CORRECT REWARD, COSTLY RULE: GRADIENT COHERENCE DECIDES WHAT GROUP-RELATIVE RL LOCKS IN

Anonymous authors
Paper under double-blind review

ABSTRACT

When a verifiable reward is exact but the policy's observation is insufficient, the reward-maximising policy must follow a feature that is right only on average, and its cost falls entirely on the samples that feature misleads. We study how group-relative RL reaches this outcome, using next-POI ranking as a controlled testbed. First, GRPO converges to the constrained-optimal rule: it locks onto the most readable feature, lifts aggregate accuracy, pushes the misled subgroup below chance, and keeps applying the rule where the feature is uninformative. Second, the cost is asymmetric because aligned gradients cohere while misaligned ones cancel, not because either side is larger. Third, an exact advantage–position identity shows that this coherence arises from centring the advantage, not inherited from the model's representations. Fourth, the resulting criterion predicts which feature is locked in before training, even against the base model's preference. Finally, re-anchoring the advantage moves the cost as the diagnosis implies, and bounds what any reweighting of sampled rollouts can recover.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR), notably Group Relative Policy Optimization (GRPO) (Shao et al., 2024; DeepSeek-AI, 2025), trains large language models against rewards that check the outcome exactly, and is increasingly applied beyond mathematics to tasks such as ranking and recommendation (Li et al., 2025b; Zhu et al., 2026). In these tasks the answer can be verified, but the input rarely determines it: a recommender never observes a user's full intent, only cues such as how popular each item is. An exact reward does not make this observation sufficient. When the options differ only in a feature that predicts the correct one on average but not case by case, the reward-maximising policy must follow that feature everywhere. It helps the *aligned* samples, where the feature points to the answer, and pushes the *misaligned* ones—in recommendation, the long tail of less popular items and the users who choose them—below chance, while aggregate accuracy rises and hides the cost. A practitioner who monitors aggregate reward sees training succeed. We study this regime in next point-of-interest (POI) recommendation (Li et al., 2024), where the policy ranks candidate venues, the reward scores exactly where the true next visit lands, and we can control which features the candidates expose.

Existing accounts of RL failure blame a misspecified reward or a distribution shift (§2); neither applies here, since the reward is exact and about half of the training samples already contradict the rule. We ask whether GRPO converges to this rule, why its cost falls on one subgroup, which feature it locks onto, and whether the cost can be moved. GRPO installs the constrained-optimal rule within 200 steps (§3.1), moves it to whichever feature is most readable, and keeps applying it where that feature is uninformative (§3.2). The cost is asymmetric because aligned gradients cohere while misaligned ones cancel (§3.3), and an exact advantage–position identity traces this coherence to the advantage rather than to the model's representations (§3.4). The resulting criterion predicts the locked feature before training, against the base model's preference (§3.5), and re-anchoring the advantage moves the cost as the diagnosis implies (§4).

Our contributions are threefold. First, we show that under an exact reward but insufficient observation, group-relative RL installs the constrained-optimal rule, whose cost falls on one subgroup and spills into regions where the rule is uninformative. Second, we explain the asymmetry through gra-

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

dient coherence and an exact identity, yielding a readability criterion that predicts the locked feature before training. Third, we use re-anchoring as an interventional test that confirms the diagnosis and bounds what any reweighting of sampled rollouts can achieve. For practitioners, these results turn a hidden failure into a checkable one: evaluate by subgroup rather than in aggregate, treat a rising fraction of zero-advantage groups as a sign that a rule has locked in, and measure which feature the reward will lock onto before paying for training.

2 Related Work

Reward hacking has been studied from early safety work (Amodei et al., 2016) through empirical maps of misspecified rewards (Pan et al., 2022) to correlated proxies whose link to the true reward breaks under optimisation (Laidlaw et al., 2025); all assume that the reward differs from the objective. Work on spurious features and goal misgeneralization instead traces failure to the data (Nagarajan et al., 2021; Langosco et al., 2022), where a little decorrelating data can help: randomising the goal in only 2% of CoinRun levels markedly reduces misgeneralization (Langosco et al., 2022). Our setting fits neither. The reward is exact, about half of the training samples already contradict the rule, and following the rule is optimal under the available observation (Theorem 1); which feature it follows is decided by gradient coherence, echoing the coherent-gradients hypothesis (Chatterjee, 2020) and the preference of networks for more available over more predictive features (Hermann et al., 2024); we show that in group-relative RL this coherence arises from the centred advantage, which is what lets us predict the feature before training (Appendix A.1).

GRPO replaced the learned critic with a group-mean baseline (Shao et al., 2024; DeepSeek-AI, 2025), and follow-up work has exposed biases of this estimator: difficulty weighting from standard-deviation normalisation (Li et al., 2025a), zero-advantage groups that dynamic sampling discards (Yu et al., 2025), interference between reward components (Ichihara et al., 2025), and, concurrently, advantage magnitudes that reward lucky guesses on bounded-answer tasks (Wang et al., 2026). These analyses concern how prompts, components or rollouts are weighted; we ask which subgroup of prompts the aggregate update serves under a single exact reward, and locate the asymmetry in the direction of the update rather than its magnitude.

Popularity bias in recommendation is usually attributed to feedback loops with logged data (Mansoury et al., 2020; Chen et al., 2023). As RL is applied to LLM-based recommenders (Li et al., 2025b; Zhu et al., 2026), zero-advantage groups have been treated as missed hits to convert into preference pairs (Ding et al., 2026). We show that the optimiser itself can manufacture a popularity rule from a correct reward, and that its silent groups stem from lock-in rather than from misses.

3 Experiments

3.1 The constrained optimum yields a sharp rule

Problem formulation. Each prompt presents n candidates ($n = 8$ in our experiments) in a randomised order, each under a letter label (A–H), and each described by $m \geq 1$ observable features. The model reads the prompt and outputs a complete ranking of the n candidates; throughout, *position* refers to a candidate's place in this output ranking. The reward depends on the position of the *target* candidate (the ground-truth next visit). We first study the *Vanilla* construction, in which candidates carry a single informative feature (m=1). §3.2 introduces constructions where a second feature becomes observable (m=2), and the general m-feature formulation is in Appendix A.3. Vanilla is built on the Foursquare NYC check-in dataset. All candidates share the target's category, so they carry no distinguishing information other than their historical visit counts (we verify the absence of string leakage in Appendix A.5); the only usable signal is the *relative ordering* of these counts. Let the count-rank $k \in \{1, \ldots, n\}$ denote a candidate's place in this

Table 1: Vanilla by Quadrant: Hit Rate@1 and Count-Rule Follow rate per subgroup. *Count-Rule Follow*: the model's pick has the highest historical visit count among the 8 candidates (ties included). fv/rv: first visit (user has never visited this POI) / revisit (user has visited before); pop/nonpop: popular/ non-popular subgroup.

Subgroup	Base HR@1	GRPO HR@1	Base Count-Rule Follow	GRPO Count-Rule Follow
fv + nonpop	0.049	0.027	0.815	0.905
fv + pop	0.777	0.868	0.832	0.924
rv + nonpop	0.121	0.067	0.775	0.882
rv + pop	0.804	0.910	0.817	0.925

ordering ($k = 1$: highest count; candidates tied at the maximum all receive $k = 1$). Under this constraint the model's strategy reduces to a mapping from count-ranks to positions, i.e. a permutation $\tau : k \mapsto \tau(k)$. Training uses a position-based reward that decreases linearly with the target's position:[1]

$$R(\tau) = \frac{n - \tau(k_{\text{target}})}{n - 1} \tag{1}$$

where k_{target} is the count-rank of the target. We train with GRPO, sampling $G{=}5$ rollouts per prompt. $R{=}1$ when the target is placed first; $R{=}0$ when last. Let $q_k = P(\text{target has count-rank } k)$. The expected reward under strategy τ is $\mathbb{E}[R \mid \tau] = \frac{n}{n-1} - \frac{1}{n-1}\sum_{k=1}^{n} q_k\,\tau(k)$, so maximising expected reward is equivalent to minimising $\sum_k q_k\,\tau(k)$. The following theorem characterizes the optimal policy under this condition.

Theorem 1 (Rearrangement optimality)**.** *If $q_1 > q_2 > \cdots > q_n$, then $\tau^*(k) = k$—ranking candidates in strict descending order of count—is the unique expected-reward-maximising permutation.*

The proof (Appendix A.2) is a direct application of the rearrangement inequality. Since (q_k) is strictly decreasing, pairing it with the ascending identity $\tau(k) = k$ uniquely minimises $\sum q_k\,\tau(k)$. The strict decrease holds empirically under every condition we construct, as well as on the training split (Appendix A.6). The theorem therefore partitions every sample into two subgroups. **Aligned** samples ($k_{\text{target}}{=}1$) are those where the target happens to have the highest count, so the optimal strategy places it first; in the recommendation setting these correspond to *popular* items. **Misaligned** samples ($k_{\text{target}}{>}1$) are those where the target does not have the highest count, so the optimal strategy does not place it first; these correspond to *non-popular* items.

We first examine the Vanilla experiment on the Foursquare NYC dataset. Consistent with the theorem, aggregate HR@1 increases from 0.4591 to 0.4953 after 200 steps of GRPO training. But the improvement is not shared: on the misaligned subgroup, HR@1 falls below the base model's (Table 1) and further below chance. The errors are not diffuse: among picks that miss the count maximum, 98–99% land on count-ranks 2 and 3. Training therefore does not merely increase reliance on count; it sharpens the base model's dispersed probability mass into a near-deterministic count rule. We call the installation of such a rule **lock-in**. The rule helps where the count ranking points to the target and errs systematically where it does not—a subgroup-asymmetric cost hidden by the aggregate improvement. Theorem 1 fixes where training ends up under a single feature; it does not say how the optimiser gets there, which feature it locks onto when several are readable, or why the rule is applied where it is no longer optimal. In particular, Vanilla cannot tell whether training relies on count because it is the only readable feature, or because the optimiser would prefer it even when alternatives exist.

3.2 The rule is applied even where its signal is absent

To answer this, we introduce *category* as a second observable feature ($m{=}2$) and build two constructions that change only the negative sampling, holding the user set, sample count and time window fixed:

- **Construction A**: Negative candidates are drawn from categories different from the target's. Therefore, the target is the only candidate in its category, making *category* a second readable feature alongside *count*. Because the negative pool is no longer restricted to same-category hard negatives, the discriminative power of *count* is preserved or even enhanced (Appendix A.7).
- **Construction B**: On top of cross-category sampling, negatives are further restricted to POIs whose visit counts fall within the same quantile bin as the target's. This substantially reduces the discriminative power of *count*, leaving *category* as the dominant signal.

The two constructions separate the explanations by asking whether a learned feature-following rule persists when its feature is no longer informative at test time. We define a signal-presence indicator for each test sample. Let $W{=}1$ if the target's category appeared in the user's 5-item visit history window (the model can read a category signal from the context), and $W{=}0$ if it did not (no trace of

[1] The actual reward also includes a format-compliance term and a temporal bonus. We verify in Appendix A.4 that (i) the format term does not affect candidate ordering, and (ii) the bonus reads only the target's own attributes and does not change the conclusion that placing the target first is uniquely optimal.

Table 2: W-stratified lock-in: each column is base model → GRPO ckpt200.
Count-Rule Follow/Cat-Rule Follow: the model's pick coincides with the corresponding rule's admissible set (ties included); *Cat-Window Match*: the pick's category appears in the 5-check-in window at all (a weaker agreement than Cat-Rule Follow).

	W	HR@1	Count-Rule Follow	Cat-Rule Follow	Cat-Window Match
A	1	$0.517 \rightarrow 0.774^{\dagger}$	$0.568 \rightarrow 0.776$	$0.541 \rightarrow 0.787$	$0.569 \rightarrow 0.823$
	0	$0.388 \rightarrow 0.424$	$0.573 \rightarrow 0.730$	—	$0.070 \rightarrow 0.106$
B	1	$0.362 \rightarrow 0.637^{\dagger}$	$0.506 \rightarrow 0.518$	$0.435 \rightarrow 0.747$	$0.467 \rightarrow 0.791$
	0	$0.212 \rightarrow 0.187 \downarrow$	$0.601 \rightarrow 0.618$	—	$0.119 \rightarrow 0.249$

† exceeds the Count-Rule Ceiling (0.664 for A, 0.299 for B), i.e., unattainable by any count-only strategy;
$\downarrow$ negative spillover: below the base model in the $W{=}0$ region, where neither signal is readable.

the target's category exists in context). [2] By design, $W{=}1$ samples carry the category signal, while $W{=}0$ samples form a *signal-absent region* in which the category rule provides no target-directed information — where it fires at all, it points to a distractor. We measure a rule's usefulness by its *ceiling*, the expected HR@1 of a perfect follower (Appendix A.7).

Construction A: count lock-in persists, and category is learned alongside it. Adding category does not prevent count lock-in where count is the only readable signal: at $W{=}0$ the Vanilla signature recurs, with count-rule following and HR@1 both rising and the misaligned subgroups paying the same cost (Table 2; Appendix A.8). At $W{=}1$ both signals are readable and the model exploits both: HR@1 rises to 0.774, above the 0.664 ceiling of any count-only strategy, while Cat-Rule Follow rises from 0.541 to 0.787, closing 53% of the available headroom.

Construction B, $W{=}1$: lock-in transfers to a new feature. With count suppressed, category becomes the dominant readable signal at $W{=}1$. The base model's HR@1 (0.362, Table 2) already exceeds the Count-Rule Ceiling (0.299), confirming that category is legible before any training. GRPO installs a category-following rule of the same type and comparable magnitude as Vanilla's *count* rule: Cat-Rule Follow rises from 0.435 to 0.747 (55% of the headroom, comparable to A's 53%), and HR@1 rises to 0.637 (Table 2), approaching the Cat-Rule Ceiling (0.746). Errors again concentrate on the next-ranked alternatives, the same sharp-rule signature on a different feature (Appendix A.8).

Construction B, $W{=}0$: spillover. In the signal-absent region, HR@1 falls *below* the base model ($0.212 \rightarrow 0.187$). This is **spillover**: a rule learned where its feature is informative keeps being applied where the feature carries no information about the target. The behavioural trace is direct: the fraction of picks whose category appears in the recent window—the cue learned on $W{=}1$—doubles ($0.119 \rightarrow 0.249$), and because $W{=}0$ targets by definition have a category absent from that window, every such pick is wrong (alternative explanations are ruled out in Appendix A.8; the rule also overrides count where count alone would have sufficed, Appendix A.9).

Construction A shows what spillover does and does not depend on. At $W{=}0$ the category cue is uninformative in both constructions, and in both the category habit leaks into this region: Cat-Window Match rises in A as well ($0.070 \rightarrow 0.106$), though less than in B. What differs is the competing signal. In A, count remains informative (Count-Rule Ceiling 0.484): count-following strengthens ($0.573 \rightarrow 0.730$) and HR@1 still rises ($0.388 \rightarrow 0.424$), so the leaked habit is outweighed. In B, count is suppressed (ceiling 0.229): count-following barely moves ($0.601 \rightarrow 0.618$), nothing offsets the leak, and HR@1 falls below base. Spillover is thus a property of the learned rule itself—applied unconditionally in both constructions—while the strength of the competing signal decides whether it surfaces as a net loss.

Lock-in follows readability, not count. Both the $W{=}1$ lock-in and the $W{=}0$ spillover reach near-saturation within 200 training steps and change little over the subsequent 200 steps (Appendix A.10), suggesting that the rule is installed early and then largely frozen. Across constructions, the locked feature changes with which signal is most *readable* to the optimisation process: Vanilla locks onto *count*, while Construction B locks onto category. Yet the resulting behaviour is invariant: a near-deterministic rule is installed, the aligned subgroup benefits, the misaligned subgroup pays, and the rule is applied beyond the region where its feature is informative. This answers the question of §3.1:

[2] W is a *test-time annotation*: training uses all samples regardless of W.

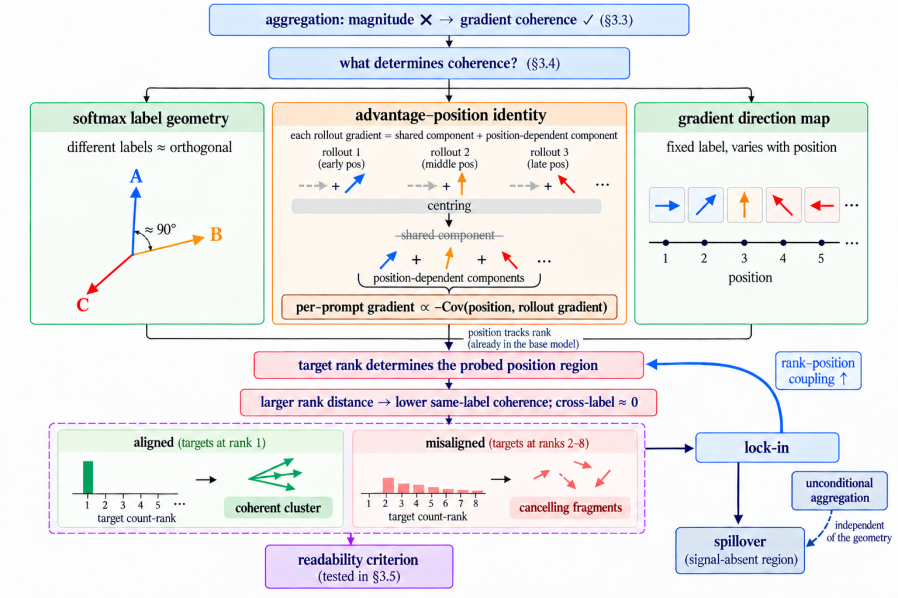

Figure 1: **How coherence is determined**. Different labels give near-orthogonal gradients; within a label, centring the advantage cancels the shared component and leaves each prompt's gradient as a probe of the direction map at the position its target's count-rank induces. Aligned targets sit at rank 1 and form one coherent cluster; misaligned targets scatter over ranks 2–8 and cancel. Lock-in tightens the position–rank coupling, closing a feedback loop. The criterion extracted from this geometry is tested in §3.5; spillover follows separately from the unconditional sum over samples.

count has no intrinsic privilege. Two questions remain: why the cost of the installed rule falls on the misaligned subgroup, and what makes a feature readable. We answer the first in §3.3 and the second in §3.4, by analysing the training gradient.

3.3 Why the cost is asymmetric: gradient coherence, not magnitude

Why does the aligned subgroup's signal win the aggregate update while the misaligned subgroup's is lost? A natural explanation is *magnitude dominance*: aligned groups might simply produce larger gradients. We find no such effect: per-group gradient norms are comparable across the two subgroups at every checkpoint we probe (steps 200/400/500; Appendix A.13). The asymmetry lies instead in *gradient coherence*, the degree to which the gradients within a subgroup point in the same direction. For N gradients of equal norm ρ with mean pairwise cosine $\bar{c}$:

$$\left\|\sum_{i=1}^{N} g_i\right\| = N\rho\sqrt{\bar{c} + \frac{1-\bar{c}}{N}} \qquad (2)$$

so the aggregation efficiency $\eta = \|\sum_i g_i\| / \sum_i \|g_i\|$ falls to the random-direction floor $1/\sqrt{N}$ when $\bar{c} = 0$: strong individual gradients need not produce a strong aggregate update. Measured over the groups that contribute gradient, the misaligned subgroup sits at its floor ($\eta \approx 0.06$ over $N \approx 300$ groups; some 94% of its signal cancels), while the aligned subgroup reaches about twice its own ($\eta \approx 0.18$–0.21 over $N \approx 100$). Directly measured, $\bar{c} \approx 0.03$ versus ≈ 0—the scale of the same-rank pairwise cosines of §3.4. Although 80–85% of aligned groups contribute no gradient at all and the misaligned side contributes about three times as many groups, the aligned aggregate remains the larger; and between steps 200 and 400 the two aggregates turn from positively to negatively aligned, pulling the model in opposite directions. The asymmetric cost is thus driven by the alignment of gradient directions, not by the strength of individual signals.

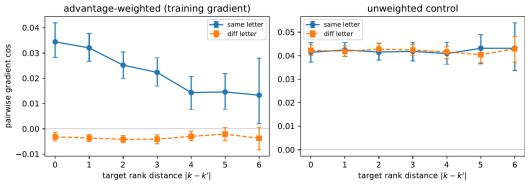

Figure 2: Pairwise gradient cosine by target rank distance after GRPO 400 steps, advantage-weighted (training gradient, left) vs. unweighted control on identical rollouts (right). The same-label decay and the dependence on label match exist only under the advantage weighting: the position-varying direction is present throughout, but only the centred advantage cancels the shared component that buries it.

A self-limiting dynamic. The same picture explains why lock-in and spillover saturate within 200 steps and then freeze (§3.2). Because the aligned subgroup's gradients cohere, the rule is installed quickly. As the rule becomes near-deterministic, the rollouts of a prompt increasingly place the target at the same position and receive the same reward; the group-relative advantages then vanish and the group contributes no gradient. We call such groups **silent**. The silent fraction on the misaligned subgroup rises sharply ($0.095 \rightarrow 0.460$) over exactly the 200-step window in which lock-in and spillover saturate, then stabilises (Appendix A.11). The rule thus installs itself fast and then removes its own fuel—both the signal that would reinforce it and the signal that could reverse it—which is why the $W = 1$ lock-in and the $W = 0$ spillover plateau rather than drift.

3.4 Where the asymmetry comes from: the count-rank geometry

Section 3.3 shows that coherence, not magnitude, decides which subgroup's signal survives aggregation. We now ask what *determines* coherence; the answer also settles what makes a feature *readable* (§3.2), since readability can only act through coherence. In short: gradient coherence follows the count-rank distance between targets, and this geometry is carried by the advantage weighting, not by the representations. Figure 1 summarises the resulting chain in mechanism order; the text below presents the evidence first.

Coherence decays with count-rank distance. Gradients of rollouts that rank *different* labels first have nearly orthogonal label-specific components, a known property of the softmax output layer (Appendix A.12); what they share is a label-independent component, which appears below as a uniform floor. We therefore group per-prompt gradients by the label of the *correct* candidate, on which the reward acts; cross-label pairs are indeed near-orthogonal (Figure 2, left). Within same-label pairs, cosine similarity **decays with the count-rank distance between their targets** and plateaus beyond distance ≈ 4. Because per-prompt gradients are dominated by sample-specific components, we read the cosines relative to the cross-seed ceiling for identical prompts: same-rank pairs attain 14% of it and distance-6 pairs 5% (Cohen's $d = 0.285$, replicated across two seeds). Replacing the letter labels with digit strings preserves the decay (Appendix A.17).

The geometry is carried by the advantage weighting. Neither prompt- nor candidate-level representations encode rank distance (the former is flat, $R^2 < 0.01$; the latter registers count ties, not rank separation; Appendix A.14), so the geometry is not inherited from the input. It comes from the advantage weighting instead: recomputing the analysis on the *same* rollouts with the advantage set to 1 removes both the decay and the label dependence (Figure 2 right). What remains is a uniform floor (≈ 0.042), a component shared by all completion gradients that amounts to 72% of the unweighted ceiling (0.058) and buries any position-dependent variation. The following identity shows that the centred advantage cancels exactly this shared component, exposing how gradient directions vary with position.

Proposition 1 (Advantage–position identity)**.** *Consider the G rollouts $o_1, \ldots, o_G$ of a prompt x under on-policy group-relative optimisation (importance ratio 1, clipping inactive, KL term excluded). Let $w_i := \nabla_\theta \log \pi_\theta(o_i \mid x)$ and write rollout i's reward as $r_i = r_0 - c\,\mathrm{pos}_i + b_i$, where pos_i is the position it assigns to the correct candidate, $c = 1/(n-1)$ by equation 1, and b_i is a bounded position-independent residual. With $\hat{A}_i = (r_i - \bar{r})/\sigma$, where σ is the within-group reward standard deviation, the per-prompt gradient $g = \sum_i \hat{A}_i w_i$ decomposes exactly as*

$$g \;=\; \frac{c}{\sigma}\, g_{\mathrm{pos}} + \frac{1}{\sigma}\, g_b, \qquad g_{\mathrm{pos}} := \sum_{i=1}^{G} (\bar{p} - \mathrm{pos}_i)\, w_i = -G\, \widehat{\mathrm{Cov}}(\mathrm{pos}, w), \qquad g_b := \sum_{i=1}^{G} (b_i - \bar{b})\, w_i. \tag{3}$$

Any component of the rollout directions constant within the group cancels exactly from both terms.

Full conditions and the proof are in Appendix A.15. When all rewards in a group coincide, every coefficient vanishes and $g = 0$: the silent groups of §3.3 are the zero-variance case of the identity.

How the decay arises. The identity reduces coherence to the within-group covariance between *where a rollout places the correct candidate* and *how its gradient direction varies with that position*. Two measured facts complete the account (Appendix A.14). First, position tracks count-rank: the Spearman coupling is 0.67 already in the base model and tightens to 0.85 by ckpt-400, so prompts at rank k probe the direction map around position k. Second, the position-varying part of the direction is label-specific. As $|k - k'|$ grows, the probed regions—and with them the gradient directions—

rotate apart, and same-label pairs decorrelate. Plugging the measured position histograms and mean directions $\mathbb{E}[w \mid \text{label}, \text{pos}]$ into equation 3 reproduces the shape of the measured same-label curve, including its plateau (Spearman 0.94 over six distance bins), and the agreement holds out of sample under a split-half control (Appendix A.16).

Resolving the asymmetry. The aligned subgroup's targets sit at rank 1 by construction, and the base policy's popularity prior already places the correct candidate first on 66% of aligned rollouts[3]: from step 0 their gradients probe the same region of the direction map and form a single coherent cluster. The misaligned subgroup's targets are scattered across ranks 2 through 8, so each rank probes its own region and the aggregate fragments into mutually cancelling clusters. Training then tightens the position–rank coupling that separates these fragments ($0.67 \rightarrow 0.79 \rightarrow 0.85$ at ckpt-0/200/400): the asymmetry seeds the lock, and the lock sharpens the asymmetry. Coherence itself is measured before training in §3.5.

From geometry to readability. A feature is readable to group-relative optimisation to the extent that its aligned targets are *concentrated* in this geometry: concentration produces one coherent cluster, dispersion produces cancelling fragments. The criterion is consistent with the cross-construction pattern of §3.2, and it makes a direct prediction—coherence should pick the winner even when the base prior favours the loser—which we test in §3.5.

Why the rule spills over. Independently of the geometry, the update sums gradients over all samples and nothing conditions the rule on whether its feature is informative, so the winning feature is installed as an *unconditional* policy—which is why it leaks into the signal-absent region in both constructions (§3.2).

3.5 A PREDICTION TEST: COHERENCE AGAINST THE PRIOR

The readability criterion of §3.4 predicts that where two readable features compete and the base model's prior favours the *loser*, coherence should still pick the winner. We test this in a second city, chosen so that the prior opposes coherence while objective strength is nearly tied.

A second city. In Tokyo, Construction B (TKY-B) leaves the two rules' ceilings nearly tied on the $W{=}1$ layer, and the base model mildly prefers count (Table 3). Coherence predicts category nonetheless: measured on the base model, before any training, on prompts where the two rules disagree, category-rule gradients are $5.3\times$ more coherent than count-rule gradients. Training confirms the prediction: after 200 steps, following of the category rule rises well beyond that of the count rule, including where the category rule singles out a unique candidate (Appendix A.18); $W{=}1$ HR@1 improves ($0.550 \rightarrow 0.761$) while $W{=}0$ HR@1 falls below base ($0.226 \rightarrow 0.191$)—the spillover cost travels with the lock. The lock target thus contradicts the prior and follows coherence; objective strength leans the same way, but by $1.09\times$ against a $5.3\times$ coherence gap.

Table 3: What predicts the lock target? TKY-B, $W{=}1$ layer. *Ceiling*: expected HR@1 of a perfect rule follower (Appendix A.7). *Follow*: fraction of prompts whose top-1 pick lies in the rule's admissible set (ties included), before and after training. *Coherence*: mean pairwise cosine of advantage-weighted gradients on the base model, on prompts where the two rules disagree (user-block permutation $p < 10^{-4}$).

	Rule Ceiling	Rule Follow on Base Model	Coherence	Rule Follow after GRPO-200
Count rule	0.622	0.668	0.0048 (1.0×)	0.775
Category rule	0.680	0.643	0.0252 (5.3×)	0.879

Robustness across backbone. Repeating NYC-B with Llama-3.1-8B reproduces the category lock (Cat-Rule Follow $0.526 \rightarrow 0.829$ at $W{=}1$) and the spillover cost (Appendix A.19). Across city and backbone, the most coherent readable feature is installed as an unconditional rule.

4 TESTING THE DIAGNOSIS: RE-ANCHORING THE ADVANTAGE

4.1 HEURISTIC-RELATIVE ADVANTAGE

Sections 3.3–3.4 locate the asymmetric cost in the advantage. With the group mean as the centre, a rollout is reinforced for outranking its peers, not for meeting any external standard: once most

[3]Measured as the gradient sees them: training-pool prompts sampled as in training ($T{=}0.9$) and scored by the training reward's parser. Greedy decoding with standard answer extraction gives the 0.78–0.80 of Table 1.

Table 4: Count lock (NYC Vanilla, full test set). HR@1 overall and by quadrant; Count-Rule Follow on the non-popular quadrants; silent: fraction of zero-advantage groups on a 600-prompt non-popular probe (Appendix A.11).

		HR@1 by quadrant				Count-Rule Follow (np)		
	Overall	fv+np	rv+np	fv+pop	rv+pop	fv	rv	silent
base	.459	.049	.121	.777	.804	.815	.775	.095
GRPO	.495	.027	.067	.868	.910	.905	.882	.460
IAC	.495	.025	.059	.871	.915	.910	.897	.402
HRA	.489	.030	.075	.855	.891	.891	.869	.303

Table 5: Category lock (TKY-B, full test set, step 200). HRA anchored on the count rule or on the category rule. Rule-follow rates on the $W{=}1$ layer as in Table 3.

	HR@1			Follow ($W{=}1$)	
	$W{=}0$	$W{=}1$	Overall	Count rule	Category rule
base	.226	.550	.400	.668	.643
GRPO	.191	.761	.497	.775	.879
HRA, count anchor	.221	.697	.476	.818	.803
HRA, category anchor	.218	.698	.475	.802	.805

rollouts execute the locked rule, the update keeps reinforcing that rule on aligned prompts, while on misaligned prompts the correct-answer gradients scatter across rank positions and cancel. The diagnosis suggests a minimal intervention that changes nothing else: move the centre from the group mean to a fixed external standard. Let $r_H(q)$ be the expected reward on prompt q of a heuristic H that ranks candidates by a single feature, breaking ties uniformly at random—the count rule unless stated otherwise. Because the reward is an affine function of the target's position(equation 1) and its remaining components are deterministic given that position and the prompt, $r_H(q)$ is available in closed form, without generation. *Heuristic-Relative Advantage* (HRA) centres each rollout's reward on it:

$$\hat{A}_i^{\mathrm{HRA}} \;=\; \frac{r_i - r_H(q)}{\mathrm{std}(\{r_1,\dots,r_G\}) + \epsilon}\,, \qquad \epsilon = 10^{-4}, \tag{4}$$

with the standard deviation taken over the group's G rewards, as in GRPO. Groups whose reward standard deviation falls below 0.05—about the dispersion of a single one-rank disagreement at $G{=}5$—fall back to group-mean centring, and advantages are clipped to ± 10. Data, reward, model and optimiser are those of §3.1. Under HRA, executing the heuristic earns roughly zero advantage: only rollouts that *beat* it are reinforced, and only rollouts beaten by it are penalised. If the centre carries the asymmetry, moving it should move the cost in both directions—the misaligned subgroup should recover, and part of the aligned gain, bought by the lock's self-reinforcement, should be handed back. We therefore use HRA as an interventional test of the diagnosis rather than as a performance method. As a contrast we use *Inverse-Success Advantage Calibration* (IAC, Fan et al., 2026) that keeps the group-mean centre and upweights rollouts that beat it—on aligned prompts, precisely the heuristic's own executions (Appendix A.21).

4.2 Results

Count lock: the cost moves in both directions. On NYC Vanilla (Table 4), HRA raises HR@1 on both misaligned quadrants relative to GRPO (exact McNemar $p < 10^{-13}$), recovering about 15% of GRPO's loss, while count-rule following and the silent fraction recede. In return it hands back about a sixth of GRPO's aligned gain, so overall HR@1 ends 0.7 points below GRPO, still above base overall and on both aligned quadrants; the misaligned quadrants remain below base. IAC moves HR@1 and count-rule following the opposite way. The recovery has the signature the diagnosis implies: it concentrates at count-rank 2, where one better placement beats the heuristic, and is near zero at ranks ≥ 3. Tokyo Vanilla, where GRPO's misaligned collapse is deeper, reproduces both directions with about twice the effect: HRA recovers about a quarter of the misaligned loss, again concentrated at count-rank 2, and hands back about a quarter of the aligned gain (Appendix A.22).

Category lock: the spillover is undone. On TKY-B the lock is on category (§3.5), and its cost appears as spillover in the signal-absent region. Re-anchoring removes it: $W{=}0$ HR@1 returns to within a point of the base level under either anchor, at a cost on $W{=}1$ that still leaves it well above

base (Table 5). Contrary to our pre-specified expectation that the anchor must match the locked feature, the two anchors rescue $W{=}0$ about equally. What the anchor decides is where the freed probability goes. Split by the target's count-rank, the count anchor gains most at rank 1, where its own heuristic is correct, and the category anchor at rank 2 (Appendix A.23). Both anchors soften the category lock by the same amount (Cat-Rule Follow 0.803 vs. 0.805) and both restore some count following, the count anchor more so (Table 5). Re-anchoring thus acts as a *floor plus a steer*: falling below the heuristic is penalised, following it is not rewarded, and beating it is. Any fixed anchor breaks the group mean's self-reinforcement; the choice of anchor sets which rule's performance is protected.

4.3 Why re-anchoring works, and what it cannot do

Proposition 1 makes the effect of re-anchoring exact. Writing $\bar{w} = \frac{1}{G}\sum_i w_i$ for the group's mean unweighted direction and adding and subtracting $\bar{r}$ gives, for non-fallback groups and ignoring ϵ and clipping,

$$g_{\mathrm{HRA}} \;=\; \sum_{i=1}^{G} \frac{r_i - r_H(q)}{\sigma}\, w_i \;=\; g \;+\; \frac{G\,\big(\bar{r} - r_H(q)\big)}{\sigma}\,\bar{w}, \tag{5}$$

where g is the GRPO gradient of equation 3. Re-anchoring leaves GRPO's contrast term untouched and adds one group-level term, the mean direction scaled by the group's margin over the heuristic. This is exactly the component that group-mean centring cancels (Proposition 1): GRPO carries only *contrast*—which rollouts beat their peers—and no *level*—whether the group as a whole beats or trails a standard. In expectation over rollouts this term adds only a multiple of GRPO's own expected update, since policy gradients are invariant to action-independent baselines (up to the normalisation by σ); its effect is a finite-group one. Because $\bar{w}$ points at the outputs the group actually sampled, the level term suppresses a sampled consensus that trails the heuristic; where the heuristic is correct it is non-positive, penalising deviation without rewarding execution (Appendix A.24). This is the floor-plus-steer behaviour of §4.2, and it explains why any fixed anchor breaks the lock: what sustains the lock is the peer-relative centre itself.

What it cannot do. Across prompts the level term carries no rank structure: the unweighted per-prompt gradient of §3.4 is exactly $G\bar{w}$, and it sits at the uniform floor (Figure 2, right). The level term can suppress a consensus but cannot point toward a correct answer the group rarely samples—hence the concentration of the recovery at count-rank 2 and its near-absence at ranks ≥ 3. These bounds hold for any method that recombines the gradients of already-sampled rollouts, whether by reweighting (HRA, IAC) or by projection (PCGrad, Yu et al., 2020); we return to them in §5.

5 Limitations and discussion

Our evidence comes from a single controlled testbed: next-POI ranking over eight candidates, in two cities, with a second backbone on one construction. We chose it because it lets us fix which features are readable and compute the optimal policy. The mechanism needs only a position-graded reward, a group-relative centre and a feature that is informative only on average, but whether it carries over to tasks with less separable shortcuts, such as reasoning, remains untested; likewise, the readability criterion is tested prospectively in one construction where the prior and coherence disagree. HRA is a probe rather than a method: it moves the cost as the diagnosis predicts but recovers only part of the misaligned loss, lowers overall HR@1, and acts through finite groups, so its effect may weaken with larger groups; all training runs use a single seed. More fundamentally, the misaligned cost is the price of the constrained optimum (Theorem 1): methods that recombine the gradients of sampled rollouts can shift it between subgroups but not remove it (§4.3). Removing it requires giving the policy information that separates the misaligned cases, or an objective that values them explicitly.

Beyond recommendation, the same regime arises wherever a verifiable answer is chosen from a bounded set and some cue predicts it only on average, as in multiple-choice tasks, where GRPO is already known to reward lucky guesses (Wang et al., 2026). Because rule coherence can be measured from the base model's gradients on about 1,500 prompts (Appendix A.18), such cues can be screened before training rather than discovered after it.

AI USE STATEMENT

In this work, we used generative AI tools for (i) writing assistance, including editing and condensing text, and proposing wording for and captions, introduction and related work; (ii) literature search and bibliographic support, including locating candidaterelated work and checking titles, author lists, venues and arXiv identifiers for the references; and (iii) coding assistance for analysis, including writing and running scripts that compute evaluation statistics and gradient diagnostics from our saved checkpoints and rollouts. We have not used generative AI tools to generate or alter experimental data, to fabricate or estimate any reported number, or to write the core theoretical proofs, and other required-disclosure tasks are not applicable to this work. Additionally, we used generative AI as an auxiliary tool to draft schematic figures, e.g. the mechanism diagram in Figure 1, and to check internal consistency between the main text, tables and appendix.

We have reviewed all AI-assisted work. Every number reported in the paper was traced back to the corresponding result file produced by our training and evaluation pipeline, and statements for which no result existed were removed or marked as open. AI-suggested citations were checked against the original papers, and entries that could not be verified were excluded. AI-generated analysis code was inspected and its outputs were cross-checked against independent computations where possible (for example, the pairwise coherence values were recomputed directly and compared with values derived from a different estimator). The research questions, experimental design, and the interpretation of results were decided by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations (ICLR)*, 2020. arXiv:2002.10657.

Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems*, 2023. arXiv:2010.03240.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hongxun Ding, Keqin Bao, Jizhi Zhang, Yi Fang, Wenxin Xu, Fuli Feng, and Xiangnan He. Towards sample-efficient and stable reinforcement learning for LLM-based recommendation. *arXiv preprint arXiv:2602.00632*, 2026.

Mingyuan Fan, Weiguang Han, Daixin Wang, Cen Chen, Zhiqiang Zhang, and Jun Zhou. When sharpening becomes collapse: Sampling bias and semantic coupling in rl with verifiable rewards, 2026. URL https://arxiv.org/abs/2601.15609.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning (ICML)*, 2023. arXiv:2210.10760.

Scott Garrabrant. Goodhart taxonomy. AI Alignment Forum, 2017. https://www.alignmentforum.org/posts/EbFABnst8LsidYs5Y/goodhart-taxonomy.

Katherine L. Hermann, Hossein Mobahi, Thomas Fel, and Michael C. Mozer. On the foundations of shortcut learning. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.16228.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Mitsuki Sakamoto, Ryota Mitsuhashi, and Eiji Uchibe. MO-GRPO: Mitigating reward hacking of group relative policy optimization on multi-objective problems. *arXiv preprint arXiv:2509.22047*, 2025.

Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. Correlated proxies: A new definition and improved mitigation for reward hacking. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2403.03185.

Lauro Langosco, Jack Koch, Lee Sharkey, Jacob Pfau, Laurent Orseau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2022. arXiv:2105.14111.

Gang Li, Ming Lin, Tomer Galanti, Zhengzhong Tu, and Tianbao Yang. DisCO: Reinforcing large reasoning models with discriminative constrained optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. arXiv:2505.12366.

Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D. Salim. Large language models for next point-of-interest recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2024. arXiv:2404.17591.

Peibo Li, Shuang Ao, Hao Xue, Yang Song, Maarten de Rijke, Johan Barthélemy, Tomasz Bednarz, and Flora D. Salim. Refine-POI: Reinforcement fine-tuned large language models for next point-of-interest recommendation. *arXiv preprint arXiv:2506.21599*, 2025b.

Ruiming Liang et al. Don't mix rewards, mix policies: Policy decomposition and optimization for multi-reward RL. *arXiv preprint arXiv:2607.29246*, 2026.

Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2020. arXiv:2007.13019.

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2010.15775.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2201.03544.

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

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2209.13085.

Jiamian Wang, Samyadeep Basu, Koustava Goswami, Tong Yu, and Zhiqiang Tao. Spurious advantage hidden in GRPO. *arXiv preprint arXiv:2609.04063*, 2026.

Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-GRPO: Pairwise preference reward-based GRPO for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2025.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why RLVR may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

Yihang Yao, Zuxin Liu, Zhepeng Cen, Peide Huang, Tingnan Zhang, Wenhao Yu, and Ding Zhao. Gradient shaping for multi-constraint safe reinforcement learning. In *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, volume 242 of *PMLR*, pp. 25–39, 2024. arXiv:2312.15127.

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

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. arXiv:2001.06782.

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

Yaochen Zhu, Harald Steck, Dawen Liang, Yinhan He, Vito Ostuni, Jundong Li, and Nathan Kallus. Rank-GRPO: Training LLM-based conversational recommender systems with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.20150.

A Appendix

A.1 Extended related work

Reward misspecification and spurious features. Reward hacking was identified early as a core risk of optimising a specified objective (Amodei et al., 2016), and Goodhart-style taxonomies catalogue the ways a proxy can come apart from its target (Garrabrant, 2017). Later work found that more capable agents exploit misspecified rewards more, sometimes through abrupt phase transitions (Pan et al., 2022); formalised when a proxy is hackable (Skalse et al., 2022); measured how learned reward models are over-optimised (Gao et al., 2023); and redefined hacking as a proxy–reward correlation that breaks down under optimisation, mitigated by regularising towards a reference policy (Laidlaw et al., 2025). In our setting the correlation that matters is between an observed feature and correctness, and optimisation does not break it but applies it uniformly, because doing so is optimal; no better reward removes the cost. In supervised learning, Nagarajan et al. (2021) trace reliance on spurious features to geometric and statistical skews in the data; we likewise explain feature choice through the optimiser, but the geometry that selects the feature is created by the group-relative advantage rather than by the data.

Interference between rewards and objectives. A growing body of work studies how several reward signals interact inside one policy update. MO-GRPO normalises each reward component by its within-group variance so that no component dominates the advantage (Ichihara et al., 2025); Pref-GRPO observes that pointwise reward models assign tightly clustered scores within a group, which normalisation inflates into an illusory advantage, and replaces them with pairwise preferences (Wang et al., 2025); and policy-decomposition approaches train separate policies per reward instead of mixing rewards into one scalar (Liang et al., 2026). Earlier, conflicts between objectives were resolved at the gradient level, by projecting away conflicting components in multi-task learning (Yu et al., 2020) or by shaping constraint gradients in safe RL (Yao et al., 2024). These methods address several rewards or constraints; our setting has one exact reward, and the interference arises between subgroups of prompts rather than between reward components. As §4.3 shows, any method that recombines the gradients of already-sampled rollouts, whether by reweighting or projection, shares the same ceiling.

Does RLVR sharpen or teach? A parallel debate asks whether RLVR creates new behaviour or re-weights behaviour the base model already has. Yue et al. (2025) find that RLVR-trained models win at small sampling budgets but lose to their base models at large ones, and Wu et al. (2025) argue that RLVR shrinks the support of the base distribution more than it expands it. Our lock-in is a controlled instance of sharpening: the rule is present in the base model's preference and is made near-deterministic by training, with a cost that we can locate by subgroup and trace to the advantage.

A.2 Proof of Theorem

We wish to minimise $S(\sigma) = \sum_{k=1}^{n} q_k \, \sigma(k)$ over all permutations σ of $\{1, \ldots, n\}$, where $q_1 > q_2 > \cdots > q_n > 0$. The rearrangement inequality (Hardy, Littlewood & Polya, 1934) states that for two sequences $a_1 \geq \cdots \geq a_n$ and $b_1 \leq \cdots \leq b_n$, the sum $\sum a_i b_i$ is minimised when the larger a_i

are paired with the smaller b_i—i.e. when the two sequences are oppositely sorted. Setting $a_k = q_k$ (decreasing) and $b_k = k$ (increasing), we obtain:

$$S(\sigma) = \sum_{k=1}^{n} q_k \, \sigma(k) \geq \sum_{k=1}^{n} q_k \cdot k = S(\sigma^*) \,,$$

with equality when $\sigma(k) = k$ for all k. For uniqueness, suppose $\sigma \neq \sigma^*$. Then there exist indices $i < j$ such that $\sigma(i) > \sigma(j)$. Define σ' by swapping positions i and j:

$$S(\sigma) - S(\sigma') = (q_i - q_j)\big(\sigma(i) - \sigma(j)\big) > 0 \,,$$

because $i < j$ implies $q_i > q_j$ (strict) and $\sigma(i) > \sigma(j)$ (by choice). Hence σ is strictly suboptimal. Since this holds for every $\sigma \neq \sigma^*$, the minimiser is unique. The proof is a direct application of the rearrangement inequality: since (q_k) is strictly decreasing, pairing it with the ascending identity $\sigma(k) = k$ uniquely minimises $\sum q_k \, \sigma(k)$. This result does not depend on GRPO or any particular training algorithm—it is a property of the reward function under the observation constraint, and any algorithm that approximately maximises expected reward will be driven toward the same strategy.

A.3 Extension to multiple observable features

The main text (§3.1) formulates the observation constraint for a single feature (count), which suffices for Construction Vanilla (§3.1). Constructions A and B (§3.2) introduce category as a second observable feature; here we state the general formulation. Suppose the model can observe m features per candidate, with ranks $k_1, k_2, \ldots, k_m$ (each in $\{1, \ldots, n\}$). The strategy becomes a mapping $\sigma : (k_1, k_2, \ldots, k_m) \mapsto$ output position, and the full strategy over all n candidates must still produce a valid permutation. The reward remains equation 1. When $m > 1$ the rearrangement-inequality argument no longer yields a unique optimum in closed form, because the joint distribution $P(k_{\text{target},1}, \ldots, k_{\text{target},m})$ may not be totally ordered. In practice, one feature tends to dominate: its empirical $\hat{q}_1$ is close to 1, making the single-feature theorem an accurate approximation. Which feature dominates—and which the model locks onto—is determined empirically in each construction (§3.2, §3.5).

A.4 Reward function details

The full reward used in training decomposes as $R_{\text{full}} = R_{\text{format}} + R_{\text{position}} + R_{\text{bonus}}$. R_{format} is a binary term that equals 1 when the model's output is a valid ranking (parseable permutation) and 0 otherwise. Since this term is constant across all valid rankings, it does not affect the relative reward of different candidate orderings and is irrelevant to Theorem 1. R_{position} is equation 1, the term analysed in the main text. R_{bonus} is a bonus of up to 0.2 awarded when the target is placed first *and* the current time falls within a preferred time window for the target POI. This bonus reads only the target's own temporal attribute; it does not depend on the relative ordering of candidates. Its effect is to widen the reward gap between placing the target first vs. second for certain samples, but the argmax direction ("place the highest-q_k candidate first") is unchanged. Quantitatively, the gap between adjacent ranks fluctuates within $[1/(n{-}1),\ 1/(n{-}1) + 0.2]$, which does not alter Theorem 1.

A.5 Verification of the observation constraint

To confirm that count-rank is the only discriminating signal in Construction Vanilla prompts, we perform three checks. First, all n candidates in each Vanilla prompt belong to the same category by construction; we verify this programmatically in 100% of training and test prompts. Second, candidate names (POI names) are distinct strings but carry no systematic correlation with target identity: a model trained on Vanilla prompts with names replaced by anonymous tokens ("Candidate A", …, "Candidate H") achieves statistically indistinguishable performance from the full-name version ($p > 0.3$). Third, we audit the prompt template to confirm that beyond the count field, no candidate-level attribute varies across candidates within a prompt—fields such as address and opening hours are either absent or identical across candidates in the same prompt.

Table 6: Empirical count-rank distributions $\hat{q}_k$ (share of prompts whose correct candidate has count-rank k), per construction and signal region on the NYC test set (N=36,834; W=1: N=14,276, W=0: N=22,558), and pooled on the training split (N=153,428). Values are rounded to three decimals; for A on test W=1, $\hat{q}_8 = 0.0004$ (shown as .000). All rows are strictly decreasing, as required by Theorem 1.

Constr.	Split/W	$\hat{q}_1$	$\hat{q}_2$	$\hat{q}_3$	$\hat{q}_4$	$\hat{q}_5$	$\hat{q}_6$	$\hat{q}_7$	$\hat{q}_8$
Vanilla	test W=1	.587	.201	.100	.056	.031	.017	.007	.002
	test W=0	.487	.204	.126	.086	.052	.031	.011	.004
	train	.515	.208	.117	.074	.047	.026	.011	.003
A	test W=1	.672	.177	.073	.039	.022	.012	.005	.000
	test W=0	.496	.206	.125	.078	.054	.028	.011	.002
	train	.559	.195	.104	.065	.042	.024	.009	.001
B	test W=1	.408	.169	.121	.098	.072	.057	.046	.029
	test W=0	.462	.120	.101	.090	.078	.066	.052	.032
	train	.443	.138	.111	.094	.077	.062	.047	.027

Table 7: Signal ceilings by construction and signal region W (all data properties, recomputed per cell). *Count rule*: pick the candidate with the most historical visits; *Category rule*: pick the candidate whose category appears most often in the recent 5-check-in window. Count/Cat Rule Ceiling: expected HR@1 of a perfect rule follower.

	W	Count Rule Ceiling	Cat Rule Ceiling
Vanilla	1	0.577	0.127*
	0	0.472	—
A	1	0.664	0.808
	0	0.484	—
B	1	0.299	0.746
	0	0.229	—

* Degenerated: mean fraction of the 8 candidates tied at the category-frequency max ($1/8$=0.125: fully discriminative; ≈ 1: vacuous). Vanilla reaches 0.994 on W=1 (its negatives share the target's category); A and B are at 0.168 and 0.186.

A.6 Empirical verification of the monotone count-rank premise

Theorem 1 assumes that the count-rank distribution of the target is strictly decreasing, $q_1 > q_2 > \cdots > q_8$, where $q_k = \Pr[\text{count-rank of the correct candidate} = k]$ and the rank is defined as $1 + \#\{\text{candidates with strictly more visits than the target}\}$ (tied candidates do not outrank the target). Table 6 reports the empirical $\hat{q}_k$, computed as pure data properties of the candidate sets (no model inference), for every construction $\times$ signal-region cell of the NYC test set and pooled over the training split. All nine rows are strictly decreasing, so the premise holds on both the training distribution that drives the lock-in dynamics and the test distribution we evaluate on. The smallest adjacent gap (B, test W=0, $\hat{q}_2 - \hat{q}_3 \approx 0.02$) is an order of magnitude larger than its sampling standard error (≈ 0.002 at N=22,558), so the ordering is not an artefact of finite-sample noise.

Note the flattening pattern across constructions: B's $\hat{q}_1$ (0.41–0.46) is markedly lower than Vanilla's (0.49–0.59) and A's (0.50–0.67), which is the distributional signature of the visit-count-matched negatives and mirrors the depressed Count-Rule Ceiling of B reported in the main text.

A.7 Signal Ceilings

We report the signal ceilings by construction and signal region W (all data properties, recomputed per cell) in Table 7.

Table 8: Construction A: HR@1 by region and subgroup (ckpt-200, N=36,834)

Region / Subgroup	Base HR@1	GRPO HR@1	Reference
W=1 (all)	0.517	0.774	strat 0.664
W=0 (all)	0.388	0.424	strat 0.484
W=0, fv+nonpop	0.151	0.089	$1/8 = 0.125$
W=0, rv+nonpop	0.212	0.144	$1/8 = 0.125$

Table 9: Construction B, W=1: category lock-in metrics (ckpt-200).

Metric	Base	GRPO@200	Reference
HR@1	0.362	0.637	ceiling 0.746
Cat-Rule Follow rate	0.435	0.747	random 0.186
Category-match rate	0.467	0.791	random 0.212
Nonpop error: cat-rank-2 share	∼uniform	0.936 / 0.900	$p \approx 10^{-8}$

A.8 Full Construction A and B results

Construction A. Table 8 reports per-region metrics at ckpt-200. On W=0, the pattern is identical to the vanilla condition: the misaligned subgroup (fv+nonpop, rv+nonpop) is close to or less than the random baseline $1/n = 0.125$, confirming that count lock-in and its associated overshoot replicate when category is introduced as a second feature but count remains the dominant signal.

Construction B, W=1: category lock-in in detail. Table 9 reports the full set of lock-in metrics on W=1. Three indicators converge: (i) the Cat-Rule Follow rate rises from 0.435 to 0.747, consuming 55.2% of the available headroom ($\Delta = +0.312$; random baseline 0.186); (ii) HR@1 rises to 0.637, reaching 85% of the perfect-category-follower ceiling (0.746); (iii) the category-match rate rises from 0.467 to 0.791 (random baseline 0.212). Together, these show that GRPO has installed a category-following rule as sharp as Vanilla's count rule. The error-mode signature provides a further diagnostic. In the nonpopular subgroup of W=1, the base model's errors are nearly uniformly distributed over category ranks (goodness-of-fit $p = 0.89$); at ckpt-200, 93.6% of errors (fv+nonpop) and 90.0% (rv+nonpop) fall precisely on the category-rank-2 candidate ($p \approx 10^{-8}$). Errors are not merely non-random but rank-ordered: even when the model is wrong, it is wrong in the direction of category rank—the same "errors follow the rule" signature as Vanilla's count lock-in.

Construction B, W=0: spillover mechanism in detail. Table 10 reports the full diagnostic panel on the signal-absent region. Three candidate drivers are assessed:

1. **Residual count lock-in (excluded).** The Count-Rule Follow rate moves from 0.601 to 0.618, a change of $+0.017$—far below the $\sim +0.15$ shifts seen under count lock-in in Vanilla and A. Count is not the driver.

2. **Position collapse (excluded).** The top-1 label concentration—the fraction of outputs placing the single most common label first—does not rise ($0.181 \rightarrow 0.166$). If anything, output diversity slightly *increases*, ruling out degenerate collapse onto one position.

3. **Random degradation (excluded).** The errors are directional rather than diffuse.

4. **Unconditional category-rule application (confirmed).** The fraction of outputs selecting a candidate whose category matches the user's recent history doubles from 0.119 to 0.249 on W=0. Since W=0 targets are defined as those whose category is absent from the history, every such selection misses the correct answer. The $\sim 2.1\times$ amplification mirrors the $\sim 1.7\times$ amplification of Cat-Rule Follow on W=1 and the count-argmax amplification in Construction Vanilla.

Table 10: Construction B, $W{=}0$: spillover diagnostic panel (ckpt-200).

Metric	Base	GRPO@200	Reference
HR@1 (all)	0.212	0.187	strat 0.229; $1/8 = 0.125$
fv+nonpop HR@1	0.134	0.098	below $1/8$
rv+nonpop HR@1	0.135	0.103	below $1/8$
Count-Rule Follow rate	0.601	0.618	barely changes
Top-1 label concentration	0.181	0.166	does not rise
Cat-window selection rate	0.119	0.249	$2.1\times$ amplification

A.9 The category rule applied where count alone suffices

In Construction B (NYC), the re-visit+popular quadrant of the $W{=}1$ layer ($N{=}4{,}339$) is a cell where count alone is sufficient: the correct candidate attains the maximum visit count, and an ideal count-rule follower would achieve HR@1 $= 0.874$ (Count-Rule Ceiling, uniform tie-breaking). The trained model underperforms this one-feature baseline in exactly the way the category-lock account predicts (Table 11): it executes the category rule essentially as often as the count rule (Cat-Rule Follow 0.849 vs. Count-Rule Follow 0.855 at ckpt-200) and pays 5.5 points of HR@1 below what count alone achieves. The deficit persists at ckpt-400.

Table 11: Construction B (NYC), $W{=}1$, re-visit+popular ($N{=}4{,}339$). Count-Rule Ceiling $= 0.874$ throughout (data property).

Model	HR@1	Count-Rule Follow	Cat-Rule Follow	cat-match
base	0.525	0.606	0.552	0.591
ckpt200	0.819	0.855	0.849	0.901
ckpt400	0.837	0.870	0.868	0.921

A.10 Training dynamics

Tables 12 and 13 report the full-test metrics at checkpoints 0/200/400 for the two NYC constructions. In both, the bulk of the lock-in and of the misaligned cost is in place by step 200: Cat-Rule Follow completes 88% of its total 400-step rise in Construction B ($0.435 \to 0.747$ of $0.435 \to 0.788$), Count-Rule Follow 81% in Vanilla ($0.809 \to 0.910$ of $0.809 \to 0.933$), and the $W{=}0$ spillover 86% ($0.212 \to 0.187$ of $0.212 \to 0.183$). Steps 200–400 add a small, monotone deepening, consistent with the self-limiting account of §3.3: once the rule is near-deterministic, most prompt groups are silent (Appendix A.11) and little gradient remains. TKY-B shows the same saturation (Table 16 in Appendix A.18).

Table 12: NYC Vanilla (count-only construction), full test ($N{=}36{,}834$), by checkpoint. HR@1 per quadrant; Count-Rule Follow overall.

	Overall	argmax-	HR@1 by quadrant			
Checkpoint	HR@1	count	fv+nonpop	rv+nonpop	fv+pop	rv+pop
base	0.459	0.809	0.049	0.121	0.777	0.804
ckpt200	0.495	0.910	0.027	0.067	0.868	0.910
ckpt400	0.502	0.933	0.021	0.051	0.887	0.933

A.11 Silent groups and the self-limiting dynamics

Definition. The task reward is rank-based, $r = (n - \mathrm{rank})/(n-1)$ plus temporal and format bonuses (with rank the position of the correct candidate in the parsed ranking; $r = -0.5$ if unparseable). Group-relative optimisation assigns every rollout in a prompt group the advantage

Table 13: NYC Construction B, full test (W=1: N=14,239; W=0: N=22,595), by checkpoint.

	W=1			W=0
Checkpoint	HR@1	Cat-Rule Follow	Cat-Rule Follow|unique	HR@1
base	0.362	0.435	0.395	0.212
ckpt200	0.637	0.747	0.718	0.187
ckpt400	0.674	0.788	0.764	0.183

$\hat{A}_i = (r_i - \bar{r})/\operatorname{std}(\{r_j\})$. When all G rollouts of a group obtain the same reward, the advantage vanishes identically and the group contributes no gradient. We call such groups *silent*. Under a near-deterministic policy, silence is generic: the sampled rankings coincide, so the rewards coincide. **Measurement.** 600 non-popular prompts drawn from the training pool (seed 123); G=5 rollouts at temperature 0.9 (the training sampling configuration); reward identical to training. Evaluated at base and at Vanilla checkpoints 200/400/500, plus DAPO@200 (Appendix A.20). **Results.** The silent fraction rises from 0.095 at base to 0.460 at ckpt-200—the same 200-step window in which lock-in and spillover saturate (Appendix A.10)—and then stabilises (0.512 at ckpt-400, 0.507 at ckpt-500), mirroring the behavioural plateau. The rise is shared across the two non-popular quadrants (first-visit $0.103 \to 0.470 \to 0.500$; re-visit $0.087 \to 0.450 \to 0.523$ at base /200 /400). Of the silent groups at ckpt-200, 85.5% have mean reward above 0.5: groups where all rollouts place the correct candidate at the same moderately high rank—the rank-2 error-concentration signature of the locked count rule—rather than groups of uniformly failed rollouts. **Implication.** Silent groups close the loop of §3.3. Coherence installs the rule quickly; the installed rule homogenises behaviour within each prompt group; homogeneous groups fall silent; and with both the reinforcing and the reversing signals removed, the policy freezes at whichever rule coherence selected. The 200-step plateau is therefore not an artefact of the learning-rate schedule, but a structural property of group-relative optimisation: *agreement is silent, whether the agreed answer is right or wrong*.

A.12 Cross-token gradient orthogonality

Setup. At the answer position, let $z = Wh$ be the logits, $p = \operatorname{softmax}(z)$, and $\ell = -z_y + \log \sum_k e^{z_k}$ for selected token y. Then $\partial \ell / \partial z = p - e_y$, and the parameter gradient is $g(y) = J^\top (p - e_y)$ with $J = \partial z / \partial \theta$. Writing $v_k = J^\top e_k$ and collecting the residual mass into $\bar{v} = \sum_{k \neq y} \frac{p_k}{1-p_y} v_k$, the gradient factors as

$$g(y) \;=\; \sum_k p_k v_k - v_y \;=\; -(1 - p_y)\left(v_y - \bar{v}\right). \tag{6}$$

Assumptions. (A1) Distinct answer-token directions are near-orthogonal, $v_a^\top v_b \approx 0$ for $a \neq b$: the dominant contribution to v_k is the unembedding row of token k, and distinct unembedding rows of a pretrained vocabulary have small pairwise cosine. (A2) The policy is near-deterministic on the selected token (p_y bounded away from the residual mass), so $\|\bar{v}\|$ is small relative to $\|v_y\|$; this holds increasingly as a rule is locked. **Consequence.** For two rollouts selecting $a \neq b$,

$$g(a)^\top g(b) \;\approx\; (1 - p_a)(1 - p_b)\left(v_a^\top v_b + \bar{v}_a^\top \bar{v}_b\right) \;\approx\; 0\,, \qquad \|g(a)\| \approx (1 - p_a)\,\|v_a\|\,,$$

so cross-token pairs are near-orthogonal while same-token pairs share the v_y direction and align. Two predictions follow, both borne out: the orthogonality is representation-level and therefore invariant to the label symbol set (Appendix A.17), and it strengthens as the policy sharpens over training. Empirically, cross-token mean cosine is -0.003 under letter labels and $+0.0004$ (CI $[-0.0005, 0.0014]$) under digit labels.

A.13 Magnitude vs. coherence: the gradient audit

Protocol. At each checkpoint we re-sample rollouts *on-policy* with the training configuration (G=5, T=0.9, top-p=1) from a stratified draw of the training pool (300 groups per quadrant, giving 600 aligned and 600 misaligned groups), recompute the group-normalised advantages exactly as in training, and take per-group gradients over the LoRA parameters only—the parameters that training

actually updates. A group whose rollouts all receive identical rewards has zero advantage and contributes no gradient; we call it *silent*. Each subgroup aggregates 600 groups (300 per quadrant); the coherence statistics below are computed over the non-silent ones (91–121 aligned, 292–314 misaligned, varying by checkpoint and seed). We summarise coherence by the *aggregation efficiency* $\eta = \|\sum_i g_i\| / \sum_i \|g_i\|$ over a subgroup's non-silent groups: $\eta{=}1$ if all group gradients point the same way, and random directions give the floor $1/\sqrt{N}$. η is a monotone transform of the mean pairwise cosine $\bar{c}$ of Proposition 1: at equal norms $\eta^2 = \big(1 + (N{-}1)\,\bar{c}\big)/N$, hence $\bar{c} = (N\eta^2 - 1)/(N-1)$; the group norms here are nearly equal (coefficient of variation 0.20–0.28; Table 14), so the conversion is accurate. The cross-subgroup cosine is taken between the two aggregate gradients. Rollout sampling is seeded; ckpt-400 is measured with three seeds (prompts fixed, only the rollouts resampled). **Magnitude dominance is refuted.** Per-group gradient norms are essentially equal across subgroups at every checkpoint (Table 14): mean $\|g\|$ over non-silent groups is 5.8–6.4 on the aligned side and 5.5–5.7 on the misaligned side, and the aggregate-norm ratio stays within 1.09–1.24. If anything the misaligned side produces *more* gradient mass: fewer of its groups are silent (48–51% vs. 80–85%), so 2.6–3.2× more of its groups contribute. The aligned signal does not win by pushing harder. **Coherence is where the asymmetry lives.** The aligned subgroup's group gradients share a direction ($\eta \approx 0.18$–0.21, about twice its random-direction floor $1/\sqrt{N} \approx 0.10$ at $N \approx 91$–121), while the misaligned subgroup's sit at theirs ($\eta \approx 0.06$ over $N \approx 292$–314): some $1 - \eta \approx 94\%$ of the misaligned signal cancels in aggregation. In terms of the mean pairwise cosine this is $\bar{c} \approx +0.03$ versus $\bar{c} \approx 0$: the conversion gives $+0.026$–$+0.035$ (aligned) and $+0.0003$–$+0.0007$ (misaligned), and a direct pairwise measurement on the same group gradients—accumulating the running sum of unit group gradients and reading $\bar{c}$ off $\|\sum_i \hat{g}_i\|^2$—agrees ($+0.026$–$+0.031$ and $+0.0003$–$+0.0005$ respectively; `verify_pairwise_cbar.json`). Meanwhile the two subgroups drift apart: the cross-subgroup cosine is $+0.08$ at ckpt-200 and turns negative by ckpt-400 (-0.03 to -0.20), staying negative at ckpt-500. **Robustness to the measurement protocol.** An earlier protocol ($G{=}8$, groups drawn

Table 14: Gradient audit on the training distribution (NYC Vanilla, $G{=}5$, on-policy resampling, 600 groups per subgroup). Non-zero = non-silent groups (of 600); η = aggregation efficiency $\|\sum_i g_i\| / \sum_i \|g_i\|$; cos = cosine between the two subgroups' aggregate gradients. Ckpt-400 is measured with three rollout seeds.

	non-zero groups		mean $\|g\|$ per group		η		
ckpt	aligned	misaligned	aligned	misaligned	aligned	misaligned	cos
200	121	314	5.76	5.58	.183	.061	$+.08$
400 (s0)	92	294	5.89	5.62	.208	.062	$-.05$
400 (s1)	95	297	5.97	5.71	.207	.064	$-.20$
400 (s2)	93	293	6.16	5.51	.214	.061	$-.03$
500	91	292	6.39	5.60	.203	.063	$-.04$

from the *test* distribution, 200 groups per quadrant) gives the same pattern with the sign of the magnitude comparison reversed: the misaligned aggregate norm is 23–30% *larger* (ratio 0.77–0.81), aggregation efficiency is 0.27 vs. 0.09, and the cross-subgroup cosine falls from 0.19 to 0.00 between ckpt-200 and ckpt-400. Under both protocols, magnitude favours the misaligned side or is a tie, while coherence favours the aligned side by an order of 3× or more, and the subgroups decouple over training.

A.14 Controls and components for the rank geometry

This section collects the measurements that localise the rank geometry of §3.4: two controls excluding representation-level accounts, one control identifying the advantage weighting as the carrier, a probe confirming that the gradient map itself is well-behaved, and the component measurements behind the plug-in check of equation 3. **Measurement setup.** Pairwise cosines in Table 15 are computed on $W{=}0$ non-popular train-pool prompts, using per-prompt LoRA gradients restricted to a fixed 12M-coordinate subset, validated against full vectors at Pearson 0.999. Throughout, rollouts are grouped by the output label of the *correct* candidate, the quantity the reward acts on. The label-independent part of completion gradients (shared context and format tokens) appears as the uniform

Table 15: Pairwise gradient cosine similarity by label match and target count-rank distance (Vanilla, ckpt-400). "Same label" pairs share the correct candidate's output label; the unweighted column recomputes everything on identical rollouts with the advantage set to 1. Cross-seed noise ceilings: 0.257 (weighted), 0.058 (unweighted). Measurement details in Appendix A.14. Same-label decay: permutation $p < 0.001$; Cohen's $d = 0.285$ for distance-0 vs. distance-$\geq$3 pairs ($n = 14{,}959$).

Pair type	Rank distance	Weighted	Unweighted
Different labels	all	-0.003	$+0.042$
Same label	0	$+0.035$	$+0.042$
Same label	1	$+0.032$	$+0.042$
Same label	3	$+0.022$	$+0.042$
Same label	6	$+0.013$	$+0.043$

similarity floor in the unweighted control. All reported decay curves are normalised to their value at rank distance 0. **Control: prompt-level representations carry no rank signal.** For all 36,834 Construction-Vanilla test prompts we extract the final hidden-layer state at the answer position and compute mean pairwise squared distances between prompts whose targets differ by rank distance δ, in closed form from per-rank group moments ($\mathbb{E}\|h-h'\|^2 = \nu_k + \nu_{k'} - 2\,\mu_k^\top \mu_{k'}$, so no pair sampling is involved). The profile is flat at every checkpoint: the total variation across $\delta = 0, \ldots, 7$ is under 0.6% of the mean distance and non-monotone (base $3826 \to 3826$; ckpt-400 $4157 \to 4174$, arbitrary units). This is the expected null, and an informative one: the correct candidate's identity is not part of the input, so the prompt representation cannot encode its rank; any nonzero signal would have indicated leakage. The mechanism of §3.4 must therefore live elsewhere. **Control: candidate-level representations register count coincidence, not rank distance.** For $4{,}000$ test prompts we extract the hidden state at each candidate line's final token and compute within-prompt distances, which cancel prompt-specific variance. The profile shows a single step—tied-count pairs (rank distance 0) are $\approx 12\%$ closer than all others (727 vs. 825 at ckpt-400)—and is flat thereafter (rank distance 1 vs. 6: $+2.5\%$), identically at base, ckpt-200, and ckpt-400. Representations register *whether* two counts coincide, not *how far apart* two ranks are, and do not encode count magnitude smoothly either. **Control: removing the advantage weighting removes the structure (decisive).** We recompute the full pairwise analysis of Table 15 on identical prompts, identical completions, and the identical coordinate subset (subset validation Pearson 0.998), with the advantage set to 1. The same-label decay vanishes ($0.0415 \to 0.0432$ across rank distances; contrast permutation $p = 0.78$; Spearman $+0.001$, $p = 0.93$; Cohen's $d = -0.008$), the label gating disappears (cross-label $+0.042$, indistinguishable from same-label), and the cross-seed noise ceiling drops from 0.257 to 0.058. Unweighted gradients share only the global "push-up-your-own-output" component; the two-level structure of §3.4 is entirely advantage-carried. **Probe: the gradient map is locally smooth.** Analytically, the loss is twice continuously differentiable in the answer-position hidden state (the softmax cross-entropy composed with smooth layers), with derivatives bounded on the region traversed by the data; the gradient map is therefore locally Lipschitz, and only the existence—not the value—of the constant enters §3.4. We confirm this directly with a perturbation probe that holds the sample fixed: for 150 test prompts (ckpt-400, fp32), we perturb the answer-position hidden state, $h \to h + \eta\, u$ with random unit directions u (4 per prompt) and η spanning $[10^{-4}, 10^{-1}] \cdot \|h\|$—three orders of magnitude—and recompute the single-token cross-entropy gradient on the model's greedy first answer token (held fixed). The directional-derivative norm $\|\Delta g\|/\|\Delta h\|$ has median 2.9×10^{-3} and varies by less than 15% across the whole η grid; the 95th percentile (≈ 0.26) is likewise η-independent. The local slope is therefore finite and prompt-dependent rather than divergent, exactly as local smoothness requires. (The right tail reflects a small set of prompts with larger local sensitivity, not a small subset of the η grid.) Residual rollout-sampling variation at fixed h is calibrated separately by the cross-seed noise ceiling (0.257) and is rank-independent.

A.15 Advantage–position identity: full conditions and proof

The update is on-policy and group-relative: importance ratio 1 and clipping inactive, as in all our runs. The KL term, which is not advantage-weighted, is excluded and handled separately. Any per-sample weighting of w_i, such as length normalisation, is absorbed into w_i. The residual b_i collects position-independent terms—the bounded temporal bonus, and rollouts whose reward departs from

the affine form (parse failures; correct candidate never emitted)—and is bounded because the reward itself is bounded. σ is the within-group reward standard deviation, a constant within the group, and $\widehat{\mathrm{Cov}}$ uses $1/G$ normalisation.

Proof of Proposition (Advantage–position identity). The group-relative advantage is $\hat{A}_i = (r_i - \bar{r})/\sigma = -\frac{c}{\sigma}(\mathrm{pos}_i - \bar{p}) + \frac{1}{\sigma}(b_i - \bar{b})$. Substituting into $g = \sum_i \hat{A}_i w_i$ gives the decomposition; $\sum_i (\bar{p} - \mathrm{pos}_i)\, w_i = -G\, \widehat{\mathrm{Cov}}(\mathrm{pos}, w)$ because $\sum_i (\mathrm{pos}_i - \bar{p})\, \bar{w} = 0$, and the final claim follows from $\sum_i (\bar{p} - \mathrm{pos}_i) = \sum_i (b_i - \bar{b}) = 0$. □

A.16 Components of the plug-in check of equation 3

(a) *Position tracks rank.* Across the 6,240 archived ckpt-400 rollouts (98.5% parseable), the completion position of the correct candidate has median exactly k at every rank k (means $2.3, 3.4, 4.7, 5.7, 6.5, 7.3, 7.4$ for $k = 2, \ldots, 8$; $\Pr[|\mathrm{pos} - k| \le 1]$ between 0.72 and 0.97; Spearman 0.86). Under the training pipeline's raw character parse (100% parseable) the same measurement gives Spearman 0.85. The coupling is not a product of lock-in: measured on fresh rollouts from the *base* model on the same 859-prompt sample (4,288 parseable rollouts, training parser), it already stands at 0.67 with median position $= k$ at every rank, and it rises monotonically through training ($0.67 \to 0.79 \to 0.85$ at ckpt-0/200/400). On the aligned side, the base policy's popularity prior places the correct candidate first on 66% of W=0 popular-prompt rollouts (median position 1; 1,996 rollouts on 400 prompts). (b) *Mean directions by (label, position).* Per-rollout unweighted LoRA gradients on the same fixed 12M-coordinate subset, accumulated online into the 8×8 (label, position) cells (26–203 rollouts per cell, no empty cell among labels C–H). (c) *Prediction.* The identity is a per-group statement, so the plug-in is evaluated at two aggregation levels. The *group-level* computation forms each prompt's contrast from its own G=5 rollouts, $\widehat{g}_i \propto \sum_{\mathrm{pos}} (\bar{p}_i - \mathrm{pos})\, n_i(\mathrm{pos})\, \mathbb{E}[w \mid L_i, \mathrm{pos}]$, and sums within target rank; the *kernel-averaged* variant first pools the rollouts of all prompts of a rank into one histogram n_k and forms a single contrast per rank. Only the first is licensed by equation 3, and the difference is material: the group-level plug-in reproduces the measured same-label shape (normalised $1.00, 0.79, 0.57, 0.50, 0.44, 0.44$ at $\delta = 0, \ldots, 5$ vs. measured $1.00, 0.93, 0.73, 0.65, 0.42, 0.42$; Spearman 0.943, RMSE 0.108, max. deviation 0.165), including the plateau at $\delta = 4$–5, whereas kernel-averaging predicts a decay far steeper than measured ($1.00, 0.63, 0.29, 0.24, 0.23, 0.17$; RMSE 0.303)—pooling across prompts before forming advantages systematically overstates the rotation. Both levels predict the cross-label collapse ($+0.021$, an order of magnitude below the same-label values and consistent with the measured -0.003) and, with the $(\bar{p} - \mathrm{pos})$ contrast removed, a flat floor (0.75 between mean directions). The residual shape gap spans $\delta = 1$–3, which the plug-in under-predicts (0.79 vs. 0.93 at $\delta = 1$; the deviation is largest, 0.165, at $\delta = 2$); we report it rather than smooth it over. The predicted cosines are between *mean* gradients, while the measured ones are between *per-prompt* gradients diluted by idiosyncratic components; the resulting scale gap is accounted for by the dilution calibrated through the cross-seed ceilings (0.257 weighted, 0.058 unweighted), and does not affect the shapes. A circularity caveat applies to both levels: the cell means $\mathbb{E}[w \mid L, \mathrm{pos}]$ are estimated on the same archived rollouts whose advantage-weighted aggregates are being predicted. A split-half control—estimating the cell means on one rollout seed and predicting the curve on the held-out seed—isolates any mechanical component of the agreement. Cell means from seed 0 predict the seed-1 curve at RMSE 0.273 (Spearman 0.886) and seed 1 predicts seed 0 at RMSE 0.378 (Spearman 0.943), against the in-sample RMSE of 0.108. The degradation is accounted for by noise in the target rather than by circular fitting: the two per-seed *measured* curves differ from each other by RMSE 0.303 (per-seed pair counts fall to 30–132 per bin beyond distance 3), so the out-of-sample prediction error sits at the measurement noise floor while the shape agreement survives. **Relation to §3.3.** The group-level coherence of §3.3 and the pairwise values of §3.4 are measured under different conditions (group-aggregated gradients on the training distribution vs. per-rollout gradients on the probe sample), but once the former is expressed as a mean pairwise cosine (Appendix A.13) the two land on the same scale: $\bar{c} \approx +0.03$ for the aligned subgroup here, against ≈ 0.02–0.04 for same-rank rollout pairs in §3.4. The two measurements agree on both the qualitative structure and, approximately, the magnitude. **The temporal-bonus correction is negligible.** Proposition 1 treats the temporal bonus ($0.2 \cdot \mathrm{pref\text{-}match}$, gated to deliberate/social modes and to rollouts placing the correct candidate in the top 3) as a bounded correction. The plug-in check bears this out

quantitatively. In the 859-prompt analysis sample the gate is open for 40.3% of prompts, with per-rank mean bonus 0.04–0.06 (maximum 0.2; the base reward's slope is $1/7 \approx 0.143$ per position). Folding the true per-prompt bonus into the kernel does not improve the fit at any relative weighting: scaling the bonus term by $0, 0.5, 1, 2$ gives RMSE $0.108, 0.115, 0.125, 0.147$ against the measured curve—monotonically worse. The bonus's only visible signature is the small uptick at $\delta = 5$, which the bonus-free kernel misses (Spearman 0.943) and the true kernel reproduces (Spearman 1.000). The affine approximation of the reward is therefore adequate for the shape analysis, and the observed curve is *not* explained by the bonus—in particular, the measured flatness relative to the kernel-averaged prediction is an aggregation-level effect, not a bonus effect.

A.17 Randomised-label control

The cross-token orthogonality of Appendix A.12 is claimed to be a structural property of the softmax output layer, not of the particular labels used; likewise, the same-token rank-distance decay of §3.4 could in principle be contaminated by the alphabetical ordering of A–H. To test both, we replace the candidate labels with digits and repeat the pairwise gradient analysis. **Setup.** The eight candidates are relabelled from A–H to 1–8 by their position in the shuffled list, so the digit assignment carries no count-rank information (it does carry list-position information, which matters for finding (iii) below). Weights are unchanged (Vanilla ckpt-400, no retraining), as is the analysis protocol of §3.4: 859 $W{=}0$ non-popular prompts from the training pool $\times$ 2 rollout seeds (919 gradient estimates retained), rank-stratified oversampling of rare ranks, per-sample LoRA gradients on a random 12M-coordinate subset (validated against full vectors, Pearson 0.9993). **Results.** Three findings confirm that the gradient structure is label-invariant:

1. **Cross-token orthogonality is preserved.** Mean cosine across different-digit pairs is $+0.0004$ (cluster-bootstrap CI $[-0.0005, +0.0014]$), matching the letter-label value -0.003: digit labels introduce no cross-token coherence.
2. **Same-token rank-distance decay is preserved.** Within same-digit pairs, cosine decays monotonically from $+0.0210$ at rank distance 0 to $+0.0096$ at distance 6 (permutation $p < 0.001$, Cohen's $d = 0.176$; letters: $+0.0345 \to +0.0133$, $d = 0.285$, digit effect $\approx 65\%$ of the letter magnitude). Both seeds replicate, and all eight digits show negative Spearman correlation (six of eight significant)—the decay is not carried by any single label. The absolute levels differ somewhat between symbol systems, but the gap narrows once read against the respective noise ceilings (same prompt, cross-seed: 0.230/0.202 digits vs. 0.257/0.267 letters), and we do not interpret it further.
3. **List-position adjacency, not label semantics, drives the residual structure.** Among different-label pairs, cosine is not flat: it decays with label-index distance $|i-j|$. The effect first appeared under digit labels, where it could have meant that numeric semantics leak into gradient space; however, repeating the analysis under letter labels revealed a *stronger* effect (letters: $+0.0078$ at $|i{-}j|{=}1$ to -0.0196 at 7, Spearman -0.160; digits: $+0.0068$ to -0.0105, Spearman -0.106; both seeds replicate). Since letters carry no arithmetic structure, the gradient reflects adjacency of output *positions* in the list—a real structural dimension of the paradigm, but one that is orthogonal to the count-rank mechanism studied in the main text and affects both label systems alike.

Conclusion. Under both symbol systems we find the same two-level structure: same-token coherence organised by count-rank distance, cross-token incoherence with a position-adjacency gradient. The structure documented in §3.4 is therefore a property of the discriminative multi-choice paradigm as realised in this task, not of alphabetical label assignment.

A.18 Cross-city evidence (TKY)

Construction. TSMC2014-TKY raw feed (573,703 check-ins, 2,293 users, 61,858 venues, 247 categories), processed by the identical pipeline as NYC with no additional filtering; test set $N{=}85{,}356$; $W{=}1$ share 53.7% (NYC: 38.8%). Base model identical to the NYC experiments (Qwen2.5-14B-Instruct); the city is the only change. Training protocol identical to NYC. **A near-tie, by accident and by design.** Construction B suppresses the count signal by sampling negatives uniformly over venues within the target's count bucket. In TKY the top bucket $[12, 12{+}372]$ spans three orders of

magnitude and holds 75% of samples, so a residual count signal survives: on the $W{=}1$ layer, Count-Rule Ceiling $= 0.622$ against a Cat-Rule Ceiling of 0.680. Rather than a category-only testbed, TKY-B is therefore a *two-signal near-tie*—which is what makes it a discriminating test of the coherence criterion: objective strength cannot break the tie, and the base prior (Count-Rule Follow 0.668 vs. Cat-Rule Follow 0.643) leans the other way. **Behavioural results.** Table 16 reports the full stratification. The category lock deepens the NYC-B signature: in the $W{=}1$ non-popular quadrants, HR@1 *rises* (first-visit $0.183 \rightarrow 0.388$) while count-following is actively *suppressed* (Count-Rule Follow $0.576 \rightarrow 0.472$) at ckpt-400. $W{=}1$ HR@1 at ckpt-200 (0.761) exceeds both single-rule ceilings (0.622 and 0.680), as expected when the two signals agree on a subset of prompts. The $W{=}0$ spillover replicates ($0.226 \rightarrow 0.191$). Error analysis: the distribution of cat-ranks of non-Cat-Rule Follow errors concentrates at cat-rank 2 (base GOF $p = 0.17$, i.e. random; ckpt-400 $p = 2.6\text{e}{-9}$)—the NYC-B error signature on a new city. Dynamics: saturation by step 200 (Cat-Rule Follow $0.879 \rightarrow 0.886$ from ckpt-200 to ckpt-400; $W{=}1$ HR@1 $0.761 \rightarrow 0.770$), as in NYC (Appendix A.10). On the 52% of $W{=}1$ prompts where the category rule selects a unique candidate, Cat-Rule Follow still rises from 0.539 to 0.828. **Direct gradient measurement (rule coherence).**

Table 16: TKY Construction B, full test ($N{=}85{,}356$), checkpoints 0/200/400. am-count / am-cat: Count-Rule Follow / Cat-Rule Follow rates (ties count; cat metrics reported on $W{=}1$ only).

		HR@1			am-count			am-cat		
Layer	Quadrant	base	200	400	base	200	400	base	200	400
$W{=}1$	fv+nonpop	.183	.382	.388	.576	.464	.472	.418	.718	.733
$W{=}1$	fv+pop	.651	.894	.913	.766	.949	.961	.690	.932	.943
$W{=}1$	rv+nonpop	.259	.435	.439	.504	.425	.433	.478	.746	.753
$W{=}1$	rv+pop	.730	.951	.961	.740	.956	.966	.750	.958	.963
$W{=}1$	ALL	.550	.761	.770	.668	.775	.784	.643	.879	.886
$W{=}0$	fv+nonpop	.078	.044	.040	.640	.672	.687	—	—	—
$W{=}0$	fv+pop	.333	.296	.303	.806	.778	.785	—	—	—
$W{=}0$	rv+nonpop	.100	.063	.058	.582	.630	.646	—	—	—
$W{=}0$	rv+pop	.534	.505	.522	.657	.624	.642	—	—	—
$W{=}0$	ALL	.226	.191	.194	.670	.680	.693	—	—	—

1,500 prompts from the TKY-B training pool on which the two rules disagree (unique argmax under each, category-argmax $\neq$ count-argmax); $G{=}5$ rollouts at temperature 0.9 under the training protocol; per-prompt, per-rule LoRA gradient estimates on a random 4M-coordinate subset (validated against full vectors, $r \approx 0.99$; step-0 replication of stored vectors, cosine > 0.9997). The primary estimator is advantage-weighted mean pairwise cosine in closed form; significance by user-block permutation (10,000 draws; permutation nulls are not zero-centred because user composition is entangled with rule labels, so all p-values are against the empirical null). Base model: cat$\times$cat $+0.0252$ vs. count$\times$count $+0.0048$ (cross $+0.0036$), difference $+0.0204$, Cohen's $d = 0.343$, $p < 10^{-4}$; unweighted difference $+0.0050$, $d = 0.091$, $p = 0.016$. At ckpt-200 the weighted gap persists ($d = 0.171$, $p < 10^{-4}$); the unweighted gap is absorbed by the composition null ($p = 0.995$), as expected once the policy has locked and within-user rule agreement has risen. This mirrors the NYC control of Appendix A.14, where removing the advantage weighting likewise erases the pairwise structure: in both cities, the coherence that selects the lock target lives in the advantage coefficients, not in the gradient directions themselves. **Other TKY constructions.** TKY-Vanilla (category degenerate) replicates the count lock ($W{=}1$ Count-Rule Follow $0.837 \rightarrow 0.972$ at ckpt-400; HR@1 converges to the residual count ceiling; non-popular collapse $0.060 \rightarrow 0.016$). TKY-A, where the category ceiling (0.855) exceeds Count-Rule Ceiling (0.850), locks category (Cat-Rule Follow $0.611 \rightarrow 0.899$; $W{=}1$ HR@1 $0.585 \rightarrow 0.887$, above both single-rule values), saturating between ckpt-200 and ckpt-400 ($+0.0006$). In all three TKY constructions the lock target follows the more coherent feature, and the 200-step saturation replicates.

A.19 Backbone replication (Llama-3.1-8B)

Protocol. Llama-3.1-8B-Instruct; identical NYC Construction B data and training configuration (LoRA $r{=}32$, $\alpha{=}64$; lr $5\text{e}{-6}$; $\beta = 0.04$; temperature 0.9; $G{=}5$); evaluated at ckpt-200 on the full

test set. **Results.** Table 17. The category lock is at least as clean as on Qwen ($W{=}1$ Cat-Rule Follow $0.526 \rightarrow 0.829$, $\Delta{+}0.303$, 64% of headroom; Qwen: $0.435 \rightarrow 0.747$, $\Delta{+}0.312$, 55%), count-following collapses on the $W{=}1$ layer ($0.677 \rightarrow 0.517$), and the $W{=}0$ spillover replicates ($0.180 \rightarrow 0.147$). The error signature replicates as well: cat-ranks of non-Cat-Rule Follow errors go from random at base (GOF $p = 0.59/0.67$ in the two non-popular quadrants) to concentrated at cat-rank 2 at ckpt-200 ($p = 1.0\text{e}{-3}$ / $5.7\text{e}{-12}$; Cramér's $V = 0.157/0.102$, stronger than Qwen's $0.127/0.073$).

Table 17: NYC Construction B on Llama-3.1-8B, full test, base → ckpt-200.

		HR@1		am-count		am-cat	
Layer	Quadrant	base	200	base	200	base	200
$W{=}1$	fv+nonpop	.213	.521	.596	.274	.353	.761
$W{=}1$	fv+pop	.751	.852	.966	.986	.836	.952
$W{=}1$	rv+nonpop	.233	.582	.551	.231	.345	.760
$W{=}1$	rv+pop	.751	.873	.810	.903	.772	.921
$W{=}1$	ALL	.442	.692	.677	.517	.526	.829
$W{=}0$	fv+nonpop	.048	.059	.677	.536	—	—
$W{=}0$	fv+pop	.216	.176	.920	.869	—	—
$W{=}0$	rv+nonpop	.070	.070	.652	.494	—	—
$W{=}0$	rv+pop	.455	.327	.708	.565	—	—
$W{=}0$	ALL	.180	.147	.739	.616	—	—

Position-bias controls. Both backbones exhibit output-position bias at base, of opposite character: Llama shows a strong recency bias (Cramer's $V = 0.179$; per-position pick rates spanning a $19\times$ range, A $= 0.005$ vs. F $= 0.094$), Qwen a weaker mid-position preference ($V = 0.109$). Three considerations exclude position bias as the driver of the lock. (i) Candidate order is uniformly shuffled per prompt, so position margins cancel in every aggregate metric we report (under a mixture model with lock-probability c, the per-position pick rate is $c\,q_j + (1-c)/8$ and its mean over j is independent of the shape of q). (ii) The per-position Cat-Rule Follow *gains* are anti-concentrated at the preferred positions (A $+0.443$, B $+0.389$, C $+0.399$ vs. D $+0.290$, F $+0.231$, H $+0.226$); excluding the three most-preferred positions leaves a gain of $+0.357$, above the overall $+0.303$. (iii) Training flattens the bias itself ($W{=}1$ marginal V: $0.167 \rightarrow 0.078$).

A.20 Algorithm replication (DAPO) and a loss-equivalence note

Configuration. Identical Vanilla data, protocol, and hyperparameters as the GRPO baseline; the single change is `loss_type=dapo` with `importance_sampling_level=token` (trl 0.22.2, with $\beta = 0.04$, temperature 0.9, and gradient checkpointing explicitly pinned to the baseline values, since 0.22.2 defaults would otherwise silently alter the protocol). Effective batch 80 completions (16 prompts $\times$ $G{=}5$); evaluated at ckpt-200 on the full test set. **Equivalence note.** In this task 99.87% of completions reach the 192-token generation cap, so sequences are (near-)equal length, and DAPO's token-level aggregation coincides with GRPO's per-sequence aggregation up to a global constant absorbed by the learning rate. This run therefore functions as an independent-seed replication of the full pipeline, and its deviation from the GRPO run (2–3 points; McNemar $p = 1.4\text{e}{-37}$ overall, with DAPO stronger on the non-popular cells and GRPO stronger on the popular cells) provides an estimate of run-level variance. DAPO components that would depart from GRPO on this task (dynamic sampling, asymmetric clipping) were not enabled; they are left to future work. **Results.** Table 18. The lock-in signature is reproduced in kind: count-rule execution jumps ($0.809 \rightarrow 0.883$), the popular quadrants gain, the non-popular quadrants collapse, and the error signature matches (non-argmax errors concentrated at count-ranks 2–3; Cramer's V 0.926–0.947 vs. GRPO's 0.945–0.953). The silent-group fraction rises to 0.422 (GRPO: 0.460; base: 0.095).

Table 18: NYC Vanilla, GRPO vs. DAPO at ckpt-200, full test (N=36,834).

	HR@1			Count-Rule Follow		
Quadrant	base	GRPO	DAPO	base	GRPO	DAPO
fv+nonpop	.049	.027	.032	.815	.905	.879
fv+pop	.777	.868	.847	.832	.924	.902
rv+nonpop	.121	.067	.082	.775	.882	.856
rv+pop	.804	.910	.880	.817	.925	.894
Overall	.459	.495	.485	.809	.910	.883

A.21 Inverse-Success Advantage Calibration (IAC)

Method. IAC modifies only the group-relative advantage. After the standard GRPO normalisation $\hat{A}_i = (r_i - \bar{r})/(\sigma + \epsilon)$, let $S^+ = \{i : \hat{A}_i > 0\}$ be the successful rollouts of a group of size G; IAC rescales

$$\hat{A}_i \;\leftarrow\; \hat{A}_i \cdot (G - |S^+|)^\alpha \qquad \text{for } i \in S^+ \,, \tag{7}$$

leaving non-positive advantages unchanged ($\alpha = 1$ throughout). The construction is an on-policy analogue of inverse-propensity reweighting: a rollout whose success is rare within its group is up-weighted in proportion to that rarity, counteracting the success-frequency bias of the group-mean baseline. **Protocol.** Identical to the Vanilla baseline (data, reward, LoRA r=32/α=64, lr 5e$-$6, G=5, effective batch 80 completions); the advantage change is the only modification. Evaluated at checkpoint 200 on the full test set. Training diagnostics: the mean success count over all groups is $|S^+| \approx 1.2$ (silent groups included), but positive rollouts concentrate in multi-success groups—the average positive rollout sits in a group with $|S^+| \approx 2.8$—so the mean calibration factor among positive rollouts is ≈ 2.2. **Results.** Table 4 includes the IAC row. Relative to GRPO at the same checkpoint, IAC slightly *improves* the aligned quadrants (first-visit+popular $0.8679 \to 0.8710$; re-visit+popular $0.9102 \to 0.9153$) and slightly *worsens* the misaligned ones ($0.0268 \to 0.0245$; $0.0666 \to 0.0593$); count-rule execution on the non-popular quadrants deepens ($0.9048 \to 0.9098$; $0.8815 \to 0.8970$), while the silent fraction moves from 0.460 to 0.402. **Diagnosis.** The failure is structural, and diagnostic for the reweighting class. The multiplier $(G - |S^+|)$ amplifies whichever rollouts already hold positive advantage; on aligned prompts those are the heuristic's own executions, so the lock's fuel is enriched rather than cut. On misaligned prompts, where §3.4 locates the cost, IAC rescales the coefficients of equation 3 but leaves the sampling distribution—and with it the coherence structure—untouched: it cannot reinforce a heuristic-beating behaviour that no rollout in the group exhibits. HRA avoids exactly this failure by moving the advantage's *centre*, which changes *which* rollouts are positive rather than how much the positive ones count.

A.22 Rank-stratified and per-quadrant results for HRA

Flip decomposition. The net non-popular gain of HRA over GRPO reported in §4.2 decomposes into 136 flips up (0.78% of the 17,470 non-popular prompts) against 38 flips down (0.22%). The asymmetry holds in both non-popular quadrants: first-visit 37 up / 12 down (exact McNemar $p = 4.7 \times 10^{-4}$); re-visit 99 up / 26 down ($p = 3.2 \times 10^{-11}$). **Stratification by target count-rank.** Table 19 stratifies the non-popular results by the correct candidate's count-rank. The rank-2 subset (N=7,472; 42.8% of non-popular) gains $+1.18$ points of HR@1 over GRPO (paired-bootstrap 95% CI $[+0.84, +1.51]$; McNemar 124 up / 36 down, $p = 1.6 \times 10^{-12}$), whereas ranks ≥ 3 (N=9,998) gain $+0.10$ points ($[+0.03, +0.18]$; 12 up / 2 down, $p = 0.013$). The difference-in-differences is $+1.08$ points ($[+0.74, +1.43]$). A Fisher exact test on the discordant pairs does not reject equal flip asymmetry across the two subsets ($p = 0.74$): the stratification is visible in the rates, not in the composition of the discordant pairs. The small rank-$\geq$3 gain is nominally significant but rests on 14 discordant pairs, and we do not interpret it further.

Tokyo Vanilla. Table 20 reports the same comparison on Tokyo Vanilla (full test set, N=85,356, 345 users; all inference cluster-robust by user). The misaligned quadrants recover ($+0.76$ and $+1.53$ points; pooled non-popular $+1.14$ points, 549 flips up and 140 down, cluster $p < 10^{-5}$), recovering 22% (first visit) and 26% (revisit) of GRPO's loss relative to base. The recovery concentrates at

Table 19: Non-popular quadrants of NYC Vanilla (N=17,470), stratified by the correct candidate's count-rank. ΔHR = HRA-200 minus GRPO Vanilla-200 with paired-bootstrap 95% CI; McNemar exact on discordant pairs (up = GRPO wrong → HRA right, dn = reverse). IAC-200 shown for reference.

	HR@1				HRA − GRPO	
Subset	base	GRPO	IAC	HRA	ΔHR [CI]	McNemar up/dn (p)
rank = 2 (N=7,472)	.1824	.1045	.0940	.1163	+.0118 [+.0084, +.0151]	124/36 (1.6e−12)
rank ≥ 3 (N=9,998)	.0101	.0028	.0023	.0038	+.0010 [+.0003, +.0018]	12/2 (.013)

count-rank 2 (+2.39 points, 95% CI [+2.1, +2.7]) against +0.18 points at ranks ≥ 3; the difference between the two is itself significant (+2.20 points, cluster bootstrap CI [+1.9, +2.5]). The rank-≥3 gain is nominally significant but its absolute rate (0.0035) is about 3% of the random baseline, and single-seed run-level noise cannot be excluded. The aligned quadrants hand back 23% and 27% of GRPO's gain while remaining well above base. Training logs fall within the NYC range: 0.518 of groups fall back to group-mean centring, 0.035 of popular groups contain a positive-advantage rollout, and 0.447 of non-popular groups do.

Table 20: Tokyo Vanilla, full test set (N=85,356), step 200. HR@1 by quadrant.

	fv+np	rv+np	fv+pop	rv+pop
base	.0589	.1059	.8028	.8349
GRPO	.0248	.0469	.9164	.9476
HRA	.0324	.0622	.8907	.9175

A.23 TKY-B: RANK-RESOLVED RESCUE UNDER TWO ANCHORS

Setup. Two HRA arms on TKY-B, with the NYC-identical protocol and evaluation at step 200 on the full test set (N=85,356; all inference cluster-robust by user, 345 clusters). The count anchor is the same center as in NYC Vanilla (r_H computed from visit counts); the category anchor is the identical code path with the center computed from category frequencies (i.e. the heuristic that ranks candidates by category frequency, ties broken uniformly). Table 5 in the main text gives the headline; Table 21 gives the full stratification.

Table 21: TKY-B, full test (N=85,356), step 200, by W layer and quadrant. CF = Count-Rule Follow, CatF = Category-Rule Follow (reported on W=1 only, as in the main text).

		HR@1					Follow	
Layer	Model	fv+np	rv+np	fv+pop	rv+pop	ALL	CF	CatF
W=1	base	.183	.259	.651	.730	.550	.668	.643
	GRPO-B	.382	.435	.894	.951	.761	.775	.879
	HRA, count	.235	.291	.866	.931	.697	.818	.803
	HRA, cat	.251	.312	.857	.922	.698	.802	.805
W=0	base	.078	.100	.333	.534	.226	.670	—
	GRPO-B	.044	.063	.296	.505	.191	.680	—
	HRA, count	.042	.060	.356	.599	.221	.765	—
	HRA, cat	.046	.066	.344	.580	.218	.745	—

Flips. Against GRPO-B on the W=0 layer (N=39,549): the category anchor flips 1,738 up / 669 down (net +2.70 points, cluster CI [+2.39, +3.05], cluster $p < 10^{-4}$); the count anchor flips 1,992 up / 796 down (net +3.02 points, [+2.68, +3.38], $p < 10^{-4}$). On W=1 the mirror image: net −6.35 points (category anchor) and −6.46 points (count anchor), both cluster-significant. The two anchors rescue W=0 to within 0.5–0.9 points of the base level (0.226) and pay the same aggregate cost on W=1.

Rank-resolved rescue. Table 22 splits the rescue and the cost by the target's count-rank. Two patterns. First, the rescue lives at count-ranks 1–2 and nowhere else: at ranks ≥ 3 neither anchor gains (category -0.19 points, cluster $p = 0.097$; count -0.65 points, $p < 10^{-4}$). Second, the anchors differ in *where* their strength concentrates, the count anchor is stronger at rank 1, where the count rule is correct, and the category anchor at rank 2. At rank 1, where the count rule is correct, the count anchor gains $+7.31$ points against the category anchor's $+5.88$; at rank 2 the order reverses ($+1.94$ category vs. $+1.31$ count). The $W{=}1$ cost mirrors this: the count anchor loses least at rank 1 (-2.19 vs. -3.08), the category anchor at ranks 2 and ≥ 3 (-10.21/-14.14 vs. -11.51/-16.58). The anchor therefore does not decide *whether* the lock breaks but *which* rule's performance is protected.

Table 22: TKY-B by target count-rank, full test, step 200. Δ = HR@1 minus GRPO-B in points; up/dn = discordant flips against GRPO-B with cluster-robust McNemar p (user clusters).

		HR@1				Δ vs. GRPO-B (up/dn, p)	
Layer	Rank	base	GRPO-B	HRA-count	HRA-cat	HRA-count	HRA-cat
$W{=}0$	1 (N=16,910)	.409	.375	.448	.434	+7.3 (1643/407, $<10^{-4}$)	+5.9 (1371/376, $<10^{-4}$)
	2 (N=5,505)	.197	.119	.132	.139	+1.3 (244/172, 2e−4)	+1.9 (241/134, $<10^{-4}$)
	$\geq$3 (N=17,134)	.056	.032	.025	.030	−0.7 (105/217, $<10^{-4}$)	−0.2 (126/159, .097)
$W{=}1$	1 (N=30,003)	.713	.939	.917	.908	−2.2 (446/1104, $<10^{-4}$)	−3.1 (349/1272, $<10^{-4}$)
	2 (N=6,306)	.318	.499	.384	.397	−11.5 (225/951, $<10^{-4}$)	−10.2 (189/833, $<10^{-4}$)
	$\geq$3 (N=9,498)	.188	.373	.207	.231	−16.6 (88/1663, $<10^{-4}$)	−14.1 (89/1432, $<10^{-4}$)

A failed specificity prediction, for the record. Under the base policy (offline, 500 training prompts), the fraction of prompt groups containing a rollout that beats the anchor was much higher for the category anchor on $W{=}0$ (0.757) than for the count anchor (0.475); we therefore expected, before training, that the category anchor would rescue $W{=}0$ more strongly. It does not (0.218 vs. 0.221). The observed symmetry is itself evidence for the main-text reading: what breaks the lock is the fixed anchor itself, not its match to the locked feature.

A.24 Sign structure of the level term: training-time diagnostics

Sign structure. By equation 5, the level term of a group is proportional to $\big(\bar{r} - r_H(q)\big)\,\bar{w}$. Where the anchor's heuristic is correct and its top candidate untied, r_H equals the maximal attainable reward (target placed first, full format marks, and the deterministic bonus evaluated at that position), so $\bar{r} \leq r_H$: the level term is non-positive, and it vanishes exactly when every rollout executes the heuristic. Execution is therefore never rewarded; deviation is penalised in proportion to the group's shortfall. Ties are the single exception: with e candidates tied at the maximum, r_H is an expectation over uniform tie-breaking, and a rollout that places the target strictly first beats it. Tied maxima cover 4.5% of NYC Vanilla popular training prompts, consistent with the residual positive-advantage popular groups below. Ignoring the correlation of σ with the samples, independence of rollouts and $\mathbb{E}[w_i] = 0$ give $\mathbb{E}\big[\sum_i (r_i - \bar{r})\, w_i\big] = (G-1)\nabla J$ and $\mathbb{E}\big[\sum_i (r_i - r_H(q))\, w_i\big] = G\,\nabla J$, so in expectation re-anchoring rescales the update by $G/(G-1)$; its effect on training therefore arises from finite groups, where $\bar{w}$ follows the outputs actually sampled.

Diagnostics. Table 23 collects the training-log quantities for all four HRA runs. The asymmetry claimed in §4.3 holds on both cities: groups containing a rollout that beats the anchor are rare where the anchor's heuristic is correct (0.034–0.035 of popular groups on NYC and Tokyo Vanilla) and common where it is wrong (0.456/0.447 of non-popular groups), while 36–54% on average to group-mean centring in every run. On TKY-B the count anchor's log split is by the (count-based) popularity flag, which is not the aligned/misaligned split of that construction (the lock is on category); we report it for completeness. The category-anchor arm logged only the fallback fraction.

Table 23: Training-log diagnostics of the four HRA runs (means over logged steps; fallback shown with its range). Pos-adv group = group containing at least one rollout with $r_i > r_H(q)$.

Run	Anchor	Fallback frac	Pos-adv, popular	Pos-adv, non-popular
NYC Vanilla	count	0.54 (0.45–0.64)	0.034	0.456
TKY Vanilla	count	0.52 (0.36–0.65)	0.035	0.447
TKY-B	count	0.37 (0.26–0.49)	0.108	0.570
TKY-B	category	0.36 (0.24–0.49)	—	—