $\bar{J}_{\text{frac+}}$ (DL is the operator most cleanly characterized by the failure being broad and unsteerable), but retry, SL-G, and SL-R, the three operators that account for 83% of decisive winners, sit between 0.50 and 0.69 on every single feature.

Cross-rollout coherence does not close the per-pid gap. To rule out the most natural alternative explanation, that the within-rollout features average over per-rollout heterogeneity and that the operator-relevant signal lives in cross-rollout *coherence*, we computed three families of cross-rollout features for every pid with ≥ 2 cached rollouts (n=2,024): spike-position pairwise agreement S_p (fraction of rollout pairs whose $\arg\max_t J_t^{(r)}$ falls within a 32-token window), the rollout-level standard deviation of $V_{t^\star}$ (steerability volatility), and the rollout-mean and rollout-std of three ancestor- disagreement quantities at the spike position (G_{cov}, $p_S^{\text{on-set}}$, Δ_{path}). On the n=507 decisive-winner subset, no single cross-rollout feature moves a median-split classifier above the 0.505 baseline (lift $+0.000$ for every feature tested). A multivariate logistic regression on all 14 cross-rollout features achieves 0.485 (lift -0.020); adding the three within-rollout features brings it to 0.485 as well. The signal that does emerge, S_p predicts DL-winners at AUC 0.75, is the same DL-routing signal already captured by $\bar{J}_{\text{frac+}}$ (AUC 0.79), with S_p and $\bar{J}_{\text{frac+}}$ tracking the same underlying property (diffuse drift $\Rightarrow$ DL wins, with scattered spikes as a corollary). For the four non-DL operators that account for 95% of decisive winners (retry, SL-G, SL-R, T_{loc}=1.5), every single feature we evaluated, within- rollout aggregates, cross-rollout coherence, ancestor-disagreement statistics, sits in $[0.50, 0.69]$ AUC. The per-pid operator winner is not in the failure trace.

Confidence-stratified dispatch lift. A natural calibration check: if the regime classifier is well-calibrated, the dispatch lift should grow with classifier confidence. We compute the per-pid ΔU (dispatch unique-rescues minus retry unique-rescues) at confidence thresholds $\text{conf}_p \geq \{0.0, 0.20, 0.40, 0.60\}$ on the deepest-K subset. The per-pid lift is $+2.9$ pp at $\text{conf}\geq 0$, $+1.9$ pp at $\geq$ 0.20, $+2.6$ pp at $\geq$ 0.40, and $+3.1$ pp at ≥ 0.60. The trend is non-monotonic and the absolute differences are within sampling noise; we do not claim a calibration-driven sharpening of the lift. The cell-level dispatch $\Delta\Sigma S = +35.7$ remains the load-bearing measurement; this slice indicates that the lift is robust to confidence filtering but that the classifier confidence is not itself a strong predictor of per-pid rescue likelihood beyond the regime label.

Routing key ablation: $\bar{V}_{\text{traj}}$ vs. $V_{t^\star}$. The dispatch H/L split uses $\log_{10}\bar{V}_{\text{traj}}$ (pooled ICC 0.88, within-pid stable). We also evaluated $\log_{10}\bar{V}_{t^\star}$ (junction Fisher info, pooled ICC 0.48) as an alternative routing key. The two variants give essentially equivalent dispatch performance: $\bar{V}_{\text{traj}}$ achieves ΣU=892, ΣS=702.6; $V_{t^\star}$ achieves ΣU=896, ΣS=703.5 ($\Delta\Sigma S$=-0.9 in favor of $V_{t^\star}$, within bootstrap noise; the per-pp efficiency vs. retry@64 is also within $\sim 1\times$). The two splits partition each regime in opposite directions (e.g., RM-G is mostly $V_{t^\star}$-high, steerable spike, but mostly $\bar{V}_{\text{traj}}$-low, low trajectory- average Fisher info), routing the regime mass to the same operator class either way. We use $\bar{V}_{\text{traj}}$ as the routing key because the per-pid H/L decision needs a deterministic split; the classifier instead uses $V_{t^\star}$ because the junction profile is what mechanistically distinguishes regimes (§5). $V_{t^\star}$ can be selected as the routing key via `-hl_feature=v_tstar` for reproducibility.

What this means. The three trajectory features identify *population-scale failure topography*, regime structure, average steerability, alignment with post-training condition, but do not predict the rescuing operator on any specific failed problem, and adding cross-rollout coherence does not close the gap. The dispatch policy in §3 converts this population structure into a routing recommendation per (model, task) deployment, not per problem instance. Closing the per-pid gap likely requires features outside the failure-trace geometry, the problem statement itself, partial-execution traces, or decoder-state information that the junction-feature cache does not capture. We leave this to future work.

R Ancestor choice: detector calibration check

The dispatch protocol throughout the main paper uses windowed junction firing (Appendix A.1) as its position-selection rule and the lineage pre-training checkpoint as the ancestor for logit-steer operators. This appendix asks two design ques-

tions:

1. Does the choice of ancestor matter? Specifically, does the lineage pre-training checkpoint recover more failures than a strong non-lineage model from the same architecture family?

2. Is this answer detector-dependent?

The answer to both is: yes, but they trade off. The ancestor identity matters under a poorly-calibrated detector, and the apparent advantage disappears under a well-calibrated one. Specifically, the 21-pp lineage advantage observed under the original trace-relative percentile detector (Detector A, the published §4.1 result, hereafter Detector A) collapses to within sampling error under the windowed junction-firing detector (the production routing rule the rest of the paper uses).

Protocol. SFT 4B specialist trained on Bespoke-Stratos-17k; CruxEval evaluation set. We compare four ancestors with the same logit-steer budget ($\alpha = 0.7$, single round): the lineage Qwen3-4B base model, two non-lineage Qwen2.5 baselines (1.5B-Instruct and 7B-Instruct), and the same Qwen2.5-7B-Instruct ancestor again under our junction detector. All numbers are reported as repair@3 (any-of-three injection attempts per problem-unit) on the same 77-pid intersection from the original ablation (pids with paired data across all three Detector A ancestor protocols).

Detector definitions. **Detector A** (the published §4.1 firing rule): single injection at the trace-relative 90th-percentile demotion score, computed on one failed rollout per pid. **junction-firing** (the production detector used by the dispatch): windowed scan $S_t = \sum_w J_t > \lambda_J$ with λ_J calibrated per-problem as the 0.99-quantile of S_t across the pid's background pool (App. A.1). junction-firing reads off the same trajectory feature space the regime classifier uses; Detector A is position-selective on a single trace.

Result. Under Detector A, the lineage ancestor recovers 21.3 pp more failures than the non-lineage Qwen2.5-7B (46.7% vs. 25.4% on the matched 77-pid set). Under junction-firing the gap collapses to -0.6 pp (50.0% vs. 50.6%, non-lineage numerically ahead by less than a sampling-noise amount). The lineage advantage is detector-dependent.

Interpretation. Two readings are consistent with the data, in increasing order of strength:

Detector A under-counted non-lineage rescue. Detector A's percentile rule is a single trace-relative cut; it selects positions where the specialist disagrees with the ancestor by a trace-relative margin. The Qwen2.5-7B ancestor's recoverable positions sit at margins that fall below Detector A's per-problem 90th percentile. junction-firing's per-problem calibration (λ_J scales with the background pool's volatility) catches positions the percentile rule misses. The published lineage advantage was a property of the firing rule's interaction with the ancestor's logit distribution.

Ancestor identity is interchangeable for our operator class under a well-calibrated detector. Within the capability range we tested (Qwen3-4B as lineage vs. Qwen2.5-7B as a stronger non-lineage same-family ancestor), both deliver rescue rates that are statistically indistinguishable when paired with the junction detector. We do not claim either ancestor is structurally privileged for this operator class. We continue to use the lineage ancestor as the reference distribution in dispatch for protocol simplicity (we already train and ship the specialist's pre-training checkpoint; no additional model download), not because it is uniquely correct geometrically.

Detector A robustness to repair@k. The lineage gap under Detector A is stable across repair budgets, not specific to the k=3 choice. Lineage vs. Qwen2.5-7B repair@k rescue on the 77-pid set: +28.6 pp at k=1, +31.3 pp at k=2, +21.3 pp at k=3 (the published §4.1 number), +18.8 pp at k=4, +12.4 pp at k=5. The advantage is detector-dependent (collapses under junction-firing; Table 25), not budget-dependent.

Why we report this. The original 49.4% vs. 18.2% result was a load-bearing claim for an earlier framing of this work that emphasized the lineage ancestor as a structural reference. Under junction-firing the claim no longer holds, and we have removed the corresponding "lineage as structurally privileged" language from the body of the paper. The dispatch result (§3) is independent of this finding: both sides of the $\Delta\Sigma S$=+35.7 and +17.7 lifts were already computed under the junction detector. The dispatch's class-routing decision remains the load-bearing engineering claim.

S Retry-by-regime + dispatch FLOPs

This appendix supplies the statistical backing for the §3.4 claim that retry's per-pid rescue rate varies by a factor of two across regimes, plus the FLOPs accounting for the current dispatch policy.

Retry rescue rate per regime, at one and three attempts (Table 26). Per-pid retry rescue under repair, stratified by regime label. Two operating points: **retry@1** (the deepest-K single rollout rescue; the cheapest attempt) and **retry@3** (any-of-three deepest-K rollouts; the protocol the dispatch operates under). Bootstrap 95% CIs (n_{boot}=10,000); pairwise contrasts use an unpaired percentile bootstrap.

The gap between regimes is largest at low budget and shrinks as attempts grow, in the direction the regime taxonomy predicts: most non-RM-G failures are sample-recoverable and retry approaches its asymptote within a few attempts, while RM-G failures cluster at a single load-bearing junction that resampling cannot move regardless of attempt count. Concretely:

- retry@1: Unresolved 48.3% vs. RM-G 21.6%, $\Delta = -26.7$ pp $[-33.4, -19.7]$, $p<0.0001$ ($\approx 2.2\times$).
- retry@3: Unresolved 65.1% vs. RM-G 39.6%, $\Delta = -25.5$ pp $[-31.8, -19.2]$, $p<0.0001$ ($\approx 1.6\times$).

RM-G gains +18 pp from two extra attempts (the bottom of the curve); Unresolved gains +17 pp (already higher to start). The gap remains significant at every attempt count we measured.

Dispatch compute accounting (current 2D-centroid policy). We measure compute in *single-rollout units*: one specialist rollout at the spec generation budget of 1024 tokens = 1.0. Each operator's per-attempt cost is its specialist generation (1.0) plus an ancestor prefix-fill to the median junction-detector firing position $\hat{t}$, scaled by the spec budget. The full per-operator breakdown and the dispatch's routed-fraction-weighted cost are in Table 27: dispatch costs $\approx$ 1.34 rollout-units/attempt, so at repair@3 the dispatch spends $\approx$ 4.0 rollout-units per failed pid versus iso-budget retry@3 at 3.0, a +34% compute overhead, not +85% as under a conservative full-ancestor accounting.

This is the cost *conditional on the specialist failing*. The router defers cleanly on passing problems (no ancestor invocation), so deployment-scale overhead is this number multiplied by the specialist's failure rate on the target traffic.

Budget-matched retry catch-up (Table 28). For the two cells where deep-K retry pools exist (sft-1.7B and GRPO-1.7B $\times$ CruxEval at T=0.8, K=64 rollouts per pid) we can measure absolute retry catch-up directly. Restricting to pids in the dispatch's Fail@K=10 set (intersection: n=220 and n=225 respectively), retry's per-pid pass rate climbs from $\sim$ 65% at K=10 to $\sim$ 77% at K=64, exceeding the dispatch's repair@3 rescue rate on the same pids by +6 to +13 pp. The catch-up is real in absolute terms, but the compute gap is substantial: retry@64 spends $\approx 16\times$ more rollout-units per failed pid than dispatch@3 (4.0 rollout-units at 3 attempts $\times$ 1.34 units/attempt), and the marginal cost beyond dispatch's operating point is in the hundreds of additional units per extra pp of rescue. At iso-rescue, retry needs $K\approx$6 on the 2-cell scope to match dispatch@3, $\sim 1.5\times$ more compute on a single-rollout basis.

Per-regime catch-up on sft-1.7B CruxEval (Table 29). Stratified by regime, retry@64 still exceeds dispatch@3 across all regimes (including R.M. (geo-local), the regime our binary-decision claim says retry "cannot move"). The "rank-locked junction" framing does not survive at this budget under T=0.8; either the higher temperature breaks the rank lock or the junction-stability claim only holds at bounded retry. The dispatch's value at large budget is therefore Pareto-efficiency, not absolute superiority.

Predictive extrapolation to the other six V9 cells was attempted (per-problem Bernoulli mixture fit on rollouts[0..9]) but the model overshoots on these two cells by $\sim$22 pp at K=32 and K=64, so we do not report extrapolated catch-up numbers for cells without measured K=64 data.

T Paired SL-G vs. SL-R by dispatch cell

Within the sparse-logit-steer class, the operator identity (SL-G vs. SL-R) carries little additional lift beyond inject-vs-retry. Restricting to problem-units with paired outcomes available for both operators, SL-G and SL-R agree on rescue within 3 pp on 59.6% of paired pids. The cell-level breakdown (Table 30) shows one cell with a decisive SL-G advantage (Distributed Deformation at high $V_{t^\star}$, $\Delta_{\text{SL-G}} = +3.5$ pp, n_{pair}=116); no cell

with $n_{\text{pair}} \geq 30$ shows a decisive SL-R advantage ($\Delta_{\text{unif}} > 8$ pp), the detector is rarely worse than random and occasionally usefully better.

U geometry_local/H Position-Gap Analysis

Rank Misrouting (geo-local) at high $\bar{V}$ is the largest single dispatch cell (n=446) and the only substantial cell where *SL-R* repair leads *SL-G* repair by a non-trivial margin in the paired sub-cell analysis (Table 30; $+4.04$ pp, "mixed" bucket). Two readings of this gap matter for paper-level claims:

- **Reading 1 (threshold tunable).** The 90th-percentile demotion threshold (§A.1) is over-aggressive in this regime, it concentrates injection mass on a single position when the recoverable injection-worthy positions are distributed across the trace. The SL-G and SL-R populations should have SL-R-winning positions *scattered* relative to SL-G's selected junction.
- **Reading 3 (multi-modal injection sites).** The regime has multi-modal injection sites that the demotion detector misses; SL-R wins because it occasionally lands at a hidden second junction. Under this reading the SL-R-winning positions should *cluster* at specific non-SL-G positions that the detector should have flagged.

We discriminate the two by computing the per-attempt position gap $\|t^{\star}_{\text{SL}-\text{G}} - t_{\text{rand}}\|$ on every SL-R-only attempt (i.e., the attempts where `rand.correct` $= 1$ and `geo_pred.correct` $= 0$). If the SL-R-winning positions cluster near $t^{\star}_{\text{SL}-\text{G}}$, the clustering would surface a second injection mode adjacent to SL-G's choice; if they are scattered, the SL-G detector is approximately right and SL-R's lead is a hedging effect.

Per-attempt setup. For the 446 Rank Misrouting (geo-local)/H problem-units we collect every junction detector row across the four cells dominated by this regime (sft0p6b/gpqa, sft1p7b/gpqa, and the smaller geometry_local contingents of the other v9 cells). This yields 3,889 valid attempts of which 439 are SL-R-only wins. For each, we record both $t^{\star}_{\text{SL}-\text{G}}$ and t_{rand} in token units, as well as the rollout length n_{tokens} used to compute the fractional gap $|t^{\star}_{\text{SL}-\text{G}} - t_{\text{rand}}|/n_{\text{tokens}}$. A within-32-token gap is the same coarse-graining window used by the cross-rollout phase-stability analysis (Appendix A).

Results.

Interpretation. The SL-R-winning positions are essentially uniform with respect to $t^{\star}_{\text{SL}-\text{G}}$. The within-32-token share is indistinguishable from a position-shuffled null (observed 4.8% vs. null 4.6%; enrichment ratio $1.05\times$), and the crude bimodality ratio of 1.64 sits below the > 2 threshold at which one would suspect a clean two-mode distribution. There is no evidence of a second injection mode adjacent to SL-G's selected junction.

The Rank Misrouting (geo-local)/H gap is therefore consistent with *Reading 1*. The 90th-percentile threshold over-concentrates injection mass on a single position when the recoverable positions are distributed across the trace, and SL-R's hedging across many positions captures the same problems SL-G misses. A natural methodological refinement is ensemble injection at the 70-80th percentile (multi-position) or a temperature-aware threshold; we leave systematic threshold tuning to future work because it lies outside the scope of the structural claim ("the regime correctly flags SL-G-recoverable failures"). The mechanistic claim, that Rank Misrouting (geo-local) failures are recoverable by *local* injection, survives. The detector correctly identifies the sub-population; it does not yet correctly identify every injection-worthy position within that sub-population.

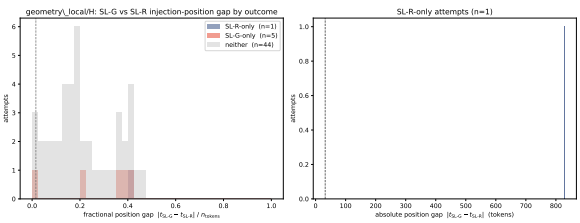

Figure 9: **Position-gap distribution within Rank Misrouting (geo-local)/H.** Left: fractional gap $|t^{\star}_{\text{SL}-\text{G}} - t_{\text{rand}}|/n_{\text{tokens}}$ by attempt outcome, SL-R-only (rand correct, SL-G wrong) in dark blue, SL-G-only in red, both in grey. Vertical dashed line marks the $|gap| < 32$-token boundary. Right: absolute position gap (tokens) for SL-R-only attempts only. Both panels show the SL-R-winning positions distributed across the trace with no enrichment near $t^{\star}_{\text{SL}-\text{G}}$ (observed 4.8% vs. null 4.6% within 32 tokens; enrichment $1.05\times$).

V Figure 1 with rescue-rate y-axis (appendix variant)

Figure 1 in the main text uses the paper's headline metric $\Sigma S = \sum_c U_c \cdot R_c^{1/2}$ (dispatch score, summed across dispatch cells c) on the y-axis. Figure 10 below is the same plot with raw rescue rate (% of failed problem-units that any rollout at the given operator and K rescues) on the y-axis instead. The two panels show the same Pareto structure, the dispatch curve dominates the single-operator curves at matched FLOPs in the budget regime we measured, but at slightly different absolute magnitudes. ΣS is sensitive to per-cell U and rewards covering more distinct dispatch cells; raw rescue is the simpler population-level average. The conclusion ("diagnose first, then route compute") is robust to the choice of metric.

W Extended Results

X Bootstrap Confidence Intervals

Table 32 reports 95% bootstrap confidence intervals (10,000 resamples, problem-level) for the headline cells in Table 13. The GSM8K SFT 4B cell has N=13 problems; its wide intervals reflect stratum size, not rescue strength. All headline claims in the main text are drawn from the CruxEval and GPQA SFT 1.7B cells where $N \geq 58$.

Table 33 reports binomial 95% CIs for the original Detector-A ancestor identity ablation. **This result is superseded by the junction-firing replication in App. R**: under the junction detector (the production calibration), the lineage advantage collapses to -0.6 pp on the same 77-pid set. The Detector-A numbers below are preserved for completeness and to document the discovered detector-dependence. The paper's current position is that ancestor identity is interchangeable within capability range under a well-calibrated detector (Table 25); the body framing has been updated accordingly. Reading this table in isolation would misrepresent that position.

Y Qualitative cases and per-cell response profiles

The regime classification and dispatch decisions look like the following at the level of individual cases and individual (model, task) cells. Three qualitative cases illustrate per-failure diagnosis (Fig. 11); Fig. 12 sweeps the response profile across representative (model, task) cells.

Distributed-Deformation case study (Case C)

Complementing Figure 11 (Rank Misrouting (junction-diffuse) and Rank Misrouting (geo-local)), Case C below illustrates the third regime from Table 34: Distributed Deformation at low $\bar{V}_{\text{traj}}$, routed by the cell-level dispatch to **SL-G**.

Z SFT × CruxEval Regime Composition

Section 6 reports that SFT × CruxEval splits into two qualitatively different regimes at our two scales: **SFT 1.7B × CruxEval** is the regime-mixing hub (broadly Unresolved with substantial minorities of every other regime), while **SFT 4B × CruxEval** collapses largely into Distributed Deformation (the same regime as GRPO at this scale). This appendix gives the per-regime composition and per-regime operator preferences inside the two cells.

Regime composition.

Regime	sft1p7b/cruxeval (n=220)		sft4b/cruxeval (n=243)	
	n	share	n	share
Unresolved	173	78.6%	76	31.3%
Distributed Deformation (DD)	1	0.5%	103	42.4%
Rank Misrouting (junction-diffuse) (RM-D)	29	13.2%	62	25.5%
Rank Misrouting (geo-local) (RM-G)	17	7.7%	2	0.8%

The two cells differ markedly. SFT 1.7B is dominated by Unresolved (78.6%), with the remaining problem-units distributed across RM-D (13.2%), RM-G (7.7%), and a single Distributed Deformation outlier. SFT 4B, by contrast, has 42.4% Distributed Deformation and 25.5% Rank Misrouting (junction-diffuse) and produces essentially no RM-G problem-units. The shift from RM-G-and-Unresolved at 1.7B to DD-and-RM-D at 4B is consistent with larger SFT models inducing broader, more distributed deformation on code tasks, the same structural signature §6 attributes to GRPO.

Operator preferences within each cell.

Cell	Regime	n	retry	SL-G	SL-R	DL	T_{loc}=1.5
sft1p7b/cruxeval	Unresolved	173	**0.676**	0.642	**0.676**	0.578	0.671
	RM-D	29	0.483	0.552	0.414	**0.621**	0.448
	RM-G	17	0.529	0.529	0.529	0.412	0.529
sft4b/cruxeval	DD	103	0.932	**0.971**	0.961	0.375	**0.971**
	RM-D	62	0.726	0.726	0.726	0.189	0.694
	Unresolved	76	0.658	0.776	0.763	0.474	**0.789**

Two structural observations: (i) within Unresolved/RM-D at both SFT scales the rescue rates cluster within an $\sim$ 8-pp band, which is the methodological-bucket fingerprint (operator choice is fungible); and (ii) DL is consistently dominated as a single operator on both cells, reinforcing the global finding in §9.

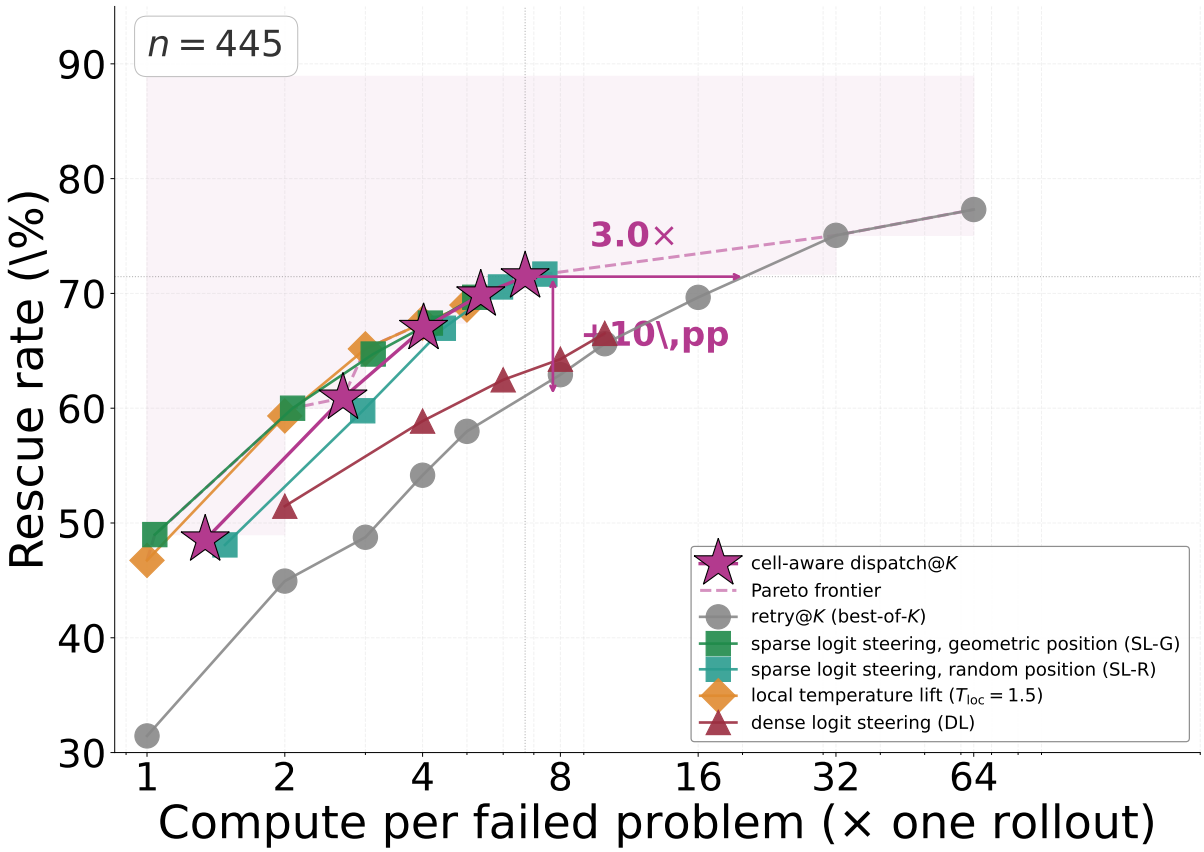

Figure 10: **Figure 1 with rescue-rate y-axis.** Same data and same scopes as Figure 1; the y-axis is raw rescue rate (% of failed problem-units) instead of dispatch score ΣS.

Case A: local temperature lift rescues a junction-diffuse failure that retry and logit-steer cannot `sft-1.7B`, CruxEval `sample_141`

Task (CruxEval). Trace `f(li) = [li.count(i) for i in li]` on input `['k','x','c','x','x','b','l','f','r','n','g']`. Expected output: `[1, 3, 1, 3, 3, 1, 1, 1, 1, 1, 1]` (per-position counts).

Failed rollout (T=0.6, marked incorrect):

```
    ⟨think⟩⟨/think⟩
k', 'x', 'c', 'x', 'x', 'b', 'l', 'f', 'r', 'n', 'g'


    The output of the function f is a list of counts of each element in the input
    list . . . 'k' appears 1 time, 'x' appears 3 times . . . So the final output is '['k',
    'x', 'c', 'x', 'x', 'b', 'l', 'f', 'r', 'n', 'g']'.
```

The model immediately commits to repeating the input list, then enumerates the correct counts in prose, but the final-commit token re-anchors to the input instead of the counts.

Feature profile. $J_{\text{frac}+}\approx 0.38$ (moderate spread); $J_{\max}/J_{\text{mean}}$ rank 0.92 (sharp spike at the early commit); $\bar{V}_{\text{traj}}\approx 0.90$ (high — overall trajectory carries high Fisher info), but $V_{t^\star}\approx 0.13$ (low at the spike token — the wrong commitment is locally rank-locked). This profile lands in **Rank Misrouting (junction-diffuse)** at high $\bar{V}_{\text{traj}}$; the cell-level dispatch routes that cell to $T_{\text{loc}}=1.5$ (Table 1).

Consistent with the dispatch routing. Sparse logit-steering has no movable target: the spike sits on a high-confidence wrong token that the ancestor signal cannot displace under bounded steering ($\alpha=0.7$). Retry at $T=0.6$ resamples the same anchored distribution and exhausts the same trap. Local temperature lift in the junction window injects enough variance to escape the early commitment, after which the (correct) counting reasoning is preserved.

Repair@3 (repair depths): retry=✗ SL-G=✗ SL-R=✗ $T_{\text{loc}}=1.5$=✓

Case B: SL-G rescues a Fail@5 geo-local failure that retry cannot `sft-1.7B`, GPQA `gpqa_recAAJoHMW45Lv5je`

Task (GPQA Diamond). Chemistry: an equimolar mixture X of two liquids decolorises bromine water; mixture Y of related compounds is given; hydrogenation of either yields the same single product Z. Identify the class of compounds and the carbon count.

Failed rollout (T=0.6, 5 attempts, all incorrect; trace runs 27k chars):

```
⟨think⟩ . . . that makes me think they are unsaturated hydrocarbons, because bromine water
decolorizes alkenes and alkynes. But the problem mentions no conjugated multiple bonds in
mixture X. So they can't have double bonds, but maybe triple bonds? Or maybe alkenes with
some other structure? Wait, no, if they have. . .  [continues for 27 k characters, never commits]
```

Feature profile. $J_{\text{frac}+}\approx 0.62$ (broad spread cascading from an early pivot); $J_{\max}/J_{\text{mean}}$ rank 0.81 (sharp spike); $\bar{V}_{\text{traj}}\approx 0.82$ (high) but $V_{t^\star}\approx 0.27$ at the junction — the pivot is steerable. This profile lands in **Rank Misrouting (geo-local)** at low $\bar{V}_{\text{traj}}$ relative to the population median; the cell-level dispatch routes that cell to **sparse logit-steer** (Table 1; cell-aggregate winner is SL-R, with SL-G ≈ SL-R at the cell level — Appendix T).

Consistent with the dispatch routing. The early pivot carries the spike and is locally steerable, so a single sparse logit-injection at that position restores the correct framing and the rest of the chain follows (the broad downstream spread is the *consequence* of one wrong premise, not an independent deformation). Retry at $T=0.6$ cannot locate the pivot through sampling alone — the trace stays in the same local basin and exhausts its budget rambling.

Repair@3 (repair depths): retry=✗ SL-G=✓ SL-R=✗ $T_{\text{loc}}=1.5$=✗

Figure 11: Routing case studies under V_{traj}-based dispatch. Cases A and B sit in two distinct (regime, $\bar{V}_{\text{traj}}$) cells of the dispatch (Table 1); the cell-level dispatch routes each to a different operator class. Case A (CruxEval, Rank Misrouting (junction-diffuse), sft-1.7B): sharp early commit token where the wrong choice is rank-locked (low $V_{t^\star}$) but the trajectory carries high entropy elsewhere (high $\bar{V}_{\text{traj}}$) — dispatch routes to local temperature lift, which breaks the commit. Case B (GPQA, Rank Misrouting (geo-local), sft-1.7B): sharp spike at an early steerable pivot — dispatch routes to sparse logit-steer; SL-G at the junction-detected position rescues. The per-trace mechanistic stories illustrate why the cell-level routing is sensible; the dispatch lifts in §3 are at the population scale, not per-pid. Case C (Distributed Deformation) and the feature glossary: Appendix Y.

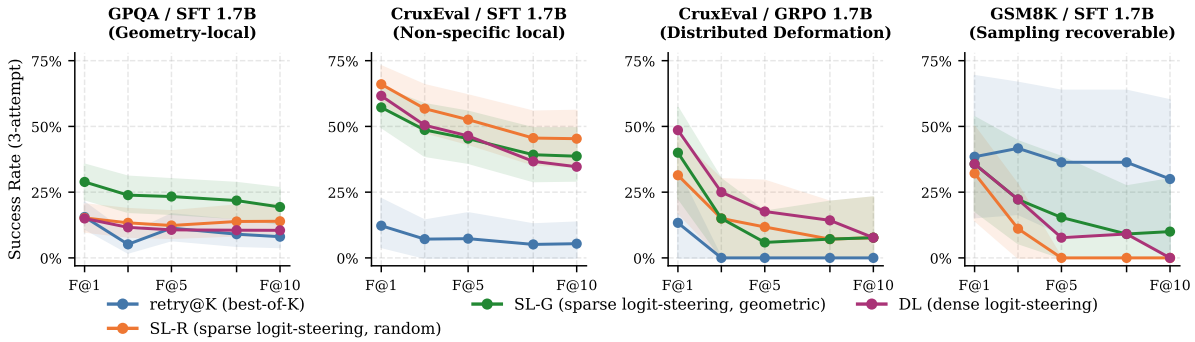

Figure 12: **Representative (model, task) response profiles across Fail@K strata.** Problem-level rescue rates at Fail@K for four representative (model, task) cells. **(A)** Rank Misrouting (geo-local) (GPQA, SFT 1.7B), SL-G outperforms random steering on hard problems; **(B)** Rank Misrouting (junction-diffuse) (CruxEval, SFT 1.7B), local intervention rescues, but junction specificity adds little (one cell where this conclusion reverses at deep K; see §3); **(C)** Distributed Deformation (CruxEval, SFT 4B), DL dominates; **(D)** Unresolved (GSM8K, SFT 1.7B), no probe consistently wins; retry is often competitive but does not dominate. Operator labels in the panel legends use the standard acronyms introduced in §2: retry@K, SL-G (sparse logit-steering, geometric position), SL-R (sparse logit-steering, random position), DL (dense logit-steering).

Case C: SL-G rescues a distributed-deformation failure that retry and SL-R cannot `sft-4B`, CruxEval `sample_348`

Task (CruxEval). Trace `f(d) = d.copy()` on input `{563: 555, 133: None}`.

Failed rollout (T=0.6, marked incorrect):

```
⟨think⟩ The function f takes a dictionary and returns its copy. . . the .copy() method
creates a shallow copy. . . the returned dictionary should have the same key-value pairs. . .
but wait, when you copy a dictionary, the keys and values are copied, but if values are
mutable they are referenced. . . in Python 3.7+ insertion order is preserved. . . the
return value is the dictionary object. . . ⟨/think⟩
Return value: {563: 555, 133: None}
```

The trace contains the correct dict literal but the model bracketed it with a markdown prefix (`**Return value:**`) that the strict matcher rejects.

Feature profile. $J_{\text{frac}+}\approx 0.53$ (broad spread); $J_{\max}/J_{\text{mean}}$ rank 0.005 (extremely diffuse — no dominant spike at any single token); $\bar{V}_{\text{traj}}$ in the low half of the population (the trace's per-rollout Fisher info averaged across positions is near the cohort median for the regime). This profile lands in **Distributed Deformation** at low $\bar{V}_{\text{traj}}$; the cell-level dispatch routes that cell to **SL-G** (Table 1).

Consistent with the dispatch routing. The deformation is diffuse but the junction-detected token is still the single most informative position to mix at: a sparse injection there sharpens the commitment and the trace exits the rambling loop with a correctly-formatted answer. Retry at T=0.6 stays in the same rambling basin; random-position injection rarely lands on the locally-strongest deformation under the same budget.

Repair@3 (repair depths): retry=✗ SL-G=✓ SL-R=✗ T_{loc}=1.5=✓

Figure 13: **Case C — Distributed Deformation routed to SL-G.** A complementary regime to Cases A and B above (Figure 11). Despite the most diffuse junction profile in the dataset (rank 0.005), the locally-strongest position still carries enough deformation signal for a single sparse injection to sharpen the commitment.

Setting	Value
Shared across all five operators	
Pre-injection temperature	0.6
Carrier token temperature	1.0
Post-injection decoding	greedy (T=0)
Repair attempts per (pid, operator)	K=10
Deep-sweep attempts (sensitivity)	$K \in \{16, 32, 50\}$
Max new tokens	2,048
Retry	
Position	none (full re-sample)
Logit modification	none
T_{loc} *(local temperature lift)*	
Position	detected junction
Temperature sweep	$\{1.0, 1.5, 2.0\}$
Canonical T_{loc} in main text	1.5
SL-G (sparse logit steering, geometric position)	
Position	detected junction
Mixing weight α	0.7
Window width w	1
Junction quantile q_J	0.99
Local-budget quantile q_V	0.75
Detector warm-up positions	20
Top-k for KL/Fisher approximation	20
Top-k for ancestor coverage	20
SL-R (sparse logit steering, random position)	
Position	uniform random (post warm-up)
Mixing weight α	0.7
Window width w	1
DL (dense logit steering)	
Position	every position in the trajectory
Mixing weight α	0.7

Table 16: Per-operator settings at repair time. All operators share the post-intervention sampling protocol: a single T=1.0 carrier token at the injection site and greedy decoding elsewhere in the window. The detector for SL-G uses the same window (w=1), warm-up, and top-k truncation as the Fisher-based junction selector defined in App. A.1; SL-R reuses these defaults but draws the injection position uniformly at random from valid positions past warm-up. DL applies the same α-blend at every token. Sensitivity to α across the rank-inverting operators is reported in App. H.

Table 17: Full-sequence token length distribution for Bespoke-Stratos-17k (Qwen3 chat template). The 0.6B and 1.7B specialists truncate at 8,192 tokens (P90); the 4B specialist at 16,384 (P99).

	Min	**P25**	**P50**	**P75**	**P90**	**P95**	**P99**
Full sequence	596	1,832	2,728	4,490	7,535	10,477	17,700
Assistant only	,	,	2,379	4,109	7,173	10,097	17,219

Specialist	**LR**	**BS/GPU**	**Grad accum**	**Optimizer**	**Warm-up**	**WD**	**Max seq.**	**Epochs / RL specifics**
Qwen3-0.6B SFT	7×10^{-5}	2	4	paged AdamW-8bit	3%	0.10	8,192	2 ep (save 352 steps)
Qwen3-1.7B SFT	4×10^{-5}	1	8	paged AdamW-8bit	3%	0.10	8,192	2 ep (SFT-704 / SFT-Final at step 704 / end)
Qwen3-4B SFT	2×10^{-5}	1	8	paged AdamW-8bit	3%	0.10	16,384	2 ep (save 176 steps)
Qwen3-1.7B GRPO	2×10^{-6}	1	8	paged AdamW-8bit	3%	0.10	4,096	2 ep; KL β=0.01; G=8; fmt-rwd 0.1; GRPO-4500 = step 4,500

Table 18: Actual training hyperparameters for the four post-trained specialists used in the main text, audited from the canonical training configs. All four runs share the same optimiser family (paged AdamW-8bit), Adam moments (β_1=0.9, β_2=0.95, ϵ=10^{-8}), gradient norm clip at 1.0, and Bespoke-Stratos-17k training data. SFT effective batch size (per-GPU BS $\times$ grad accum $\times$ n GPUs) is 16 for 0.6B (2 GPUs, L40S), 16 for 1.7B (2 GPUs, L40S), and 32 for 4B (4 GPUs, H100). GRPO uses G=8 generations per prompt sampled at the rollout temperature and adds the KL anchor against the ancestor logits; format reward weight 0.1 blends a structural "has-a-boxed-answer" bonus into the verifier-backed correctness reward. The 0.6B GRPO and 4B GRPO variants exist in the codebase but are not used in the main-text results.

Hyperparameter	**Value**
Optimizer	AdamW
Learning Rate	1e-5
Batch Size	128
Max Seq Length	2048
Weight Decay	0.1
LR Schedule	Cosine
Warmup Steps	100

Table 19: Legacy template values, retained only as a fall-back for any forward references; the actual per-specialist settings used in the paper are in Table 18.

Task	Model	N_{fail5}	Retry	Rand	Geo	Dense (T=0.0)
CruxEval	SFT 0.6B	103	0.104	0.155	0.175	0.388
	SFT 1.7B	97	0.073	0.526	0.454	0.278
	SFT 4B	169	0.104	0.574	0.505	0.462
	GRPO 1.7B	17	0.000	0.118	0.059	0.059
GSM8K	SFT 0.6B	13	0.364	0.154	0.385	0.154
	SFT 1.7B	13	0.364	0.000	0.154	0.154
	SFT 4B	13	0.333	0.462	0.308	0.308
	GRPO 1.7B	12	0.111	0.000	0.167	0.083
GPQA	SFT 0.6B	131	0.130	0.138	0.145	0.061
	SFT 1.7B	150	0.113	0.123	0.233	0.067
	SFT 4B	136	0.103	0.316	0.294	0.132
	GRPO 1.7B	144	0.069	0.194	0.132	0.069

Table 20: Mechanistic Profile evaluated uniformly at Fail@5. Greedy dense logit steering provides an upper bound on structural capacity without the aid of temperature sampling. Rescue rates are reported as fractions.

Round	Attempted	Rescued (this round)	Cumul. rescue rate
Round 1	1906	759	39.8%
Round 2	1147	188	49.7%
Round 3	959	44	52.0%
Unresolved	915	,	,

Table 21: **Round-by-round iterative repair results (CruxEval, SFT 1.7B).** Problems enter Round r only if all preceding rounds failed to repair the trajectory. Cumulative rescue rate is computed over all 1906 initially failing problems. The steep drop from Round 1 to Round 3 confirms that the vast majority of recoverable problems have a single dominant high-yield intervention point; multi-round repair provides diminishing but non-zero returns.

Specialist	Task	$\bar{J}_{\mathrm{frac}+}$	$\log_{10}\bar{C}$	$\log_{10}\bar{V}_{\mathrm{traj}}$	$\log_{10}\bar{V}_{t^\star}$
SFT 0.6B	CruxEval	0.72	0.63	0.69	0.26
	GSM8K	0.76	0.50	0.45	0.39
	GPQA	0.74	0.52	0.64	0.36
SFT 1.7B	CruxEval	0.84	0.76	0.72	0.37
	GSM8K	0.77	0.76	0.67	**0.09**
	GPQA	0.73	0.37	0.65	0.43
SFT 4B	CruxEval	0.45	0.61	0.42	0.25
	GSM8K	0.92	0.62	0.85	**0.11**
	GPQA	0.82	0.48	0.73	0.25
GRPO 1.7B	CruxEval	0.67	0.30	0.71	0.26
	GSM8K	0.83	0.38	0.85	0.24
	GPQA	0.70	0.19	0.64	0.20
Pooled (n=2,024 pids)		**0.95**	**0.84**	**0.89**	**0.48**

Table 22: **Within-pid intraclass correlation by feature and cell.** Pooled across all 12 cells (n=2,024 problem-units with ≥ 2 cached failed rollouts). $\bar{J}_{\mathrm{frac}+}$, $\log_{10}\bar{C}$, and $\log_{10}\bar{V}_{\mathrm{traj}}$ are within-pid concentrated (pooled ICC $0.95, 0.84, 0.89$); $\log_{10}\bar{V}_{t^\star}$ has pooled ICC 0.48. The ICC gap between the two V variants is diagnostic: $V_{t^\star}$'s within-pid variance reflects that the junction position $t^\star$ shifts between failed rollouts of the same problem, different failures expose different junction profiles, information the regime classifier consumes as part of the failure signature. $\bar{V}_{\mathrm{traj}}$ averages this variability out and serves the complementary role of the dispatch H/L routing key, where the per-pid decision needs a deterministic split.

Predictor	Buckets	Per-pid accuracy
Baseline (always predict retry)	1	0.505
Quadrant ($\log_{10}\bar{C}$, $\log_{10}\bar{V}_{\mathrm{traj}}$)	4	0.505
Logistic regression ($\log_{10}\bar{C}$, $\log_{10}\bar{V}_{\mathrm{traj}}$)	continuous	0.503
Logistic regression (3 features incl. $\bar{J}_{\mathrm{frac}+}$)	continuous	0.501
Cell (model $\times$ task)	8	**0.544**
Regime label	5	0.500
Regime $\times$ $\bar{V}_{\mathrm{HL}}$ (10 dispatch cells)	10	0.504

Table 23: **Operator-winner predictive accuracy by predictor.** Decisive-winner subset, n=507; gap threshold 5 pp. Only the cell-level predictor exceeds baseline by a meaningful margin ($+3.9$ pp). The three trajectory features, in either the median-split quadrant form or the continuous logistic-regression form, do not improve over always-predict-retry. The regime $\times$ $\bar{V}_{\mathrm{HL}}$ dispatch (10 buckets) achieves $+0.6$ pp at the per-pid level, consistent with the dispatch lift coming from *aggregate* cell-level preferences rather than from per-pid predictability.

Operator	n_{wins}	$\log_{10}\bar{C}$	$\log_{10}\bar{V}_{\mathrm{traj}}$	$\log_{10}\bar{V}_{t^\star}$	$\bar{J}_{\mathrm{frac}+}$
retry	256	**0.69**	0.62	0.63	0.62
SL-G	106	0.52	0.55	0.53	0.55
SL-R	64	0.61	0.60	0.63	0.59
T_{loc}=1.5	45	**0.67**	0.60	0.60	0.60
DL	36	0.77	0.78	0.71	**0.79**

Table 24: **Per-operator one-feature AUC.** For each operator, the AUC of a univariate logistic regression predicting "this operator is the decisive winner." Dense is the only operator that any single feature predicts at AUC > 0.75; the others sit near chance to mid-range from every feature. Dense's signal sits in $\bar{J}_{\mathrm{frac}+}$ (broad deformation $\Rightarrow$ DL wins), consistent with the regime semantics.

Detector	Ancestor	repair@3 rescue	N
Detector A	Qwen3-4B (lineage)	46.7% $(28/60)$	60
Detector A	Qwen2.5-1.5B (matched)	32.4% $(23/71)$	71
Detector A	Qwen2.5-7B (non-lineage)	25.4% $(18/71)$	71
junction-firing	Qwen3-4B (lineage)	$\mathbf{50.0}\%$ $(37/74)$	74
junction-firing	Qwen2.5-7B (non-lineage)	$\mathbf{50.6}\%$ $(39/77)$	77

Table 25: **Lineage advantage is detector-dependent.** repair@3 (any-of-three injection attempts per problem-unit) on the matched 77-pid intersection from the SFT 4B $\times$ CruxEval lineage ablation. Detector A (trace-relative 90th-percentile demotion score) shows a $+21.3$ pp lineage advantage; junction-firing (windowed $S_t > \lambda_J$ with per-problem calibration) reduces it to -0.6 pp (non-lineage numerically ahead by less than sampling noise). The published "lineage ancestor as structural reference" interpretation of the original 49.4% vs. 18.2% result no longer holds; we remove the lineage-privileging language throughout the body and treat ancestor choice as an engineering convenience rather than a structural requirement. The dispatch result in §3 is independent of this finding (both sides of $\Delta\Sigma S = +34.9$ were already under junction-firing). N varies by row because junction-firing requires sufficient conditioning-pool depth per pid; Detector A's three rows share an exact 3-way (pid, rid) intersection.

Regime	n	retry@1 [95% CI]	retry@3 [95% CI]
Unresolved	464	0.483 $[0.438, 0.528]$	0.651 $[0.608, 0.694]$
R.M. (junction-diffuse)	127	0.465 $[0.378, 0.551]$	0.646 $[0.559, 0.724]$
Distributed Deformation	523	0.423 $[0.380, 0.465]$	0.591 $[0.549, 0.633]$
R.M. (geo-local)	422	0.216 $[0.178, 0.258]$	0.396 $[0.348, 0.443]$
Total n (pids with all four repair operators)			1,536

Table 26: **Retry rescue rate stratified by regime, at one and three attempts.** retry@1 = deepest-K single rollout rescue; retry@3 = any-of-three deepest-K rollouts (the dispatch's operating point). Bootstrap 95% CIs over per-pid outcomes. Headline pairwise contrasts (Unresolved vs. RM-G): at retry@1, $\Delta = -26.7$ pp $[-33.4, -19.7]$; at retry@3, $\Delta = -25.5$ pp $[-31.8, -19.2]$. Both $p<0.0001$.

Table 27: **Per-attempt deployment cost by operator.** Cost is measured in single-rollout units (one specialist rollout at the spec generation budget of 1024 tokens = 1.00). SL-G and SL-R add an ancestor prefix-fill to the median junction-detector firing position $\hat{t}$; T_{loc} and retry use no ancestor; DL mixes at every token (full ancestor pass). Dispatch cost is the routed-fraction-weighted average under the V_{traj} routing (19.1% SL-G + 70.4% SL-R + 9.6% T_{loc} + 0.8% retry). At repair@K, multiply by K.

Operator	Median $\hat{t}$	Cost	Breakdown
retry	(fresh sample)	1.00	single specialist rollout
T_{loc}=1.5	no ancestor	1.00	specialist rollout at lifted T at junction
SL-G	$\hat{t}$=41 (of 1024)	1.04	1.0 specialist + $\frac{41}{1024}$=0.04 ancestor prefix
SL-R	$\hat{t}$=488 (of 1024)	1.48	1.0 specialist + $\frac{488}{1024}$=0.48 ancestor prefix
DL	every token	2.00	1.0 specialist + 1.0 ancestor (full-trace mixing)
dispatch (V_{traj} routed)	weighted	**1.34**	$0.191\times1.04 + 0.704\times1.48 + 0.096\times1.00 + 0.008\times1.00$

Cell	n	retry@10 (10 units)	retry@32 (32 units)	retry@64 (64 units)	dispatch@3 (4.0 units)	iso-K retry	compute eff. disp/retry@64
sft-1.7B $\times$ CruxEval	220	0.654	0.754	0.777	0.650	$K\approx10$	$\sim 13\times$
GRPO-1.7B $\times$ CruxEval	225	0.658	0.747	0.769	0.707	$K\approx13$	$\sim 15\times$

Table 28: **Retry@K curves vs. dispatch@3 on the same pid set.** Restricted to the dispatch's Fail@K=10 set intersected with the K=64 retry pool. T=0.8 for the K=64 pool vs. T=0.6 for the dispatch (more permissive baseline for retry). Compute measured in single-rollout units (one specialist rollout = 1.0). Dispatch@3 cost = 3 attempts $\times$ 1.34 units/attempt ≈ 4.0; retry@K cost = K units. *iso-K retry* is the smallest retry attempt count whose rescue rate $\geq$ dispatch@3's rescue rate. *compute eff.* = (dispatch rescue / 4.0) $\div$ (retry@64 rescue / 64). retry@K columns are bootstrap means over per-pid any-of-rollouts[0..K-1] indicators. We have K=64 data only for these two cells; the dispatch claim itself covers ten cells.

Regime (sft-1.7B CruxEval)	n	retry@64	dispatch@3	Δ
Unresolved	173	0.769	0.671	+0.098
R.M. (junction-diffuse)	29	0.862	0.448	+0.414
R.M. (geo-local)	17	0.706	0.529	+0.177
Distributed Deformation	1	1.000	0.000	+1.000

Table 29: **Per-regime retry@64 vs. dispatch@3 on sft-1.7B CruxEval.** R.M. (geo-local) absolute retry catch-up at K=64, T=0.8: $\Delta = +7.7$ pp. The absolute rank-locked claim does not hold at this budget; the per-FLOP efficiency claim (Table 28) does.

Regime	V_t	n_{pair}	SL-G R	SL-R R	gap (pp)	bucket
DD	H	116	0.414±0.046	0.379±0.045	-3.45±4.71	**SL-G decisive**
DD	L	417	0.657±0.023	0.671±0.023	+1.44±1.89	methodological
RM-G	H	431	0.339±0.023	0.369±0.023	+3.02±2.63	mixed
RM-G	L	1†	1.000±0.000	1.000±0.000	+0.00±0.00	methodological
Unresolved	H	94	0.628±0.050	0.638±0.050	+1.06±4.38	methodological
Unresolved	L	171	0.684±0.036	0.661±0.036	-2.34±2.98	methodological
RM-D	H	28†	0.714±0.085	0.679±0.088	-3.57±7.96	**SL-G decisive**
RM-D	L	119	0.639±0.044	0.639±0.044	+0.00±3.94	methodological
Unresolved	H	84	0.857±0.038	0.857±0.038	+0.00±2.92	methodological
Unresolved	L	52	0.577±0.069	0.615±0.067	+3.85±6.64	mixed

Table 30: **Paired SL-G vs. SL-R rescue per (regime, V_t) cell.** Restricted to problem-units with outcomes available for both operators (apples-to-apples). gap $= R(\text{SL-R}) - R(\text{SL-G})$ in pp. Bucket assignment: *mechanistic* if gap $> +8$ pp; mixed if $+3 <$ gap $\leq +8$; methodological if $|\text{gap}| \leq 3$; SL-G decisive if gap < -3. Bucket totals (cells with $n \geq 30$): mechanistic: 0 / methodological: 885 / mixed: 483 / SL-G-decisive: 116. Excluded 29 pids in †-marked cells.

Metric	Value
n SL-R-only attempts	444
Mean $\lvert t^{\star}_{\text{SL-G}} - t_{\text{rand}} \rvert$ (tokens)	459.6
Median (tokens)	452
Std (tokens)	282.4
Mean fractional gap (over n_{tokens})	0.241
Share with gap < 32 tokens ("near SL-G")	4.7%
Null (rand-position shuffled across attempts)	4.4%
Enrichment ratio (observed / null)	$1.06\times$
Crude bimodality ratio ($(\mu_{\text{hi}} - \mu_{\text{lo}})/\sigma$)	1.64

Table 31: SL-R-only position-gap audit (444 SL-R-winning attempts). Within-32-token share matches a position-shuffled null (observed 4.7% vs null 4.4%, $1.06\times$ enrichment); the bimodality ratio 1.64 is below the > 2 threshold for a clean two-mode distribution.

Task	Model	K	N	Retry	[95% CI]	Rand	[95% CI]	Geo	[95% CI]
CruxEval	SFT 4B	5	169	10.4%	[2.1, 20.8]	57.4%	[48.5, 66.3]	50.5%	[40.6, 59.4]
CruxEval	SFT 1.7B	5	97	7.3%	[0.0, 17.1]	52.6%	[43.3, 61.9]	45.4%	[36.1, 55.7]
GPQA	SFT 1.7B	1	180	15.6%	[11.1, 21.1]	15.2%	[9.9, 20.5]	28.9%	[22.2, 35.6]
GPQA	SFT 1.7B	5	150	11.3%	[6.7, 16.7]	12.3%	[7.5, 17.8]	23.3%	[16.7, 30.0]
GSM8K	SFT 4B	5	13	33.3%	[0.0, 66.7]	46.2%	[23.1, 69.4]	30.8%	[7.7, 53.8]

Table 32: **Bootstrap 95% CIs for headline Fail@K cells (10,000 resamples).** All methods evaluated on matched PID sets; bootstrap is applied at the problem level. Retry for CruxEval and GSM8K uses the global retry pool; for GPQA, the already-failed rollout pool indexed at K:K+3. GSM8K SFT 4B (N=13) CIs are too wide for directional claims.

Ancestor	Rescue Rate	95% CI
Lineage (Qwen3 base)	49.4%	[38.2, 60.6]
Qwen2.5-1.5B (non-lin.)	27.3%	[17.3, 37.3]
Qwen2.5-7B (non-lin.)	18.2%	[9.6, 26.8]

Table 33: **Detector-A ancestor identity ablation (SFT 4B, CruxEval, N=77), superseded; see App. R.** Binomial Wilson 95% CIs. The non-overlapping CIs were the load-bearing evidence for the (now retracted) lineage-specificity claim. Under the junction detector on the same 77 pids, the lineage–Qwen2.5-7B gap is -0.6 pp (no lineage advantage). The result is detector-specific, not a paper-level finding.

Case	Regime	$J_{\mathrm{frac}+}$	Junction rank	Cell-level dispatch
A. CruxEval, sft-1.7B	Rank Misrouting (junction-diffuse)	0.38	0.92 (sharp early spike)	T_{loc}=1.5
B. GPQA, sft-1.7B	Rank Misrouting (geo-local)	0.62	0.81 (sharp pivot)	sparse logit-steer
C. CruxEval, sft-4B	Distributed Deformation	0.53	0.005 (very diffuse)	sparse logit-steer

Table 34: **Where each case lands in the regime space under $\bar{V}_{\mathrm{traj}}$-based dispatch.** Cases A and B (this figure) and Case C (Appendix Y). The three features ($J_{\mathrm{frac}+}$, $J_{\mathrm{max}}/J_{\mathrm{mean}}$ rank, $\bar{V}_{\mathrm{traj}}$) place each case in a different regime/cell, and the cell-level dispatch routes the regime to a different operator class. The features predict the *regime* (and the cell-level dispatch operator class), not the per-pid outcome; aggregate dispatch lifts (§3) are at the population scale.