LEARNED DRIFT ON THE LOGIT SUBSPACE FOR LLM REASONING EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning from verifiable rewards (RLVR) can only reinforce what it samples, which makes the rollout group the bottleneck: the policy can progress no further than the spread of rewards in the group allows, and only along the trajectories it has already sampled. Existing exploration methods push on both axes blindly, temperature scaling, entropy regularization, and looser clipping widening the sampling distribution in every direction at once while group filtering simply samples harder. We introduce **DRIFT** (**D**irected **R**ollout with **I**nferred **F**eedback at **T**riggers), which applies a Langevin step to steer generation itself, to produce groups with diverse rewards and trajectories. The optimal drift is the gradient of expected reward in logit space, an expectation over completions that no rollout can evaluate before it terminates. With a first-order expansion, we show that a step along any partially aligned direction earns the optimal drift's gain scaled by the cosine between the two. DRIFT therefore estimates such a direction online, from prior steering feedback, using the entropy remaining at branch points as its signal. Training on Qwen3-4B-Base and Qwen3-8B-Base, DRIFT improves reasoning performance under both GRPO and DAPO, raising the benchmark average by 1.5 to 3.1 points in every pairing and AIME accuracy by up to 4.6 points.

1 INTRODUCTION

RLVR is the primary way to elicit strong reasoning from pretrained language models (Guo et al., 2025; OpenAI, 2024): for each problem the policy samples a group of rollouts, each rollout is scored by a verifiable reward, and the policy is updated toward the ones that succeed. Group-Relative Policy Optimization (GRPO) (Shao et al., 2024) is the standard instance, replacing the learned value network of actor-critic methods with the within-group reward spread as its advantage estimate. The update on a prompt is therefore a reweighting of exactly the rollouts drawn for it, so what the policy can learn is bounded by what it happens to sample. On hard problems, where correct rollouts are rare, the standard remedy is to sample more broadly: raise the temperature, add an entropy bonus (Mnih et al., 2016; Schulman et al., 2017), or widen the clipping range (Yu et al., 2025). Each adds stochasticity to the sampler without any signal about where reward lies. We argue, however, that undirected noise is a blunt primitive: a group that produces a useful gradient must be both diverse and correct, and spreading noise uniformly improves the former far more readily than the latter.

Two properties determine how useful a group is. It should contain both correct and incorrect rollouts, since uniform reward gives zero advantage and no correctness gradient. Its correct rollouts should also be different solutions, since a group that repeats one solution only sharpens the mode the policy already prefers: entropy falls, the next group is drawn from a narrower distribution, and the policy settles into the first solution it found. Existing remedies satisfy these conditions one at a time. Undirected noise keeps rollouts apart but knows nothing about reward, so on hard problems it yields groups that are varied and uniformly wrong. Group curation such as DAPO (Yu et al., 2025) resamples a prompt until its group carries reward variance, buying the first condition with additional rollout compute, but the rollouts still come from the same sampler, so the correct ones it keeps concentrate in a single family of solutions that the policy then refines until there is nothing left to gain. Both act on rollouts they have already drawn, by adding noise before generation or selecting afterwards. What neither does is steer the generation itself: bias the sampler, as each token is produced, toward the continuations more likely to end in a correct solution. This is the *directed* exploration we argue is missing.

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.

Steering generation this way requires biasing the sampler toward reward without suppressing the randomness that makes rollouts differ, since a group needs both. Langevin dynamics is built around precisely this balance, pairing a reward-gradient drift with a noise term so that sampling concentrates on higher-reward regions while retaining coverage (Roberts & Tweedie, 1996; Welling & Teh, 2011). Applied to the next-token logits it is the primitive we want, with one obstacle: the drift is *the gradient of expected downstream reward*, which threads through a stochastic trajectory of thousands of tokens and cannot be computed during generation. Our observation is that it need not be computed. A first-order expansion shows that the gain from a step falls off only with its alignment to the true gradient, so every direction on the correct side of the gradient earns a share of the oracle's gain and the estimate has to land in a cone rather than on a point. That is a problem a rollout can solve as it runs, from evidence it is already producing about which perturbations keep it productive. The drift can be *learned online*, as the trajectory unfolds.

We introduce **DRIFT** (**D**irected **R**ollout with **I**nferred **F**eedback at **T**riggers), which learns *how to explore* while it explores. Where prior work treats exploration as a fixed quantity of noise to be tuned, DRIFT treats the exploration direction as a quantity estimated online from the rollout in progress. At the branch points where the model is weighing a few competing continuations, DRIFT perturbs the logits along a single direction that mixes this learned drift with noise. The learned component steers toward continuations favored by the feedback, while independent noise preserves variation across rollouts; whether the resulting groups are both correct and diverse is evaluated in Section 3. The drift is estimated from a signal the rollout supplies before any reward is known, the entropy still remaining at its branch points, on the reasoning that a solution can only be reached while it is still reachable, so a trajectory that gives up its alternatives too early gives up the reward with them. Because the perturbed sampler remains an exactly evaluable softmax, the resulting off-policy trajectories return to the update through an importance correction rather than being discarded. DRIFT therefore governs how a group is generated rather than which groups survive, which lets it sit on top of group-relative algorithms generally and compose with curation instead of replacing it.

Contributions.

- We identify a limitation shared by current RLVR exploration: because the update flows only through the sampled group, perturbing the sampler without regard to reward, or curating groups after they are drawn, leaves the generation itself unsteered and the groups it produces correct in only one way.
- We propose **DRIFT**, which steers generation with a learned direction in a Langevin-style logit perturbation at the branch points it identifies by effective candidate count, and returns the off-policy trajectories through an *advantage correction* that leaves the policy-gradient ratio intact. DRIFT acts on how a group is generated, so it applies to group-relative algorithms generally and composes with group curation.
- We give a local analysis of branch-point steering. An exact branch-value identity shows that, at matched perturbation magnitude, learned steering has a first-order advantage over random steering precisely when its direction is value-aligned. An exact rollout decomposition aggregates the signed local effects, and prompt-wise group analysis maps any resulting single-rollout gain into changes in coverage and mixedness.
- We show empirically across Qwen3 base models that DRIFT draws groups that are more often correct and more textually varied among their correct rollouts, translating to faster learning and stronger held-out reasoning under both GRPO and DAPO.

2 DRIFT: DIRECTED ROLLOUT WITH INFERRED FEEDBACK AT TRIGGERS

Section 1 argued that a useful rollout group must mix correct and incorrect rollouts, with the correct ones differing from one another, and that neither undirected noise nor group curation reliably supplies both because neither steers generation itself. DRIFT does. This requires deciding *where* to intervene (Section 2.2), *how* to perturb the sampler (Section 2.3), and which direction to perturb along before the reward is known (Section 2.4). Because the perturbed trajectories are off-policy, the update must also correct for the behavior-policy mismatch (Section 2.5). Algorithm 1 combines these into one rollout procedure; Algorithm 2 (Appendix B) places it in training, and Appendix C lists all symbols.

2.1 PRELIMINARIES

Group-relative policy optimization. GRPO (Shao et al., 2024) is an actor-only reinforcement learning (RL) algorithm for language-model fine-tuning that eliminates the need for a learned value network. Given a prompt x^i, n rollouts $\{y^{i,j}\}_{j=1}^{n}$ are sampled from the current policy and each is scored by a verifiable reward $r^{i,j}$. The within-group normalized advantage $\hat{A}^{i,j} = \frac{r^{i,j} - \mathrm{mean}_{j'}\, r^{i,j'}}{\mathrm{std}_{j'}\, r^{i,j'} + \epsilon_{\mathrm{std}}}$ replaces a learned critic, and the policy is updated by the clipped proximal policy optimization (PPO) objective with $\hat{A}^{i,j}$ as the advantage signal. Two properties of this update matter for what follows. The advantage is measured against the group mean, so a group whose rollouts all receive the same reward gives $\hat{A}^{i,j} = 0$ and $\nabla_\theta \mathcal{L}_{\mathrm{GRPO}}$ vanishes; and the update reweights the sampled rollouts themselves, so a solution absent from the group cannot be reinforced. The group a prompt receives therefore determines what that prompt can teach the policy.

Langevin dynamics on the logits. Steering generation means biasing the sampler toward reward without suppressing the randomness that makes rollouts differ, and Langevin dynamics is built around exactly this balance. Applied to the next-token logits $l_t \in \mathbb{R}^{|V|}$,

$$\tilde{l}_t = l_t - \eta \nabla E(l_t) + \sigma \zeta_t, \qquad \zeta_t \sim \mathcal{N}(0, I), \tag{1}$$

the drift $-\eta \nabla E(l_t)$ concentrates sampling on lower-energy, that is higher-reward, continuations, while the diffusion $\sigma \zeta_t$ preserves coverage. The two terms play complementary roles: the drift pushes rollouts toward correctness, the diffusion keeps them from collapsing onto one another. Acting in logit space rather than on the probability simplex matters here, since the softmax map keeps the perturbed distribution inside the same parametric family as π_θ, with an exactly evaluable log-density that makes the off-policy correction in Section 2.5 possible.

Every term in Eq. (1) is available at decoding time except the one that does the steering. **The energy of a logit vector is the negative expected reward of what follows from it,** $E(l_t) = -\mathbb{E}_{\tau \sim \pi(\cdot|l_t)}[R(\tau)]$**, so** $\nabla E(l_t)$ **is determined by trajectories of thousands of tokens that have not been generated yet.** At the moment we must choose a direction, the quantity that defines the right direction does not yet exist. This circularity is what stands between Langevin dynamics as a principle and Langevin dynamics as an RL algorithm.

The drift can be evaluated via a surrogate rather than exactly computed. The circularity is an obstruction only if the step requires $\nabla E(l_t)$ exactly. It does not. Write $u^\star = -\nabla E(l_t)/\|\nabla E(l_t)\|$ for the *oracle* direction, the one we would use if the expectation could be evaluated, and let $\Delta(u)$ denote the first-order gain in expected reward from a step of size ρ along an arbitrary unit direction u, of which $u^\star$ is one particular choice. A first-order expansion gives

$$\Delta(u) \;=\; \underbrace{\rho\,\|\nabla E(l_t)\|}_{\Delta(u^\star)} \cdot \underbrace{\langle u, u^\star \rangle}_{a(u)} \;=\; \Delta(u^\star)\, a(u), \qquad a(u) \in [-1, 1].$$

The direction enters only through the scalar $a(u)$, its alignment with the oracle, and the gain is simply the oracle's gain scaled by it. A surrogate lower bound follows immediately:

$$\underbrace{\Delta(u^\star)}_{\substack{\text{oracle} \\ a=1}} \;\geq\; \underbrace{a\,\Delta(u^\star)}_{\substack{\text{any direction with} \\ a(u) \geq a > 0}} \;>\; \underbrace{\mathbb{E}\big[\Delta(u_{\mathrm{rand}})\big]}_{\substack{\text{isotropic noise} \\ = 0}}. \tag{2}$$

The bound separates two regimes. Any direction with alignment at least $a > 0$ attains at least an a-fraction of the oracle gain, whereas an isotropic direction attains zero in expectation. The exact drift is thus not required: it would suffice to estimate a direction G_t with $a(G_t) > 0$. DRIFT uses an entropy-based estimate.

DRIFT constructs a candidate steering direction through a zeroth-order update. At each selected point, it perturbs the logits along ϵ and continues decoding; the following branch returns an entropy-maintenance score s, producing the update $s\epsilon$. Momentum averages these noisy updates along the trajectory. This yields the drift G_t, whose magnitude is not used by the sampler. Positive alignment with $u^\star$ is the condition required by Eq. (2), rather than a property guaranteed by the entropy score; Section 3 and Appendix F characterize this condition. Using G_t as the steering component yields DRIFT's Langevin-style logit perturbation. What remains is to say which sampling points to use (Section 2.2), what form the perturbation takes (Section 2.3), and what score to read back from it (Section 2.4).

2.2 WHERE TO INTERVENE: BRANCH POINTS

A rollout offers many sampling points, but not every decoded token is one of them. At most positions the policy is effectively deterministic, completing a phrase, closing a bracket, or carrying a digit with nearly all its mass on a single token. A perturbation there changes the logits without changing what is sampled, so the score that returns says nothing about the direction applied, and the update $s \cdot \epsilon$ moves G_t on noise. The usable positions are the *branch points*, where the policy is weighing a handful of genuine alternatives and the token it commits to sets the direction of the reasoning that follows.

DRIFT finds them by the effective number of candidates at t, computed from $p_t = \mathrm{softmax}(l_t)$:

$$N_{\text{eff}}(t) = \Big(\sum_{v \in V} p_t(v)^2 \Big)^{-1}, \qquad M_t = \mathbf{1}[N_{\min} \le N_{\text{eff}} \le N_{\max}], \tag{3}$$

DRIFT perturbs only where $M_t = 1$, with $N_{\min} = 2$ and $N_{\max} = 8$. Lemma 2 in Appendix F.3 shows that the upper bound on the attainable first-order effect vanishes as $N_{\text{eff}} \to 1$ or ∞ and is maximized at $N_{\text{eff}} = 2$, where it equals $1/\sqrt{2}$. At $N_{\text{eff}} = 8$, the bound is $1/\sqrt{8}$, exactly half of this maximum, motivating the practical trigger range $[2, 8]$; the upper cutoff remains a modeling choice. Intuitively, the lower threshold excludes positions where the policy has already committed to one continuation, while the upper threshold excludes positions where probability mass is too diffuse for a bounded perturbation to have substantial leverage.

Algorithm 1 DRIFT rollout generation. The policy is queried exactly as in standard decoding; DRIFT intervenes only at branch points, where it updates the drift from the previous branch point's outcome and perturbs the logits before sampling. Lines marked ▷ are the only departures from ordinary autoregressive generation.

Require: Prompt x, policy π_θ, branch-point range $[N_{\min}, N_{\max}]$, subspace size K, step σ, mix α, entropy target α_{target}, momentum γ
Ensure: Trajectory $\tilde{y}$, branch-point mask M, and the log-probabilities $\log \tilde{\pi}_{\text{old},t}, \log \pi_{\text{old},t}$ needed by Eq. (7)

$\mathcal{T} \leftarrow \emptyset$; $H_{\text{first}} \leftarrow \perp$; $\epsilon_{\text{prev}} \leftarrow 0$; $G \leftarrow 0$ {▷ drift is local to this rollout}
for $t = 1, \dots, T$ **do**
  $l_t \leftarrow f_\theta(x, \tilde{y}_{<t})$; $p_t \leftarrow \mathrm{softmax}(l_t)$
  $N_{\text{eff}} \leftarrow \big(\sum_v p_t(v)^2\big)^{-1}$ {▷ effective number of candidates, Eq. (3)}
  if $N_{\min} \le N_{\text{eff}} \le N_{\max}$ **then**
    $H_t \leftarrow -\sum_v p_t(v) \log p_t(v)$
    if $H_{\text{first}} = \perp$ **then**
      $H_{\text{first}} \leftarrow H_t$ {▷ anchor the entropy target to the first branch point}
    end if
    if $\mathcal{T} \neq \emptyset$ **then**
      $s_t \leftarrow H_t - \alpha_{\text{target}} H_{\text{first}}$ {▷ score the *previous* perturbation}
      $G \leftarrow \gamma G + (1-\gamma)\, s_t\, \epsilon_{\text{prev}}$ {▷ one-point zeroth-order optimization, Eq. (6)}
    end if
    $\mathcal{K}_t \leftarrow \mathrm{TopK}_{v \in V \setminus \{\text{EOS}\}}(l_t(v), K)$
    Sample $\xi_t \sim \mathcal{N}(0, I_K)$
    if $\|G\| > 0$ **then**
      $u_t \leftarrow \alpha\, G/\|G\| + (1-\alpha)\, \xi_t/\|\xi_t\|$
      $\epsilon_t \leftarrow u_t/\|u_t\|$ {▷ directed step, Eq. (4)}
    else
      $\epsilon_t \leftarrow \xi_t/\|\xi_t\|$ {▷ cold start: undirected until G has signal}
    end if
    $\tilde{l}_t \leftarrow l_t$; $\tilde{l}_t[\mathcal{K}_t] \leftarrow l_t[\mathcal{K}_t] + \sigma\sqrt{K}\,\epsilon_t$ {▷ Eq. (5)}
    Sample $\tilde{y}_t \sim \mathrm{Categorical}(\mathrm{softmax}(\tilde{l}_t))$; $M_t \leftarrow 1$
    Store $\log \tilde{\pi}_{\text{old},t}$ and $\log \pi_{\text{old},t}$ at $\tilde{y}_t$ {▷ both exactly evaluable}
    $\epsilon_{\text{prev}} \leftarrow \epsilon_t$; $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$
  else
    Sample $\tilde{y}_t \sim \mathrm{Categorical}(p_t)$; $M_t \leftarrow 0$ {not a branch point: ordinary decoding}
  end if
end for
return $\tilde{y}$, M, $\{\log \tilde{\pi}_{\text{old},t}, \log \pi_{\text{old},t}\}_{t:M_t=1}$

2.3 HOW TO PERTURB: A LANGEVIN-STYLE STEERING STEP ON THE LOGITS

At a branch point DRIFT applies a Langevin-style steering step motivated by Eq. (1), with the learned drift G_t supplying its directional component. The step is built in two stages.

Choose what to move toward. The step combines the two things a group needs. The drift G_t provides learned steering toward continuations the feedback has favored, while independent draws of ξ_t preserve variation across rollouts that share a prefix. Both enter as unit vectors, so α sets how much of each the sampler follows:

$$u_t = \alpha\, \frac{G_t}{\|G_t\|} + (1-\alpha)\, \frac{\xi_t}{\|\xi_t\|}, \qquad \epsilon_t = \frac{u_t}{\|u_t\|}, \qquad \xi_t \sim \mathcal{N}(0, I_K). \tag{4}$$

Normalizing the blend keeps α from changing the step length, so it sets only the direction and σ below sets the distance. Eq. (1) scales its two terms separately, by η and σ; writing the step as a direction and a distance replaces that pair with a ratio and a magnitude, which is the form the estimator can supply, since G_t is an estimate of where $u^\star$ points and not of how far $\nabla E(l_t)$ reaches.

Apply it to the sampler. The step enters rollout generation as a modification of the decoding step: at a branch point the policy emits l_t as usual, DRIFT displaces it, and the next token is drawn from the displaced distribution. Writing $v[I] \in \mathbb{R}^{|I|}$ for the sub-vector of a vector v indexed by a coordinate set I, and $\mathcal{K}_t$ for the K coordinates of l_t carrying the most mass,

$$\tilde{l}_t[\mathcal{K}_t] = l_t[\mathcal{K}_t] + \sigma\sqrt{K}\,\epsilon_t, \qquad \tilde{y}_t \sim \mathrm{softmax}(\tilde{l}_t). \tag{5}$$

Since ϵ_t is a unit vector in K dimensions, its coordinates are of size $1/\sqrt{K}$, and the factor $\sqrt{K}$ restores σ as the displacement a single logit receives, so that changing K widens the subspace the step acts on without also shrinking the step. We use $K = 128$ and $\sigma = 0.1$; Appendix B gives the remaining settings.

2.4 Where the Direction Comes From

Recall that Eq. (2) requires far less than the oracle drift. Any direction with $a(u) > 0$ secures an a-fraction of the oracle's gain, so the admissible directions form a whole cone about $u^\star$ and an estimator would only need to land inside it, not to recover $\nabla E(l_t)$.

Part of $E(l_t) = -\mathbb{E}_{\tau\sim\pi(\cdot|l_t)}[R(\tau)]$ survives the intractability. It is the reward that is out of reach, not the distribution it is averaged over: $\pi(\cdot \mid l_t)$ is available at decoding time, with one-step marginal p_t. Its support is the set of solutions still reachable from l_t, and the solution the rollout eventually returns is drawn from it, so the reward the oracle chases lies inside the region p_t describes. That region only narrows: committing to a token at a branch point selects one of the subspaces the alternatives led to and discards the rest, and the entropy $H_t = -\sum_{v\in V} p_t(v)\log p_t(v)$ reports how much of it is left.

Containment is what makes the rate of that shrinkage worth steering. Lose the support too early and the trajectory fixes on a solution while its prefix is still too short to tell which is right, so it is both likely to be wrong and likely to land where its siblings landed; hold them open too long and the policy branches until the budget is gone with nothing resolved. The rollouts worth having fall between, resolving steadily while several correct solutions remain in play.

DRIFT therefore holds exploration inside that region, taking a fixed fraction of the entropy at a trajectory's first branch point as the floor to hold, and reading H_t as the outcome of the direction $\epsilon_{t'}$ of Eq. (4) applied at t', the branch point before t:

$$s_t = H_t - \alpha_{\mathrm{target}} H_{\mathrm{first}}, \quad G \leftarrow \gamma G + (1-\gamma)\, s_t\, \epsilon_{t'}, \tag{6}$$

with $\alpha_{\mathrm{target}} = 0.7$ and $\gamma = 0.7$. The drift is revised only where the score is read, so the G_t of Eq. (4) is the value in force at t, the one last revised at t'. Each rollout keeps its own Gs, so no drift is shared across the rollouts of a group or carried across training steps. The sign of s_t steers: below the floor the update turns G away from the perturbation that closed the region, above it turns G toward the one that kept it open, and the ceiling of Eq. (3) prevents the loop from acting at positions beyond $N_{\max}$, with γ damping the velocity G carries across branch points.

Holding that floor supplies a local steering signal: a negative score moves the update away from a perturbation followed by premature entropy loss, whereas a positive score reinforces a perturbation followed by retained branch entropy. Turning this signal into reward improvement additionally requires the direction carried to later branch points to align with their continuation values. Section 3 and Appendix F isolate this condition and evaluate its end-to-end consequence.

When the score is near zero, the update contributes little new directional information, while the fresh random component of Eq. (4) continues to diversify sampling. Because Eq. (4) normalizes G_t, decay of its magnitude alone does not change the steering–noise mixture while $G_t \neq 0$.

2.5 Returning the Trajectories to the Update

Tokens at branch points were sampled from the displaced $\tilde{\pi}_{\text{old}}$ of Eq. (5) rather than from π_{old}, so an uncorrected update would read the drift's preference as the policy's own. DRIFT corrects the group-normalized advantage $\hat{A}^{i,j}$ of Section 2.1 rather than the PPO ratio $\rho_t^{i,j} = \pi_\theta(\tilde{y}_t^{i,j} \mid x^i, \tilde{y}_{<t}^{i,j})/\pi_{\text{old}}(\tilde{y}_t^{i,j} \mid x^i, \tilde{y}_{<t}^{i,j})$, which carries the gradient $\nabla_\theta \log \pi_\theta$ and cannot be rescaled without distorting the quantity being optimized:

$$w_t = \begin{cases} \dfrac{\pi_{\text{old},t}(\tilde{y}_t)}{\tilde{\pi}_{\text{old},t}(\tilde{y}_t)}, & M_t = 1, \\ 1, & M_t = 0, \end{cases} \qquad \hat{A}_t^{i,j} \leftarrow w_t \, \hat{A}^{i,j}. \tag{7}$$

The weight down-weights tokens the drift made more likely and up-weights those it suppressed. It is the exact likelihood ratio of its own token under the history-dependent behavior policy, so any baseline cancels exactly; omitting the cumulative product over earlier perturbed tokens introduces a bias linear in their total variation and avoids a variance that grows geometrically in the number of branch points (Appendix F.8). A token's contribution to the policy gradient is proportional to $\hat{A}\rho$, so scaling $\hat{A}$ leaves ρ and the clipping untouched and the objective remains the usual clipped surrogate:

$$\mathcal{L}_{\text{DRIFT}}(\theta) = \mathbb{E}_{i,j,t}\Big[\min\big(\rho_t^{i,j}\, w_t \hat{A}^{i,j}, \text{clip}(\rho_t^{i,j}, 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})\, w_t \hat{A}^{i,j}\big)\Big]. \tag{8}$$

Nothing in Eq. (8) is specific to GRPO, so the correction applies unchanged to other RL algorithms such as DAPO, whose group curation then operates on groups DRIFT has made more informative. Directed exploration composes with group curation rather than competing with it.

3 Why DRIFT for RLVR Exploration?

Existing approaches to stalled RLVR exploration either enlarge the sampling distribution, by raising the temperature, adding an entropy bonus (Mnih et al., 2016; Schulman et al., 2017), or widening the clipping range (Yu et al., 2025), or curate groups after they are drawn, as DAPO (Yu et al., 2025) does by filtering zero-advantage groups and resampling prompts. Both rest on an implicit assumption: that the rollouts a policy would have sampled anyway are the right raw material, and that exploration is a matter of drawing more of them or keeping the better ones. We believe this assumption is what limits them.

A group provides a correctness contrast only when it contains at least one correct and one incorrect rollout. Its correct rollouts should also be *diverse*, rather than one solution resampled. The second condition is easy to overlook because its failure is silent: a group that repeats a single solution still produces a gradient, but that gradient only sharpens the mode the policy already prefers, so entropy falls, the next group is drawn from a narrower distribution, and the policy converges onto the first solution it found while better alternatives are never sampled. We measure the first condition by *coverage*, the fraction of prompts whose group contains at least one correct rollout, and the second by *correct-only diversity*, the mean pairwise distance among a group's correct rollouts. Neither enlarging nor filtering an undirected sampling distribution reliably delivers both. We ask what can be established about a perturbed sampler's mechanism, and then what it is measured to deliver.

3.1 What the Mechanism Analysis Shows

The analysis separates five questions: where a perturbation can have first-order influence, what distinguishes learned steering from random steering at matched perturbation magnitude, what information the entropy feedback contains, how the local effects accumulate over a complete rollout, and how a rollout-level gain changes the quality of a group. Let $\mathcal{G}$ be the set of rewarding trajectories, $Q_t(v)$ the probability of reaching it after emitting v at t and then following the unperturbed decoder, and

$$V_t(d) = \mathbb{E}_{v \sim \text{softmax}(l_t + P_K^\top d)}\big[Q_t(v)\big]$$

the *branch value* under a perturbation d of the top-K logits, so $V_t(0)$ is the value of leaving the branch point alone. Write $\rho_{\text{step}} = \sigma\sqrt{K}$ for the step size of Eq. (5) and define the value alignment $A_t =$

Table 1: Two measurements on the 8B policy. (a) Groups drawn at matched budget from the step-180 DRIFT checkpoint on the same 200 training prompts, once with standard GRPO rollouts and once with DRIFT rollouts, $n = 8$ rollouts each at temperature 1.0; coverage is the fraction of prompts whose group holds a correct rollout, mixed the fraction holding both a correct and an incorrect one, and diversity the mean pairwise normalized token-level Levenshtein distance. (b) Two runs from the same base model, one trained with GRPO and one with DRIFT, reported as DRIFT minus GRPO on the training pass rate, over which GRPO itself rises from 30.0% to 34.3%, and on held-out AIME24 mean@16.

(a) Group quality

Rollout	**GRPO**	**DRIFT**
Solve rate (%)	35.1	**39.1**
Coverage (%)	62.0	**65.5**
Mixed group (%)	51.0	**52.5**
Diversity, all	0.658	**0.720**
Diversity, correct only	0.590	**0.642**

(b) Order of the gains

	DRIFT − GRPO	
Step	**Train %**	**AIME24**
80	−0.5	+0.9
120	+0.4	−1.4
140	+3.8	+3.9
180	**+5.8**	**+4.1**

$\langle G_t/\|G_t\|, \nabla V_t(0)\rangle$ when $G_t \neq 0$. Because V_t is a linear functional of a softmax, $\partial V_t/\partial d_k(0) = p_t(v_k)(Q_t(v_k) - V_t(0))$ exactly; the required local smoothness is supplied by this softmax structure rather than imposed as an additional assumption. Appendix F treats the first and third questions: Lemma 2 bounds the attainable first-order effect by N_{eff}, and Lemma 4 decomposes the entropy feedback. We state the other two results here.

Proposition 1 (Value alignment determines the first-order steering gap)**.** *Let $d_t = \rho_{\text{step}}\epsilon_t$ with ϵ_t from Eq. (4), and let $m(\alpha)$ be the mean projection of ϵ_t on $G_t/\|G_t\|$, which is strictly increasing with $m(0) = 0$ and $m(1) = 1$. Then*

$$\mathbb{E}\, V_t(d_t)\big|_\alpha - \mathbb{E}\, V_t(d_t)\big|_{\alpha=0} = \rho_{\text{step}}\, m(\alpha)\, A_t + O(\rho_{\text{step}}^2), \tag{9}$$

while the random-steering arm $\alpha = 0$ has no first-order term at all: $\mathbb{E}\, V_t(d_t) - V_t(0) = \frac{\rho_{\text{step}}^2}{2K}\,\text{tr}\,\nabla^2 V_t(0) + O(\rho_{\text{step}}^3)$. Thus, whenever the first-order term dominates the local curvature term, the sign of the matched-magnitude gap is the sign of A_t.

The step size scales the first-order effect, while value alignment determines its sign. The proposition is conditional: it identifies what a useful direction must satisfy, without asserting that the entropy estimator always produces such a direction. Improvement over a rollout requires the carried directions to have positive aggregate alignment with the value gradients where they are used; Appendix F.6 makes that condition explicit.

Theorem 1 (Rollout gain and group coverage)**.** *Fix a prompt and suppose conditionally independent rollouts succeed with probability p under the base sampler and $p + \Delta$ under DRIFT, with $0 \leq p \leq p + \Delta \leq 1$. For $C_n(p) = 1 - (1 - p)^n$, the coverage gain of a group of n rollouts satisfies*

$$n(1 - p - \Delta)^{n-1}\Delta \leq C_n(p + \Delta) - C_n(p) \leq n(1 - p)^{n-1}\Delta. \tag{10}$$

For small p and Δ, the coverage gain is approximately $n\Delta$; the multiplier contracts as success becomes common. The appendix also shows that the probability of a mixed group increases when the interval from p to $p + \Delta$ remains below the single-rollout success threshold $1/2$, the hard-prompt regime. None of these results establishes that the correct rollouts are distinct solution strategies. Correct-only diversity remains an empirical property measured below.

3.2 What the Mechanism Delivers

A policy can only reinforce a solution it is still able to sample, so the groups it draws during training are the ceiling on what any objective can teach it. We therefore measure DRIFT where that ceiling is set: in the group that reaches the update and in what the policy does with it.

Rollout-group quality. Table 1a fixes the step-180 DRIFT checkpoint and draws groups from it twice, once with standard GRPO rollouts and once with DRIFT rollouts. Only the rollout procedure differs. Both use matched budget, the same 200 training prompts, $n = 8$ rollouts each at temperature 1.0, and the training verifier. Correctness improves on all three counts, solve rate $35.1 \rightarrow 39.1$,

coverage $62.0 \rightarrow 65.5$, and mixed groups $51.0 \rightarrow 52.5$, which together say the groups are not only more often right but more often *learnable*: a group with no correct rollout has nothing correct to reinforce, and a mixed group provides direct correctness contrast under group normalization. Diversity improves alongside them, $0.658 \rightarrow 0.720$, and restricted to the *correct* rollouts alone it still rises, $0.590 \rightarrow 0.642$, so the additional textual spread

is present among successful completions rather than only among failures. Correctness and textual diversity move up together, which is the pair the two conditions of Section 1 ask for.

Training and held-out transfer. Table 1b trains the same base model twice, once with GRPO and once with DRIFT. DRIFT is slightly *behind* on training pass rate at step 80 (-0.5), level at 120 ($+0.4$), and ahead from 140 on, reaching $+5.8$ by step 180. A single better group is worth little on its own, and the early steps show as much; the gain accumulates as the groups keep feeding the update, and it is the accumulation that separates the two runs. The held-out column is mixed over the same early stretch, $+0.9$ at step 80 and -1.4 at step 120, and positive from 140.

The two measurements give a consistent account. The DRIFT checkpoint draws groups that are more often correct and more textually varied among correct rollouts, and the trained policy eventually gains on both training and held-out evaluation. Both measure downstream behavior and do not by themselves identify every step of the causal chain. Section 4 evaluates DRIFT against GRPO and DAPO under this account.

4 Experiments

Section 3 measured DRIFT at the level of a single group. Here we ask whether groups drawn that way are worth their cost over a full training run, and whether the answer survives a change of scale and a change of the algorithm underneath.

4.1 Setup

Models and training. We train Qwen3-4B-Base and Qwen3-8B-Base (Team, 2025) on the math subset of the Guru reasoning corpus (Cheng et al., 2025), with rule-based binary correctness rewards and no reward shaping. All runs use $8\times$H200 GPUs through verl (Sheng et al., 2025), n=8 rollouts per prompt at temperature 1.0, AdamW at learning rate 1×10^{-6}, and batch size 1024. **Base algorithms.** We pair DRIFT with two group-relative algorithms that differ in how they treat a group after it is drawn. GRPO (Shao et al., 2024) takes each group as sampled. DAPO (Yu et al., 2025) curates it, filtering zero-advantage groups and resampling, with asymmetric clipping and token-level loss normalization. Because DRIFT changes how a group is generated rather than which groups survive, it should compose with the second as readily as with the first, and the two pairings test exactly that. Within a pairing, the base run and the +DRIFT run share data, optimizer, batch size, temperature, and reward function, and differ only in whether branch-point steering is active during rollout. **Evaluation.** Every configuration is trained with five different seeds, and we report the mean across seeds with its standard deviation, both taken at the same training step as in Section 3.2. Within a seed we report mean@16 at temperature 1.0 on AIME 25, AIME 24, MATH-500, the `OE_TO_maths_en_COMP` subset of OlympiadBench (He et al., 2024), and GPQA-Diamond (Rein et al., 2023). The first four are mathematics at descending difficulty; GPQA-Diamond is graduate-level science multiple choice and carries no mathematics training signal, so it measures whether the gains stay inside the trained domain. Appendix B gives the remaining details.

4.2 Main Results

DRIFT improves the average in every pairing (Table 2), by $+1.52$ and $+3.08$ points over GRPO at 4B and 8B and by $+1.84$ and $+1.97$ over DAPO. The two runs in a pairing differ only in the rollout intervention. AIME 24 improves by 21.0% to 28.3% and AIME 25 by 8.1% to 20.8%. The DAPO gains suggest that steering how groups are drawn complements selecting among groups already drawn (Section 2.5). GPQA-Diamond, which carries no mathematics training signal, rises in three of the four pairings and falls by 1.45 in 4B DAPO. At DRIFT's trigger positions on Qwen3-4B-Base, random steering, an entropy bonus, and KL-matched temperature reach 29.49, 30.40, and 30.03, against 32.79 for DRIFT and 31.27 for GRPO (Appendix D.2).

Table 2: mean@16 accuracy (%) after training, for each base algorithm run with and without DRIFT. The two runs in a pairing differ only in whether branch-point steering is active during rollout. Subscripts give the standard deviation. Each Δ row gives the absolute change in accuracy points, with a superscript for the same change as a fraction of the base algorithm's own score, so $+2.82^{28.3\%}$ reads as 2.82 points on top of a base of 9.98.

Model	Method	AIME25	AIME24	MATH500	Olympiad	GPQA-D	AVG
Qwen3-4B-Base	GRPO	$13.79_{\pm 1.20}$	$9.98_{\pm 1.10}$	$65.23_{\pm 0.86}$	$28.16_{\pm 0.79}$	$39.19_{\pm 1.36}$	$31.27_{\pm 0.84}$
	+DRIFT	$\mathbf{16.00}_{\pm 1.60}$	$\mathbf{12.80}_{\pm 1.00}$	$\mathbf{66.12}_{\pm 0.89}$	$\mathbf{28.63}_{\pm 0.87}$	$\mathbf{40.39}_{\pm 1.58}$	$\mathbf{32.79}_{\pm 1.02}$
	Δ	$+2.21^{16.0\%}$	$+2.82^{28.3\%}$	$+0.89^{1.4\%}$	$+0.47^{1.7\%}$	$+1.20^{3.1\%}$	$+1.52^{4.9\%}$
Qwen3-4B-Base	DAPO	$17.21_{\pm 1.96}$	$13.98_{\pm 1.12}$	$68.52_{\pm 0.98}$	$30.71_{\pm 0.83}$	$\mathbf{35.95}_{\pm 1.15}$	$33.27_{\pm 0.96}$
	+DRIFT	$\mathbf{20.75}_{\pm 1.57}$	$\mathbf{16.91}_{\pm 0.98}$	$\mathbf{71.11}_{\pm 1.13}$	$\mathbf{32.27}_{\pm 0.85}$	$34.50_{\pm 1.37}$	$\mathbf{35.11}_{\pm 1.00}$
	Δ	$+3.54^{20.6\%}$	$+2.93^{21.0\%}$	$+2.59^{3.8\%}$	$+1.56^{5.1\%}$	$-1.45^{-4.0\%}$	$+1.84^{5.5\%}$
Qwen3-8B-Base	GRPO	$22.08_{\pm 1.62}$	$18.66_{\pm 1.22}$	$70.02_{\pm 0.89}$	$33.30_{\pm 0.97}$	$43.39_{\pm 1.09}$	$37.49_{\pm 0.84}$
	+DRIFT	$\mathbf{26.67}_{\pm 1.74}$	$\mathbf{22.83}_{\pm 1.08}$	$\mathbf{72.54}_{\pm 1.02}$	$\mathbf{35.36}_{\pm 0.86}$	$\mathbf{45.46}_{\pm 1.00}$	$\mathbf{40.57}_{\pm 0.95}$
	Δ	$+4.59^{20.8\%}$	$+4.17^{22.3\%}$	$+2.52^{3.6\%}$	$+2.06^{6.2\%}$	$+2.07^{4.8\%}$	$+3.08^{8.2\%}$
Qwen3-8B-Base	DAPO	$27.92_{\pm 1.81}$	$20.60_{\pm 0.98}$	$75.52_{\pm 1.02}$	$37.09_{\pm 0.95}$	$45.69_{\pm 1.01}$	$41.36_{\pm 0.91}$
	+DRIFT	$\mathbf{30.17}_{\pm 1.62}$	$\mathbf{25.16}_{\pm 1.22}$	$\mathbf{76.29}_{\pm 0.89}$	$\mathbf{38.31}_{\pm 0.81}$	$\mathbf{46.73}_{\pm 0.95}$	$\mathbf{43.33}_{\pm 1.03}$
	Δ	$+2.25^{8.1\%}$	$+4.56^{22.1\%}$	$+0.77^{1.0\%}$	$+1.22^{3.3\%}$	$+1.04^{2.3\%}$	$+1.97^{4.8\%}$

5 Related Work

DRIFT relates to several lines of work; see Appendix A for extended discussion. GRPO (Shao et al., 2024) and DeepSeek-R1 (Guo et al., 2025) scale RLHF-style training (Ouyang et al., 2022) to verifiable reasoning rewards, and DAPO (Yu et al., 2025) stabilizes early training; DRIFT is orthogonal, modifying rollout *generation* to widen reward variance within groups while leaving the policy-gradient update unchanged. Entropy bonuses (Haarnoja et al., 2018) apply at every token, and curiosity (Pathak et al., 2017) and count-based rewards (Bellemare et al., 2016) define intrinsic rewards from novelty or visitation, which are hard to measure over token prefixes; DRIFT instead perturbs the logits only at high-entropy branch points. SGLD (Welling & Teh, 2011) and ULA (Roberts & Tweedie, 1996) perturb model *parameters* to sample from a posterior; DRIFT instead applies a Langevin-structured perturbation to the *token logit space* during rollout, with a direction updated online from entropy feedback rather than from an estimate of the energy gradient. Test-time tree-search methods (Lightman et al., 2024; Snell et al., 2025) require a separate value network and an explicit tree; DRIFT uses neither, and perturbs individual rollouts at branch points without backtracking or selecting among continuations. Finally, DRIFT applies an importance correction to the advantage at triggered positions, leaving the policy-gradient ratio and its clipping untouched.

6 Conclusion

In RLVR the policy learns only from the group of rollouts it happens to sample, so the quality of that group bounds what any amount of optimization can extract from a prompt. We introduced **DRIFT**, which improves the group by steering generation itself: at the branch points where the model is weighing a few competing continuations, it perturbs the logits along a direction that mixes a drift learned online from the rollout in progress with isotropic noise, and returns the resulting off-policy trajectories through an importance correction on the advantage. A local, first-order analysis shows that the gain of a step at a branch point scales with its alignment to the optimal drift rather than with perturbation magnitude. Empirically, DRIFT draws groups that are more often correct and more textually varied among correct rollouts, translating to faster learning and stronger held-out reasoning under both GRPO and DAPO. Because DRIFT changes how a group is generated rather than which groups are kept, it composes with group curation instead of competing with it. Limitations and future work appear in Appendix E.

AI USE STATEMENT

Generative AI tools were used only to polish the writing of this paper; the authors are responsible for all content.

ETHICS STATEMENT

This work studies rollout generation for reinforcement learning with verifiable rewards on reasoning tasks and does not involve human subjects, personal data, or new datasets. All benchmarks used are publicly released.

REPRODUCIBILITY STATEMENT

Section 2 specifies the branch-point trigger, the mixed perturbation direction, the online drift estimator, and the corrected objective, and Algorithms 1 and 2 give step-by-step pseudocode for the rollout and the training step. Section 4.1 and Appendix B report the base models, training corpus, reward function, evaluation benchmarks and metrics, hardware, optimizer, batch sizes, sequence lengths, and every DRIFT hyperparameter; within each pairing, the base and +DRIFT runs share all settings except branch-point steering. All models, training data, and evaluation benchmarks are publicly available, and training builds on the open-source verl framework. Appendix F states the assumptions and gives complete proofs for every theoretical result in Section 3. We are unable to release code at submission time because it is pending our organization's legal approval; we will release the code upon approval should the paper be made public.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021. URL https://arxiv.org/abs/2108.07732.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1471–1479, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/afda332245e2af431fb7b672a68b659d-Abstract.html.

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

Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Yonghao Zhuang, Nilabjo Dey, Yuheng Zha, Yi Gu, Kun Zhou, Yuqi Wang, Yuan Li, Richard Fan, Jianshu She, Chengqian Gao, Abulhair Saparov, Haonan Li, Taylor W. Killian, Mikhail Yurochkin, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. Revisiting reinforcement learning for LLM reasoning from A cross-domain perspective. *CoRR*, abs/2506.14965, 2025. doi: 10.48550/ARXIV.2506.14965. URL https://doi.org/10.48550/arXiv.2506.14965.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, Sep 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pp. 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.211. URL https://doi.org/10.18653/v1/2024.acl-long.211.

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In Claude Sammut and Achim G. Hoffmann (eds.), *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pp. 267–274. Morgan Kaufmann, 2002.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd*

International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, JMLR Workshop and Conference Proceedings, pp. 1928–1937. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/mniha16.html.

OpenAI. Openai o1 system card. *CoRR*, abs/2412.16720, 2024. doi: 10.48550/ARXIV.2412.16720. URL https://doi.org/10.48550/arXiv.2412.16720.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research, pp. 2778–2787. PMLR, 2017. URL http://proceedings.mlr.press/v70/pathak17a.html.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 759–766, 2000.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022, 2023. doi: 10.48550/ARXIV.2311.12022. URL https://doi.org/10.48550/arXiv.2311.12022.

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341 – 363, 1996.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pp. 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In Lise Getoor and Tobias Scheffer (eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 681–688. Omnipress, 2011. URL https://icml.cc/2011/papers/398_icmlpaper.pdf.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476, 2025. doi: 10.48550/ARXIV.2503.14476. URL https://doi.org/10.48550/arXiv.2503.14476.

APPENDIX

This appendix is organized as follows. Appendix A provides a more detailed review of related work. Appendix B describes the experimental setup, including the datasets, evaluation metrics, and implementation details. Appendix C summarizes the notation used throughout the paper. Appendix D reports additional experiments. Appendix E discusses the limitations of our approach and potential directions for future work. Appendix F provides theoretical insights into the properties of the DRIFT algorithm.

A EXTENDED RELATED WORK

Reinforcement learning for LLM reasoning. Reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) established the paradigm of fine-tuning language models with RL, using PPO (Schulman et al., 2017) against a learned reward model trained from human preferences. In the reasoning setting, verifiable binary rewards replace the learned reward model and group-relative baselines replace the value network. GRPO (Shao et al., 2024) formalizes this approach, and DeepSeek-R1 (Guo et al., 2025) scales it to achieve strong mathematical reasoning through RL on verifiable correctness signals. DAPO (Yu et al., 2025) addresses early-training instability with dynamic group filtering, asymmetric clipping, token-level loss normalization, and overlong reward shaping. Our contribution is orthogonal: DRIFT modifies the rollout *generation* procedure to increase reward variance within groups, after which any standard GRPO update performs the optimization.

Exploration in policy gradient methods. Maximum-entropy RL augments the reward with an entropy bonus to encourage stochastic behavior (Haarnoja et al., 2018). In the LLM setting, entropy bonuses operate uniformly across all token positions and can cause entropy growth or collapse depending on the reward signal (Yu et al., 2025). Curiosity-driven exploration (Pathak et al., 2017) and count-based methods (Bellemare et al., 2016) define intrinsic rewards from novelty or visitation statistics; these do not naturally extend to the autoregressive setting, where states are token prefixes of exponentially many distinct strings. DRIFT instead perturbs the logits only at high-entropy branch points, and updates its direction to keep entropy at later branch points above a floor rather than letting it collapse early.

Stochastic gradient Langevin dynamics. Stochastic gradient Langevin dynamics (SGLD) (Welling & Teh, 2011) injects Gaussian noise into stochastic gradient descent to approximate posterior sampling over model *parameters*. The unadjusted Langevin algorithm (ULA; Roberts & Tweedie 1996) and its variants characterize convergence of discrete Langevin chains in parameter space. DRIFT applies a Langevin-structured perturbation to the *token logit space* during rollout generation, not to the model parameters. Its direction is updated online from entropy feedback (Eq. (6)) rather than from an estimate of the energy gradient.

Test-time compute and tree search. Process reward models (Lightman et al., 2024) and explicit tree-search decoding (Snell et al., 2025) use a learned value function to evaluate and select among intermediate reasoning steps, enabling richer exploration at inference time. These approaches require a separate trained reward model and explicit tree construction. DRIFT uses neither. The Langevin perturbation at a branch point changes the continuation of a single rollout without backtracking, and its direction G_t is updated from entropy feedback within the trajectory, not from a value estimate. DRIFT does not evaluate or select among sibling continuations, so its connection to tree search is loose.

Off-policy importance sampling. Importance-sampling corrections for off-policy policy gradient are standard in RL (Precup et al., 2000). DRIFT applies an importance correction to the advantage at triggered branch positions (Section 2.5), leaving the policy-gradient ratio and its clipping untouched.

Algorithm 2 DRIFT training step

Require: Batch of prompts $\{x^i\}_{i=1}^{B}$, old policy π_{old}, reference policy π_{ref}, group size n
Ensure: Updated policy parameters θ

```
for each prompt x^i do
    for j = 1, ..., n do
        Generate (ỹ^{i,j}, M^{i,j}, log π̃^{i,j}) with Algorithm 1
        Score rollout with verifiable reward r^{i,j}
    end for
    Compute Â^{i,j} by normalizing r^{i,j} within the group
end for
for each token t in each rollout (i, j) do
    Compute ρ_t^{i,j} = π_{θ,t}(ỹ_t^{i,j}) / π_{old,t}(ỹ_t^{i,j})
    if M_t^{i,j} = 1 then
        w_t ← π_{old,t}(ỹ_t^{i,j}) / π̃_{old,t}(ỹ_t^{i,j}) {▷ Eq. (7)}
    else
        w_t ← 1
    end if
    Accumulate the clipped term in Eq. (8)
end for
Update θ by gradient ascent on Eq. (8)
return θ
```

B Experimental Setup Details

Base models. We use Qwen3-4B-Base and Qwen3-8B-Base (Team, 2025), transformer base models pre-trained on general text with no instruction or reasoning tuning, with the Qwen3 chat template throughout.

Datasets. Training uses the math subset of the Guru reasoning corpus (Cheng et al., 2025), comprising competition-style AIME-family and MATH-family problems with rule-based rewards via a sympy-equivalence checker, yielding a binary $\{0, 1\}$ correctness signal with no shaping. We evaluate on four benchmarks spanning the difficulty and domain spectrum: AIME 24 and AIME 25 (30 problems each) for in-distribution competition mathematics; the `OE_TO_maths_en_COMP` single-answer subset of OlympiadBench (He et al., 2024) (674 problems) for harder out-of-distribution math; and GPQA-Diamond (Rein et al., 2023) (198 problems) for cross-domain transfer to graduate-level science MCQ. To prevent contamination, we filter out of the training set any problem that appears in an evaluation set. AIME 24 is evaluated in-training every 5 steps via the rollout logger; the full external suite is run as a post-hoc sweep on saved checkpoints at every 20-step boundary.

Implementation. All training runs use 8×H200 (140 GB) GPUs through verl (Sheng et al., 2025), with FSDP for the actor and vLLM for the rollout engine. We sample $n = 8$ rollouts per prompt during training and $n = 16$ per problem during evaluation, both at temperature 1.0; metrics are mean@16 (per-sample solve rate) and, for AIME, additionally best@16 (probability that at least one of 16 samples is correct). Optimization uses AdamW with learning rate 1×10^{-6}, weight decay 0.1, and 10 warmup steps; batch size 1024 prompts per training step, mini-batch 256, micro-batch 16 per GPU; maximum prompt length 1024 tokens, maximum response length 4096 tokens.

DRIFT hyperparameters. Branch points are selected by the effective-candidate range of Eq. (3) with $N_{\min} = 2$ and $N_{\max} = 8$. The step of Eq. (5) uses top-K subspace $K = 128$ and scale $\sigma = 0.1$, and the perturbation is applied to the logits in fp32. Drift estimation uses entropy-floor fraction $\alpha_{\text{target}} = 0.7$ and momentum $\gamma = 0.7$, as in Eq. (6). The clip range follows the base algorithm, with or without DRIFT: GRPO uses $\epsilon_{\text{low}} = \epsilon_{\text{high}} = 0.2$, and DAPO uses clip-higher with $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. The KL coefficient is $\lambda_{\text{KL}} = 0.001$.

C NOTATION

Symbol	Meaning
Inputs and policy.	
x	Prompt.
y_t, $\tilde{y}_t$, $y_{<t}$	Token sampled from the unperturbed policy, the DRIFT-perturbed rollout policy, and the token prefix before position t.
h_t	Autoregressive context $(x, y_{<t})$ or $(x, \tilde{y}_{<t})$.
T	Maximum rollout length.
V	Vocabulary; $\lvert V\rvert$ is vocabulary size.
θ	Trainable policy parameters.
f_θ	Language-model forward map from context to logits.
l_t, $\tilde{l}_t$	Original and perturbed next-token logits at position t.
p_t	Unperturbed next-token distribution $\mathrm{softmax}(l_t)$.
π_θ, $\tilde{\pi}_\theta$	Unperturbed policy and DRIFT rollout policy.
π_{old}, π_{ref}	PPO/GRPO behavior snapshot and fixed reference policy for KL regularization.
Branch points and Langevin rollout.	
H_t	Shannon entropy of $\pi_\theta(\cdot \mid x, y_{<t})$.
$N_{\mathrm{eff}}(t)$	Effective number of candidate tokens at t, $\left(\sum_v p_t(v)^2\right)^{-1}$, in Eq. (3).
$N_{\min}$, $N_{\max}$	Lower and upper thresholds on $N_{\mathrm{eff}}(t)$ that define a branch point; $N_{\min} = 2$, $N_{\max} = 8$.
$\mathcal{T}$, M_t	Set of branch-point positions and binary trigger mask of Eq. (3).
K, $\mathcal{K}_t$	Number of perturbed logit coordinates and the top-K index set at t; $K = 128$.
P_K	Projection onto the top-K coordinates of the current branch point.
G, G_t	Learned drift, carried in the rank-ordered top-K frame of Appendix F.1; G_t is its value in force at t.
ϵ_t, ξ_t	Mixed unit perturbation direction of Eq. (4), and the Gaussian draw $\xi_t \sim \mathcal{N}(0, I_K)$ whose normalization is uniform on $\mathbb{S}^{K-1}$.
ζ_t	Diffusion term of the generic Langevin step, Eq. (1); DRIFT introduces its noise through ξ_t instead.
η, σ	Drift step size and diffusion scale of Eq. (1)
d_t, ρ_{step}	Top-K displacement $d_t = \sigma\sqrt{K}\,\epsilon_t$ of Eq. (5), and its length $\rho_{\mathrm{step}} = \sigma\sqrt{K}$.
α, γ	Drift-exploitation mixture weight of Eq. (4) and Polyak momentum of Eq. (6); $\alpha = 0.6$, $\gamma = 0.7$.
$m(\alpha)$	Mean projection of ϵ_t on $G_t/\lVert G_t\rVert$, Lemma 3; $m(0.6) = 0.83$.
H_{first}, α_{target}	Entropy at the first branch point of a trajectory and target maintenance fraction; $\alpha_{\mathrm{target}} = 0.7$.
s_t, $\bar{s}$	Entropy-maintenance feedback $H_t - \alpha_{\mathrm{target}} H_{\mathrm{first}}$ of Eq. (6), and the score the unperturbed branch would receive.
GRPO objective.	
i, j, n	Prompt index, rollout index, and number of rollouts per prompt.
$r^{i,j}$, $\tilde{r}^{i,j}$	Verifiable reward and length-adjusted reward of rollout j for prompt i.
$\hat{A}^{i,j}$	Group-normalized GRPO advantage.
$\rho_t^{i,j}$, ρ_t	PPO/GRPO likelihood ratio $\pi_\theta/\pi_{\mathrm{old}}$ at token t.
w_t	Off-policy correction $\pi_{\mathrm{old},t}/\tilde{\pi}_{\mathrm{old},t}$ at trigger positions.
ϵ_{low}, ϵ_{high}	Lower and upper PPO clipping radii.
λ_{KL}	Kullback-Leibler (KL) divergence regularization weight; default 0.001.
Theoretical analysis (Appendix F).	
$\mathcal{G}$	Set of rewarding complete trajectories.
π_d	Rollout distribution after the top-K perturbation d, $\mathrm{softmax}(l_t + P_K^\top d)$.
$Q_t^\pi(h_t, v)$	Probability of reaching $\mathcal{G}$ after emitting v at t and continuing with the unperturbed decoder π.
$V_t(d)$	Branch value $\mathbb{E}_{v\sim\pi_d}[Q_t(v)]$; $V_t(0)$ is the value of leaving the branch point alone.
$A_t(G_t; h_t)$	Unnormalized value alignment $\langle G_t/\lVert G_t\rVert, \nabla V_t(0; h_t)\rangle$ at a branch point.
$Z_{t'}(d)$, $\Phi_{t'}(d)$	Entropy-maintenance feedback attributed to a source perturbation, zero if no later scored branch occurs, and its conditional expectation.

Symbol	Meaning
$\lambda_{\parallel}, \lambda_{\perp}$	Parallel and orthogonal eigenvalues of $\mathbb{E}[\epsilon_t \epsilon_t^\top]$ in the frame of $\hat{G}$, Lemma 4.
$\mathcal{A}(\widetilde{\pi}, \pi)$	Visitation-weighted sum of value alignments encountered by the DRIFT rollout policy relative to the base decoder.
p, Δ, $C_n(p)$	Base single-rollout success probability, its increase under DRIFT, and the probability that an independent group of size n contains a success.
$\mathcal{M}_n(p)$	Probability that an independent group of n rollouts contains both a success and a failure.

D ADDITIONAL EXPERIMENTS

D.1 VALIDATION TRAJECTORIES

Figure 1 tracks held-out accuracy for GRPO and GRPO+DRIFT on the checkpoint sweep of Appendix B, every 20 training steps. Early in training DRIFT is not uniformly ahead. At 4B it is lower on AIME25 from step 60 to 140. At 8B it is lower on AIME24 at five of the first six checkpoints and on GPQA-D over the first 60 steps. DRIFT is ahead on all five benchmarks from step 140 at 8B and from step 160 at 4B. From step 160 the gap is largest on AIME24 at both scales.

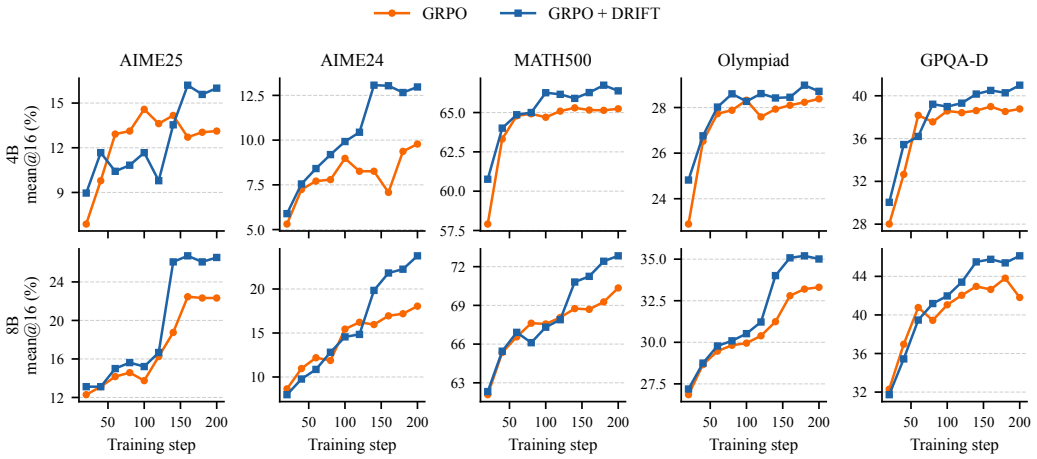

Figure 1: Held-out mean@16 accuracy (%) during training for GRPO and GRPO+DRIFT on Qwen3-4B-Base (top) and Qwen3-8B-Base (bottom), evaluated every 20 steps. Curves show a single training seed, so step-200 values differ from the seed means in Table 2.

D.2 DIRECTED VERSUS UNDIRECTED EXPLORATION

The thesis of DRIFT is that directed exploration beats undirected exploration. This section tests the thesis directly, against four undirected exploration baselines run at the same setup as DRIFT.

Exploration baselines. All runs use the setup of Section 4.1 on Qwen3-4B-Base and are trained from scratch with five different seeds, as in Table 2. *Clip-higher* is the DAPO run of Table 2: DAPO raises the upper clipping bound ($\epsilon_{\text{high}} = 0.28$ against GRPO's 0.2) to keep the policy's entropy from collapsing, and we repeat it here with DRIFT on top. The other three baselines run with GRPO underneath. *Random exploration* sets $\alpha = 0$ in Eq. (4), so the step direction reduces to $\xi_t/\|\xi_t\|$. It perturbs the logits at the same trigger positions of Eq. (3), at the same σ and with the same magnitude, and differs from DRIFT only in the direction it steers. *Entropy bonus* keeps DRIFT's trigger and everything else unchanged, but at each triggered position $t \in \mathcal{T}$ it adds the policy's token-level entropy to the clipped GRPO objective instead of steering the logits,

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{clip}}(\theta) + \beta\, \mathbb{E}_{t\in\mathcal{T}}\big[\mathcal{H}\big(\pi_\theta(\cdot \mid x, y_{<t})\big)\big], \qquad \beta > 0.$$

It acts on the same tokens as DRIFT, but widens the sampling distribution there in every direction rather than displacing it along one. *Temperature* changes only how triggered tokens are drawn. At every position with $M_t = 1$ it samples from

$$\mu_{T,t} = \text{softmax}(l_t/T), \qquad T > 1,$$

and elsewhere it samples from $\text{softmax}(l_t)$ as usual. It raises sampling entropy at exactly DRIFT's positions but carries no learned direction. It uses the same trigger mask as DRIFT. We tune T so that the average KL divergence of $\mu_{T,t}$ from $\pi_{\text{old},t}$ at triggered positions matches that of DRIFT's $\tilde{\pi}_{\text{old},t}$, rather than fixing a nominal temperature. Triggered tokens receive the same correction as in Eq. (7), with $w_t = \pi_{\text{old},t}(\tilde{y}_t)/\mu_{T,t}(\tilde{y}_t)$.

Table 4: mean@16 accuracy (%) of directed (DRIFT) and undirected exploration on Qwen3-4B-Base, all trained from scratch under the setup of Table 2. Random exploration and the entropy bonus act at DRIFT's trigger positions, as does temperature sampling, whose T matches DRIFT's average KL divergence from π_{old}; random exploration uses DRIFT's σ with $\alpha = 0$, so it differs from DRIFT only in steering along a random direction rather than an estimated one.

Method	AIME25	AIME24	MATH500	Olympiad	GPQA-D	AVG
GRPO	$13.79_{\pm 1.20}$	$9.98_{\pm 1.10}$	$65.23_{\pm 0.86}$	$28.16_{\pm 0.79}$	$39.19_{\pm 1.36}$	$31.27_{\pm 0.84}$
+random (α=0)	$10.33_{\pm 1.01}$	$8.62_{\pm 1.15}$	$64.01_{\pm 1.00}$	$27.47_{\pm 0.82}$	$37.04_{\pm 1.21}$	$29.49_{\pm 0.85}$
+entropy bonus	$11.58_{\pm 1.35}$	$7.46_{\pm 1.20}$	$66.33_{\pm 1.02}$	$28.21_{\pm 0.62}$	$38.41_{\pm 1.30}$	$30.40_{\pm 0.90}$
+temperature	$10.58_{\pm 1.05}$	$8.46_{\pm 0.82}$	$64.83_{\pm 1.16}$	$28.01_{\pm 0.42}$	$38.26_{\pm 0.93}$	$30.03_{\pm 0.72}$
+DRIFT	$16.00_{\pm 1.60}$	$12.80_{\pm 1.00}$	$66.12_{\pm 0.89}$	$28.63_{\pm 0.87}$	$40.39_{\pm 1.58}$	$32.79_{\pm 1.02}$
DAPO	$17.21_{\pm 1.96}$	$13.98_{\pm 1.12}$	$68.52_{\pm 0.98}$	$30.71_{\pm 0.83}$	$35.95_{\pm 1.15}$	$33.27_{\pm 0.96}$
+DRIFT	$20.75_{\pm 1.57}$	$16.91_{\pm 0.98}$	$71.11_{\pm 1.13}$	$32.27_{\pm 0.85}$	$34.50_{\pm 1.37}$	$35.11_{\pm 1.00}$

A random direction at the same positions scores lower than DRIFT. A random step at DRIFT's trigger positions reaches 29.49 average points, 1.78 below the GRPO run it perturbs, while DRIFT reaches 32.79 from the same positions at the same step size. The two runs differ only in α, so this comparison suggests that the gain comes from the estimated direction rather than from firing at branch points. The drop below GRPO is in line with the local picture of Proposition 1 at $m(0) = 0$, where a step with no expected alignment pays a second-order cost with no first-order term to offset it. The proposition is local and first-order, however, and does not predict an end-to-end training outcome.

An entropy bonus at the same positions does not match DRIFT. The entropy bonus acts at the same trigger positions as DRIFT and reaches 30.40, 2.39 below DRIFT. It is also 0.87 below the GRPO baseline, a gap comparable to GRPO's seed standard deviation in Table 2. Widening the sampling distribution at a branch point spends the rollout budget on every continuation the policy already considers, in proportion to how likely it already was. This is the undirected exploration that Section 3 argues is unlikely to reach the trajectories a group is missing. Its one advantage over DRIFT is 0.21 points on MATH-500, the benchmark in the set on which the base policy is already strongest and on which the rollout group is least likely to be the binding constraint.

Matching DRIFT's shift in distribution is not enough. Temperature sampling moves the rollout distribution as far from π_{old} as DRIFT does on average, at the same positions and with the same correction. It reaches 30.03, 2.76 below DRIFT and 1.24 below GRPO, and it trails DRIFT on all five benchmarks. The largest gaps are on AIME 25 and AIME 24, by 5.42 and 4.34 points. Under a matched average KL, the two methods differ mainly in where the extra probability mass goes. Temperature spreads it over the tail of the policy, while DRIFT places it along an estimated direction. This comparison suggests that the size of the shift alone does not account for DRIFT's gain.

Directed exploration adds to clip-higher rather than duplicating it. Clip-higher keeps the policy's entropy from collapsing, but leaves the extra spread undirected. DRIFT on top of it raises the average from 33.27 to 35.11, and gains on every math benchmark, by 1.56 to 3.54 points. Preserving entropy keeps more continuations available; DRIFT additionally steers sampling at branch points.

D.3 Transfer to Code Generation

The experiments in Section 4 train on mathematics, where a branch point separates competing lines of algebraic reasoning. Nothing in DRIFT is specific to that domain: Eq. (3) reads the effective number of candidate tokens, and Eq. (6) reads entropy, neither of which refers to what the tokens mean. We ask whether the method still helps when the branch points separate competing implementations instead.

We train Qwen3-4B-Base on the code subset of the Guru corpus (Cheng et al., 2025), stdin/stdout programming problems drawn from PrimeIntellect and TACO, with reward given by test execution. The setup is otherwise that of Section 4.1. We evaluate on HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) every ten training steps.

Table 5: Pass rate (%) on HumanEval and MBPP during code training on Qwen3-4B-Base, evaluated every ten steps.

	HumanEval			MBPP		
Step	**GRPO**	**+DRIFT**	**Δ**	**GRPO**	**+DRIFT**	**Δ**
10	49.43	**50.00**	+0.57	45.58	**48.87**	+3.29
20	53.43	**58.92**	+5.49	48.07	**50.16**	+2.09
30	53.20	**62.84**	+9.64	47.75	**53.98**	+6.23
40	57.03	**66.84**	+9.81	43.35	**57.38**	+14.03
50	53.54	**69.36**	+15.82	45.01	**58.54**	+13.53
60	57.54	**70.01**	+12.47	42.19	**59.49**	+17.30
70	50.69	**71.53**	+20.84	45.17	**58.39**	+13.22
80	55.83	**71.04**	+15.21	51.44	**58.72**	+7.28
90	57.45	**71.26**	+13.81	57.64	**60.36**	+2.72
100	54.34	**70.33**	+15.99	55.88	**60.43**	+4.55

E Limitations and Future Work

Our experiments use a single training corpus, a single base-model family, and binary verifiable rewards.

We have not investigated annealing the drift as training proceeds, a natural extension once correct rollouts become common and Theorem 1 predicts diminishing returns. More broadly, treating the logit space as the right substrate for structured exploration in language-model RL appears to scale and generalize beyond the math-reasoning setting we study here; extending DRIFT to agentic and multi-step interactive tasks, where the dead zone is arguably more severe, is the most promising direction we leave open.

F THEORETICAL ANALYSIS

This appendix analyzes DRIFT's mechanism under explicit local conditions. The results separate five questions that are easy to conflate: where a bounded logit perturbation can have first-order influence; what distinguishes learned steering from random steering at matched perturbation magnitude; what information the entropy feedback contributes to the drift update; how the local effects accumulate over a complete rollout; and how a positive change in single-rollout success changes the quality of a rollout group. The analysis does not assert that entropy is a reward surrogate. Instead, it isolates the value-alignment condition under which the signal used by DRIFT improves reward, and it shows exactly how such aligned local effects accumulate along a rollout and propagate to group coverage and mixedness.

F.1 SETUP

Fix a prompt x, a partial trajectory $y_{<t}$, and the context $h_t = (x, y_{<t})$. Write $l_t = f_\theta(h_t)$ and $p_t = \mathrm{softmax}(l_t)$. Let $\mathcal{K}_t$ be the top-K coordinate set of Eq. (5), let $P_K \in \mathbb{R}^{K\times|V|}$ project onto it, and index its coordinates by $k = 1, \dots, K$, with v_k denoting the corresponding token. For $d \in \mathbb{R}^K$, define

$$\pi_d(\cdot \mid h_t) = \mathrm{softmax}(l_t + P_K^\top d). \tag{11}$$

Let π denote the unperturbed decoder at the same checkpoint and decoding temperature, and let $\mathcal{G}$ be the set of rewarding complete trajectories. Define the unperturbed continuation value

$$Q_t^\pi(h_t, v) = \Pr_\pi[\tau \in \mathcal{G} \mid h_t, \text{ emit } v \text{ at } t] \tag{12}$$

and the branch value

$$V_t(d; h_t) = \mathbb{E}_{v\sim\pi_d(\cdot|h_t)}[Q_t^\pi(h_t, v)]. \tag{13}$$

Thus $V_t(0; h_t)$ is the success probability obtained by sampling the current token without a logit displacement and then following π. We suppress h_t when it is clear.

At a triggered position, Algorithm 1 applies

$$d_t = \rho_{\mathrm{step}}\epsilon_t, \qquad \rho_{\mathrm{step}} = \sigma\sqrt{K}, \qquad \epsilon_t = \frac{\alpha\hat{G}_t + (1-\alpha)\hat{\xi}_t}{\|\alpha\hat{G}_t + (1-\alpha)\hat{\xi}_t\|_2}, \tag{14}$$

where $\hat{G}_t = G_t/\|G_t\|_2$ when $G_t \neq 0$ and $\hat{\xi}_t = \xi_t/\|\xi_t\|_2$ for $\xi_t \sim \mathcal{N}(0, I_K)$. At a cold start, $G_t = 0$ and the algorithm uses $\epsilon_t = \hat{\xi}_t$.

The coordinates of G_t are ranks, rather than token identities: coordinate k is carried from one branch point to the next as the k-th coordinate of the current ordered top-K set. This makes the update well defined, but it does not imply that a useful direction at one branch remains useful at a later one. Cross-branch persistence is part of the alignment quantity isolated in Section F.6, rather than a consequence of the coordinate identification.

F.2 THE GRADIENT OF BRANCH VALUE

Lemma 1 (Exact gradient of branch value)**.** *$V_t(\cdot; h_t)$ is real analytic and, for every $k \in \{1, \dots, K\}$,*

$$\frac{\partial V_t}{\partial d_k}(0; h_t) = p_t(v_k)\big(Q_t^\pi(h_t, v_k) - V_t(0; h_t)\big). \tag{15}$$

Proof. The map $d \mapsto \pi_d$ is the composition of an affine map and softmax, hence is real analytic. Since V_t is a finite linear functional of π_d, it is real analytic as well. Using the softmax Jacobian gives

$$\begin{aligned}\frac{\partial V_t}{\partial d_k}(0; h_t) &= \sum_{v\in V} Q_t^\pi(h_t, v)p_t(v)\big(\mathbf{1}[v = v_k] - p_t(v_k)\big) \\ &= p_t(v_k)\big(Q_t^\pi(h_t, v_k) - V_t(0; h_t)\big).\end{aligned}$$

□

The identity separates probability mass from continuation advantage. A token can affect branch value to first order only when the policy gives it nontrivial mass and its continuation value differs from the branch average.

F.3 What the Trigger Excludes

Lemma 2 (The trigger bounds the attainable first-order effect)**.** *Let* $N = N_{\text{eff}}(t) = (\sum_v p_t(v)^2)^{-1}$. *Then*

$$\|\nabla V_t(0)\|_2 \le \min\left\{\sqrt{2}\cdot\left(1-\frac{1}{N}\right), \frac{1}{\sqrt{N}}\right\}. \tag{16}$$

The displayed upper bound tends to zero as $N \to 1$ *and as* $N \to \infty$*, and is largest at* $N = 2$*.*

Proof. Put $\Delta(v) = Q_t^\pi(h_t, v) - V_t(0; h_t)$. Since both terms are probabilities, $|\Delta(v)| \le 1$. Lemma 1 therefore gives

$$\|\nabla V_t(0)\|_2^2 \le \sum_{k=1}^{K} p_t(v_k)^2 \le \sum_{v\in V} p_t(v)^2 = \frac{1}{N}.$$

For the other branch, let $v_\star$ be a most likely token, $p_{\max} = p_t(v_\star)$, and $\varepsilon = 1 - p_{\max}$. Because $\sum_v p_t(v)^2 \le p_{\max}$, we have $\varepsilon \le 1 - 1/N$. Moreover,

$$\left|Q_t^\pi(h_t, v_\star) - V_t(0; h_t)\right| \le \varepsilon.$$

If $v_\star \in \mathcal{K}_t$, its coordinate contributes at most ε^2 and all others at most $\sum_{v\neq v_\star} p_t(v)^2 \le \varepsilon^2$; if it is excluded, all perturbed coordinates obey the latter bound. Hence $\|\nabla V_t(0)\|_2 \le \sqrt{2}\varepsilon \le \sqrt{2}\cdot(1-1/N)$. The two branches meet at $N = 2$. □

This is an exclusion result. It shows that positions far toward either extreme have little available first-order effect under any unit direction. It does not show that every position inside $[N_{\min}, N_{\max}]$ is useful, nor does it determine the optimal thresholds; those remain modelling choices.

F.4 Value Alignment at a Branch Point

For $G_t \neq 0$, define the unnormalized value alignment

$$A_t(G_t; h_t) = \left\langle \hat{G}_t, \nabla V_t(0; h_t)\right\rangle. \tag{17}$$

Unlike a cosine, A_t remains well defined when the branch-value gradient is zero. At a cold start we set $A_t(0; h_t) = 0$.

Lemma 3 (Mean of the mixed direction)**.** *For* $G_t \neq 0$ *and* $\alpha \in [0, 1)$*, put* $\beta = \alpha/(1-\alpha)$*. Then* $\mathbb{E}[\epsilon_t \mid G_t] = m(\alpha)\hat{G}_t$*, where*

$$m(\alpha) = \mathbb{E}_c\left[\frac{\beta + c}{\sqrt{\beta^2 + 2\beta c + 1}}\right], \qquad c = \langle \hat{\xi}_t, \hat{G}_t\rangle. \tag{18}$$

The function m *extends continuously to* $[0, 1]$ *and is strictly increasing, with* $m(0) = 0$ *and* $m(1) = 1$*. For* $\alpha = 0$*,* $\mathbb{E}[\epsilon_t] = 0$ *and* $\mathbb{E}[\epsilon_t\epsilon_t^\top] = I_K/K$*.*

Proof. Decompose $\hat{\xi}_t = c\hat{G}_t + \sqrt{1-c^2}\, w$, where $w \perp \hat{G}_t$ is a unit vector. Conditional rotational symmetry gives $\mathbb{E}[w \mid c] = 0$, yielding Eq. (18). If $f(\beta, c) = (\beta + c)(\beta^2 + 2\beta c + 1)^{-1/2}$, then

$$\frac{\partial f}{\partial \beta} = \frac{1 - c^2}{(\beta^2 + 2\beta c + 1)^{3/2}} > 0$$

almost surely for $K \ge 2$. The endpoint statements follow by symmetry and by taking $\beta \to \infty$. A uniform unit vector has second moment I_K/K. □

Proposition (At matched step size, value alignment determines the first-order steering gap (restated from Section 3.1, Proposition 1))**.** *Conditional on* h_t *and* $G_t \neq 0$*,*

$$\mathbb{E}[V_t(d_t; h_t) \mid \alpha, G_t] - \mathbb{E}[V_t(d_t; h_t) \mid \alpha = 0] = \rho_{\text{step}} m(\alpha) A_t(G_t; h_t) + O(\rho_{\text{step}}^2). \tag{19}$$

and, against no perturbation,

$$\mathbb{E}[V_t(d_t; h_t) \mid \alpha, G_t] - V_t(0; h_t) = \rho_{\text{step}} m(\alpha) A_t(G_t; h_t) + O(\rho_{\text{step}}^2). \tag{20}$$

The random-steering arm has no first-order term:

$$\mathbb{E}[V_t(d_t; h_t) \mid \alpha = 0] - V_t(0; h_t) = \frac{\rho_{\text{step}}^2}{2K}\operatorname{tr}\nabla^2 V_t(0; h_t) + O(\rho_{\text{step}}^3). \tag{21}$$

Proof. Taylor expansion at $d = 0$ gives

$$V_t(d) = V_t(0) + \langle \nabla V_t(0), d\rangle + \tfrac{1}{2} d^\top \nabla^2 V_t(0) d + O(\|d\|_2^3).$$

Softmax and the bounded value $Q_t^\pi \in [0, 1]$ give bounded derivatives on every fixed neighbourhood of the origin, so the remainder is uniform over unit directions at a fixed branch. Lemma 3 makes the learned-steering linear term $\rho_{\text{step}} m(\alpha) A_t(G_t; h_t)$ and makes the random-steering linear term zero. Its isotropic second moment gives Eq. (21); subtracting the two expansions gives Eq. (19). □

The proposition is deliberately conditional. It does not say that the drift constructed by Eq. (6) has positive A_t. It says that, at a matched displacement, positive value alignment is exactly the first-order quantity separating learned steering from random steering. The perturbation magnitude multiplies this effect; it does not supply its sign.

F.5 What Information the Entropy Feedback Contains

Let t' be a triggered position and let $\tau_+(t')$ be the next triggered position, with $\tau_+(t') = \infty$ if none occurs before termination. Define the feedback attributable to the perturbation at t' by

$$Z_{t'}(d) = \mathbf{1}[\tau_+(t') < \infty]\big(H_{\tau_+(t')} - \alpha_{\text{target}} H_{\text{first}}\big), \qquad \Phi_{t'}(d) = \mathbb{E}[Z_{t'}(d) \mid h_{t'}]. \tag{22}$$

The expectation includes the tokens generated before the next scored branch. The zero value when none occurs matches Algorithm 1, in which the final perturbation supplies no subsequent update. Write $\bar{s} = \Phi_{t'}(0)$.

Lemma 4 (One-step expected feedback decomposition)**.** *Conditional on the source history and $G \neq 0$, the expected feedback contribution attributed to the perturbation at t' is*

$$\begin{aligned}\mathbb{E}[Z_{t'}\epsilon_{t'}] = \bar{s}\, m(\alpha)\hat{G} \\ + \rho_{\text{step}}\left[\lambda_\perp \Pi_{\hat{G}^\perp} \nabla\Phi_{t'}(0) + \lambda_\parallel \langle \nabla\Phi_{t'}(0), \hat{G}\rangle \hat{G}\right] + O(\rho_{\text{step}}^2),\end{aligned} \tag{23}$$

where $\lambda_\parallel = \mathbb{E}[\langle \epsilon_{t'}, \hat{G}\rangle^2]$ and $\lambda_\perp = (1 - \lambda_\parallel)/(K - 1) > 0$ for $\alpha < 1$.

Proof. Conditional on $\epsilon_{t'}$, expand the expected score:

$$\mathbb{E}[Z_{t'} \mid \epsilon_{t'}] = \bar{s} + \rho_{\text{step}} \langle \nabla\Phi_{t'}(0), \epsilon_{t'}\rangle + O(\rho_{\text{step}}^2).$$

Multiplication by $\epsilon_{t'}$ and expectation give

$$\mathbb{E}[Z_{t'}\epsilon_{t'}] = \bar{s}\, m(\alpha)\hat{G} + \rho_{\text{step}} \mathbb{E}[\epsilon_{t'}\epsilon_{t'}^\top] \nabla\Phi_{t'}(0) + O(\rho_{\text{step}}^2).$$

Rotational symmetry around $\hat{G}$ gives

$$\mathbb{E}[\epsilon_{t'}\epsilon_{t'}^\top] = \lambda_\parallel \hat{G}\hat{G}^\top + \lambda_\perp \Pi_{\hat{G}^\perp},$$

which yields Eq. (23). Multiplication by $(1 - \gamma)$ and addition of the momentum term γG give the update in Eq. (6) whenever the next scored branch occurs. □

The decomposition identifies the information in the update without claiming convergence. The score offset changes the component along the current drift, whereas sensitivity of the next-branch entropy-maintenance feedback to the source perturbation supplies a preconditioned local gradient term. The target $\Phi_{t'}$ changes with the trajectory, the realized update is noisy, and the resulting rank-frame direction is used at a later branch. Consequently, this lemma alone does not imply that G_t aligns with either $\nabla\Phi$ or the reward gradient at the position where it is next applied.

For completeness, the expected feedback has the same softmax-gradient form as branch value. If

$$\bar{Z}(v) = \mathbb{E}[Z_{t'} \mid h_{t'},\ \text{emit } v],$$

then

$$\frac{\partial \Phi_{t'}}{\partial d_k}(0) = p_{t'}(v_k)\big(\bar{Z}(v_k) - \Phi_{t'}(0)\big). \tag{24}$$

Thus the feedback favours tokens whose continuations receive a higher entropy-maintenance score, including the possibility of reaching no subsequent scored branch. Whether those tokens also have higher reward value is a property of the decoder and task, not an identity supplied by softmax.

F.6 From Local Alignment to Rollout Success

The previous proposition evaluates one intervention followed by the base decoder. The actual DRIFT decoder perturbs repeatedly. The following exact telescoping identity connects these objects without assigning a final success to one particular branch point.

Let $\widetilde{\pi}$ denote the complete DRIFT rollout policy. Let $\mathcal{F}_t$ contain the token context h_t, the internal state G_t, and all randomness realized before the perturbation and token draw at t. Thus $\widetilde{\pi}_t(\cdot \mid \mathcal{F}_t)$ is well defined even when the same token prefix can be reached with different internal states. Let

$$J(\pi) = \Pr_{\tau\sim\pi}[\tau \in \mathcal{G}], \qquad J(\widetilde{\pi}) = \Pr_{\tau\sim\widetilde{\pi}}[\tau \in \mathcal{G}].$$

Theorem 2 (Rollout-value decomposition)**.** *For a finite-horizon decoder,*

$$J(\widetilde{\pi}) - J(\pi) = \mathbb{E}_{\tau\sim\widetilde{\pi}}\left[\sum_{t=1}^{T}\left(\mathbb{E}_{v\sim\widetilde{\pi}_t(\cdot|\mathcal{F}_t)}[Q_t^{\pi}(h_t, v)] - V_t(0; h_t)\right)\right]. \tag{25}$$

At non-triggered positions the summand is zero. At a triggered position with $G_t \neq 0$, its conditional expectation over the mixed perturbation is

$$\rho_{\text{step}} m(\alpha) A_t(G_t; h_t) + O(\rho_{\text{step}}^2). \tag{26}$$

Proof. Let $V_t^{\pi}(h_t)$ be the base-policy probability of eventual reward before token t, and let $V_{T+1}^{\pi}(h_{T+1})$ be the terminal reward. For a trajectory sampled from $\widetilde{\pi}$,

$$\begin{aligned} J(\widetilde{\pi}) - J(\pi) &= \mathbb{E}_{\widetilde{\pi}}[V_{T+1}^{\pi}(h_{T+1}) - V_1^{\pi}(h_1)] \\ &= \mathbb{E}_{\widetilde{\pi}}\left[\sum_{t=1}^{T}\left(V_{t+1}^{\pi}(h_{t+1}) - V_t^{\pi}(h_t)\right)\right]. \end{aligned}$$

Conditioning each difference on $\mathcal{F}_t$ and the token sampled by $\widetilde{\pi}_t$ gives Eq. (25). At a non-triggered position the two policies have the same next-token distribution. At a triggered position, the inner difference is $\mathbb{E}_{\epsilon_t}[V_t(d_t; h_t) - V_t(0; h_t) \mid \mathcal{F}_t]$, and Proposition 1 gives Eq. (26). □

Define the aggregate alignment encountered by DRIFT as

$$\mathcal{A}(\widetilde{\pi}, \pi) = \mathbb{E}_{\tau\sim\widetilde{\pi}}\left[\sum_{t: M_t=1,\, G_t\neq 0} A_t(G_t; h_t)\right]. \tag{27}$$

To first order, the rollout-level gain is $\rho_{\text{step}} m(\alpha)\mathcal{A}$. This formulation allows some branch points to hurt and others to help; improvement requires their visitation-weighted sum to dominate the accumulated second-order terms. It also exposes the exact gap left by the entropy estimator: Eq. (23) describes a local learning signal at a source branch, whereas Eq. (27) asks whether the direction carried to later branches predicts their reward value. The theory isolates this temporal alignment but does not guarantee its sign.

F.7 From Rollout Success to Group Quality

The rollout decomposition concerns a single completion. The next result maps a single-rollout success change into the group quantities consumed by group-relative training.

Theorem (Rollout gain and group coverage (restated from Section 3.1, Theorem 1))**.** *Fix a prompt and suppose conditionally independent rollouts succeed with probability p under the base sampler and $p + \Delta$ under DRIFT, where $0 \leq p \leq p + \Delta \leq 1$. If $C_n(p)$ is the probability that a group of n rollouts contains at least one success, then*

$$C_n(p) = 1 - (1 - p)^n \tag{28}$$

and

$$\begin{aligned} C_n(p + \Delta) - C_n(p) &= n\int_p^{p+\Delta}(1 - s)^{n-1}\, ds, \\ n(1 - p - \Delta)^{n-1}\Delta \leq C_n(p + \Delta) - C_n(p) &\leq n(1 - p)^{n-1}\Delta. \end{aligned} \tag{29}$$

Proof. Conditional independence gives a group failure probability $(1-p)^n$, hence Eq. (28). Since $C_n'(s) = n(1-s)^{n-1}$, the fundamental theorem of calculus gives the integral. Its integrand is decreasing, so its endpoint values give the two bounds. □

For small Δ,

$$C_n(p+\Delta) - C_n(p) = n(1-p)^{n-1}\Delta + O(\Delta^2). \tag{30}$$

Thus group sampling can make a small rollout-level improvement visible on hard prompts, while the multiplier vanishes as $p \to 1$. The factor is not always n: it is approximately n only when both the base success probability and the change are small.

The result is prompt-wise. Across a heterogeneous prompt distribution, the coverage change is $\mathbb{E}_x[C_n(p_x+\Delta_x) - C_n(p_x)]$; a positive average change in single-rollout success alone does not imply positive average coverage when Δ_x can have either sign. This is why Table 1a reports solve rate and coverage separately.

Coverage alone does not give the correctness contrast used by an unadjusted binary-reward group-relative update. Let $\mathcal{M}_n(p)$ be the probability that the group contains both a success and a failure. Under the same conditional-independence premise,

$$\begin{aligned} \mathcal{M}_n(p) &= 1 - (1-p)^n - p^n, \\ \mathcal{M}_n(p+\Delta) - \mathcal{M}_n(p) &= n\int_p^{p+\Delta} \left[(1-s)^{n-1} - s^{n-1}\right] ds. \end{aligned} \tag{31}$$

The integrand is positive below $1/2$ and negative above $1/2$. Increasing single-rollout success therefore raises coverage everywhere, but raises mixedness only while the interval from p to $p+\Delta$ lies in the hard-prompt regime below one half. This explains why the two group-level quantities in Table 1a need not change by the same amount.

These group formulas assume independent random draws conditional on the prompt and the sampler state. Sharing a prompt, logits, or a deterministic drift does not itself violate conditional independence; sharing mutable randomness or adaptive state across rollouts would. DRIFT keeps its drift G per rollout, so the rollouts of a group share no adaptive state and the premise holds. If an implementation introduces such dependence, Eqs. (28)–(31) require a joint-distribution analysis instead.

F.8 What the Per-Token Correction Estimates

Eq. (7) reweights each triggered token by its own likelihood ratio and applies no cumulative correction for earlier perturbed tokens. This section states what that objective estimates under the history-dependent behavior of DRIFT, which property it preserves exactly, and how far its gradient can be from the on-policy one. The per-token weight is not claimed to be an unbiased estimator of the on-policy gradient.

The behavior policy. Write $h_t = (x, y_{<t})$. At a triggered position the sampler draws $z_t = \xi_t \sim q = \mathcal{N}(0, I_K)$, independent of the tokens; at other positions z_t is empty. The score s_t of Eq. (6) is computed from the unperturbed p_t, so the drift G_t is a deterministic function of $(h_t, z_{<t})$, and the behavior probability of the next token is

$$\mu_t(\cdot \mid h_t, z_{\le t}) = \begin{cases} \tilde{\pi}_{\mathrm{old},t}(\cdot \mid h_t, z_{\le t}), & M_t = 1, \\ \pi_{\mathrm{old}}(\cdot \mid h_t), & M_t = 0. \end{cases} \tag{32}$$

The behavior depends on the history, but it is Markov on the extended state $(h_t, z_{\le t})$. The joint laws of a trajectory and its noise under DRIFT and under the unperturbed policy are

$$P^{\mu}(\tau, z) = \prod_t q(z_t)\, \mu_t(y_t \mid h_t, z_{\le t}), \qquad P^{\pi}(\tau, z) = \prod_t q(z_t)\, \pi_{\mathrm{old}}(y_t \mid h_t), \tag{33}$$

and the τ-marginal of P^{π} is the on-policy law. For each t, let $P^{(t)}$ be the hybrid law that uses μ_s at every $s \neq t$ and π_{old} at t. Algorithm 1 records $\tilde{\pi}_{\mathrm{old},t}(\tilde{y}_t)$ at sampling time, so $w_t = \pi_{\mathrm{old}}(y_t \mid h_t)/\mu_t(y_t \mid h_t, z_{\le t})$ uses the exact behavior probability conditioned on everything that produced the token.

Lemma 5 (Per-token exactness). *For every t and every integrable $\varphi(\tau, z)$, $\mathbb{E}_{P^{\mu}}[w_t\, \varphi] = \mathbb{E}_{P^{(t)}}[\varphi]$.*

Proof. P^μ and $P^{(t)}$ share every factor in Eq. (33) except the t-th token factor, and w_t is the ratio of the $P^{(t)}$ factor to the P^μ factor. A softmax with finite logits is positive on every token, so the ratio is finite and the change of measure is valid. When $M_t = 0$ the two factors coincide and $w_t = 1$. □

Corollary 1 (Exact baseline cancellation)**.** *Let b be any baseline that, given $(h_t, z_{\le t})$, does not depend on y_t. Then $\mathbb{E}_{P^\mu}\big[w_t \nabla_\theta \log \pi_\theta(y_t \mid h_t)\, b\big] = 0$ at $\theta = \theta_{\text{old}}$.*

Proof. By Lemma 5, the expectation equals $\mathbb{E}_{P^{(t)}}\big[b \sum_y \pi_{\text{old}}(y \mid h_t) \nabla_\theta \log \pi_\theta(y \mid h_t)\big] = \mathbb{E}_{P^{(t)}}\big[b \nabla_\theta \sum_y \pi_\theta(y \mid h_t)\big] = 0$. □

Without w_t the inner sum is taken under μ_t and is generally nonzero, so the update would move the policy toward the tokens the drift favored even when the rewards carry no signal. Corollary 1 is the formal content of the statement that the correction keeps the update from reading the drift's preference as the policy's own. The leave-one-out group baseline satisfies its premise exactly; the group mean that includes the rollout's own reward adds the usual $O(1/G)$ term, which does not depend on DRIFT.

What Eq. (8) estimates. At $\theta = \theta_{\text{old}}$ we have $\rho_t = 1$ and the clip is inactive, so, writing $\psi_t = \nabla_\theta \log \pi_\theta(y_t \mid h_t)$, R for the reward, and treating the positive group-wise standard-deviation scale as a constant, the gradient of Eq. (8) is a sample estimate of

$$\hat{g} = \sum_t \mathbb{E}_{P^\mu}\big[w_t\, \psi_t\, (R - b)\big] = \sum_t \mathbb{E}_{P^{(t)}}\big[\psi_t\, R\big], \tag{34}$$

where the equality uses Lemma 5 and Corollary 1. The on-policy gradient is $g^\star = \sum_t \mathbb{E}_{P^\pi}[\psi_t\, R]$. Each term of Eq. (34) is therefore an unbiased, baseline-invariant estimate of the policy-gradient contribution of token t when token t is drawn from π_{old} and the other positions keep the behavior law. This is a local policy-improvement objective in the sense of Kakade & Langford (2002), taken relative to the behavior distribution, and the clipped form of Eq. (8) is the standard trust-region approximation around it (Schulman et al., 2017).

Proposition 2 (Bias from the omitted cumulative correction)**.** *Assume $\|\psi_t\| \le B$ and $R \in [0, 1]$, and let $\delta_s = \mathrm{TV}\big(\pi_{\text{old}}(\cdot \mid h_s), \tilde{\pi}_{\text{old},s}(\cdot \mid h_s, z_{\le s})\big)$. Then*

$$\|\hat{g} - g^\star\| \le 2B \sum_t \mathbb{E}_{P^{(t)}}\Big[\sum_{s \ne t} M_s\, \delta_s\Big], \qquad \delta_s \le \frac{\sqrt{2}}{4}\, \sigma \sqrt{K}. \tag{35}$$

Proof. The function $\varphi_t = \psi_t R$ satisfies $\|\varphi_t\| \le B$, so $\|\mathbb{E}_{P^{(t)}}[\varphi_t] - \mathbb{E}_{P^\pi}[\varphi_t]\| \le 2B\, \mathrm{TV}(P^{(t)}, P^\pi)$. The two laws are built from the same noise factors and the same token kernels except at triggered positions $s \ne t$, where $P^{(t)}$ uses $\tilde{\pi}_{\text{old},s}$ and P^π uses π_{old}. The maximal coupling of the two kernels at each step, run sequentially, gives $\mathrm{TV}(P^{(t)}, P^\pi) \le \mathbb{E}_{P^{(t)}}[\sum_{s \ne t} M_s \delta_s]$. Summing over t gives the first inequality.

For the second, let $d = P_K^\top \sigma \sqrt{K} \epsilon_s$ be the logit displacement, zero outside $\mathcal{K}_s$. Then $\mathrm{KL}(\pi_{\text{old}} \,\|\, \tilde{\pi}_{\text{old},s}) = \log \mathbb{E}_{\pi_{\text{old}}}[e^{d_v}] - \mathbb{E}_{\pi_{\text{old}}}[d_v] \le (\max_v d_v - \min_v d_v)^2/8$ by Hoeffding's lemma. The coordinates of d include 0, and for a unit vector ϵ_s, $\max(\max_k \epsilon_{s,k}, 0) - \min(\min_k \epsilon_{s,k}, 0) \le \sqrt{2}$, so the range is at most $\sqrt{2}\, \sigma \sqrt{K}$ and the KL divergence is at most $\sigma^2 K/4$. Pinsker's inequality, $\mathrm{TV} \le \sqrt{\mathrm{KL}/2}$, gives $\delta_s \le \sigma \sqrt{K}/(2\sqrt{2})$. □

The bias is linear in the total variation that the other perturbed positions introduce. It vanishes as $\sigma \to 0$, receives no contribution from untriggered positions, and has the same form as the prefix-distribution shift between π_θ and π_{old} that PPO already ignores (Schulman et al., 2017). The worst-case constant in Eq. (35) is conservative; the realized δ_s depends on the drift direction and on how much of p_s lies in $\mathcal{K}_s$.

Why not the cumulative correction. The trajectory weight $W = \prod_t w_t$ removes the bias: $\mathbb{E}_{P^\mu}[W\varphi] = \mathbb{E}_{P^\pi}[\varphi]$ by the same argument as Lemma 5 applied to every factor. With an outcome reward, per-decision truncation (Precup et al., 2000) cannot drop the factors after t, because R depends on every token, so the exact estimator needs the full product. Its second moment grows geometrically in the number of branch points.

Proposition 3 (Variance of the cumulative weight)**.** *If every rollout has at least m triggered positions and $\chi^2\big(\pi_{\mathrm{old}}(\cdot \mid h_s) \,\|\, \tilde{\pi}_{\mathrm{old},s}(\cdot \mid h_s, z_{\le s})\big) \ge c$ at each of them, then $\mathbb{E}_{P^\mu}[W^2] \ge (1+c)^m$.*

Proof. $\mathbb{E}_{P^\mu}[W^2] = \mathbb{E}_{P^\pi}[W]$ by change of measure. Conditioning on $(h_s, z_{\le s})$ at the last position s, $\mathbb{E}_{P^\pi}[w_s \mid h_s, z_{\le s}] = \sum_y \pi_{\mathrm{old}}(y \mid h_s)^2 / \mu_s(y \mid h_s, z_{\le s}) = 1 + \chi^2(\pi_{\mathrm{old}} \,\|\, \mu_s)$, which is at least $1 + c$ if s is triggered and equals 1 otherwise. All weights are nonnegative, so peeling the positions off one at a time from the last gives the bound. □

The per-token weight therefore trades a bias that is additive in $\sum_s M_s \delta_s$ for the removal of a variance that is multiplicative in the number of branch points. Every token in a group enters the update, and a product of m weights would concentrate the gradient on the few rollouts whose perturbations happened to agree with π_{old}.

F.9 SCOPE

Four conclusions follow. The trigger excludes regimes with little available first-order effect. At matched step size, value alignment determines the first-order gap between learned and random steering. The entropy update contains a local gradient signal for its next-branch feedback. Any resulting single-rollout gain produces the exact group-level changes above.

The local expansion can be overturned by curvature at a finite step. More centrally, Lemma 4 does not prove positive aggregate alignment $\mathcal{A}$: entropy and reward can disagree, and a rank-frame direction need not transfer across branches. Nor does the theory establish distinct correct solution strategies; correct-only diversity remains empirical.