PROCESS COHERENCE GAMES FOR CODE ALIGNMENT: BOOTSTRAPPING FROM ZERO

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) excel at code generation and complex multi-step reasoning, yet aligning them toward functional correctness remains a severe challenge. Standard alignment paradigms, such as Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO), rely on syntactic matching against canonical reference traces, thereby penalizing valid but syntactically diverse reasoning paths. Conversely, outcome-based Reinforcement Learning (RL) suffers from extreme gradient variance and produces zero gradient signal in complex domains where the probability of an initial correct solution is near zero (pass@k = 0). In this paper, we overcome these bottlenecks by formulating code generation alignment via *Pairwise Coherence*. We propose C2-GRPO and its probabilistic extension PC-GRPO, which bypass the need for exhaustive ground-truth oracles by optimizing for functional semantic consistency over dynamically generated unit tests. To solve the pass@k = 0 cold-start problem, we introduce two algorithmic techniques, *Soft Plausibility Weighting* and *Process Coherence*, which dynamically synthesize an implicit Process Reward Model (PRM) entirely through unsupervised group consensus, reducing the group size needed for a consensus signal on long-horizon tasks from exponential to linear under step-wise decomposition (via tree-based simulation or, for code generation, test-decomposed independent rollouts). We prove that standard GRPO's gradient signal vanishes as pass@$k \to 0$ while PC-GRPO maintains $\Omega(1/K)$ advantage variance whenever the group's functional disagreement is non-degenerate (an explicit count-separation condition), and establish convergence and Rademacher generalization bounds for an idealized form of the coherence objective. Empirically, on Qwen2.5-Coder-7B-Instruct, PC-GRPO is the strongest trainable method across APPS Intro, APPS Interview, and LiveCodeBench release_v1. Even at 7B scale, APPS Interview retains a substantial cold-start regime; in these groups, PC-GRPO produces usable gradient variance in 91.0% of cases where binary rewards provide none. We further prove fundamental impossibility results for ungrounded consensus (the *Degenerate Consensus* collapse) and its partial converse (*grounded one-step improvement*), and integrate bounded Minimax Polynomial estimators (Mohri et al., 2026b) for finite-sample stability, and structurally bound reward hacking via the consensus formulation.

1 INTRODUCTION

Large Language Models (LLMs) achieve remarkable success in code generation (Rozière et al., 2023; Guo et al., 2024) and agentic environments (Zhou et al., 2024b; Jimenez et al., 2024), yet aligning them for functional correctness remains a fundamental challenge.

The Problem. The core difficulty is structural: programming admits vast spaces of *semantically equivalent* but *syntactically diverse* solutions. Standard paradigms, SFT (Ouyang et al., 2022) and DPO (Rafailov et al., 2023), penalize valid reasoning paths that deviate from a single canonical reference trace, discarding the rich diversity of correct code. Execution-based RL overcomes this syntactic rigidity but faces a more insidious barrier: the *cold-start problem*. When training a 1.5B-parameter model on APPS Interview-level problems, for a large fraction of problems the probability that the initial policy generates a fully correct solution is effectively zero (pass@k = 0). Standard GRPO (Shao et al., 2024) assigns binary rewards, and since *every* candidate fails, every advantage is identically zero: the model receives no gradient signal and training ceases to progress. Process

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.

Reward Models (PRMs) (Lightman et al., 2024) could provide dense step-level feedback, but require costly step-level supervision or a separately trained reward model (Wang et al., 2024).

The Idea of the Solution. Our key insight is that even when no candidate is fully correct, the *pattern of disagreements* among candidates on dynamically generated test inputs carries rich information about partial correctness. We formalize this via *Pairwise Coherence* (Mohri et al., 2026a): instead of requiring ground-truth labels, the reward for each trace is determined by how well its outputs agree with the *group consensus distribution*. This yields a continuous reward signal that persists even when pass@k = 0, because candidates that get early reasoning steps right will agree with each other more often than those that fail immediately.

Theoretical Justification. We establish that this dynamic reward landscape is rigorous and robust: the consensus formulation structurally caps the reward at zero, bounding reward over-optimization (Theorem 2); a constrained Minimax estimator provides finite-sample stability with a bounded, tuning-free reward (Theorem 15); soft plausibility weighting recovers the game equilibrium under maximum-entropy cold-start; and process coherence factorizes the consensus, reducing the group size needed for a consensus signal on long-horizon tasks from exponential to linear.

Concretely, we introduce PROBABILISTIC COHERENCE GRPO (PC-GRPO), which instantiates this game for code generation. Our contributions are:

1. **Solving the pass@k = 0 Cold-Start:** We introduce two algorithmic techniques, *Soft Plausibility Weighting* and *Process Coherence*, which dynamically synthesize an implicit PRM entirely from unsupervised group consensus (Proposition 14), reducing the group size needed for a consensus signal on long-horizon tasks from exponential to linear for both tree-based and independent-rollout settings (Theorem 1).
2. **Theoretical Guarantees:** We prove that standard GRPO's gradient signal vanishes as pass@$k \to 0$, while PC-GRPO maintains $\Omega(1/K)$ advantage variance under an explicit count-separation condition on the consensus structure (Theorem 3). We establish convergence, generalization, and a provable cap on reward over-optimization under the idealized assumptions of Appendix A (Theorems 2, 16 and 18), and formalize the *Degenerate Consensus* impossibility with a partial converse, *grounded one-step improvement* (Theorems 4 and 6).
3. **Empirical Validation:** As shown in Section 5, on Qwen2.5-Coder-7B-Instruct, PC-GRPO is the strongest trainable method across APPS Intro, APPS Interview, and LiveCodeBench release_v1, outperforming Vanilla GRPO, Partial-Credit GRPO, CodeRM-style GRPO, self-play SFT, and R1-style GRPO. We further show that even at 7B scale, APPS Interview retains a substantial cold-start regime: 33.3% of sampled groups contain no correct candidate, yet PC-GRPO produces usable gradient variance in 91.0% of these cold-start groups.

Related Work. *Augmented SFT* (Rozière et al., 2023; Wang et al., 2023) captures diverse syntax but is statically capped. *RLHF/RLAIF* with learned reward models (Ouyang et al., 2022; Rafailov et al., 2023) scales better but is susceptible to reward hacking (Skalse et al., 2022); our consensus reward is capped at zero, which bounds over-optimization (Theorem 2). *PRMs* (Lightman et al., 2024) provide dense step-wise supervision but require expensive annotation, and *ORMs* (Outcome Reward Models) scale but suffer sparse gradients; our framework bridges this gap by synthesizing implicit PRMs from unsupervised consensus. *Tree-Search* (AlphaCode (Li et al., 2022), rStar (Qi et al., 2024)) and test-time scaling (Snell et al., 2025) are complementary *inference-time* strategies. *Self-Play* (SPIN (Chen et al., 2024c), Self-Rewarding LMs (Yuan et al., 2024)) and consensus-based inference (Chen et al., 2024a) lack our convergence guarantees; *majority-vote self-rewarding RL* (TTRL (Zuo et al., 2025), SRT (Shafayat et al., 2025)) is the closest precedent, and the collapse SRT reports under prolonged self-reward is the degenerate dynamic that Theorem 4 formalizes and that grounding counteracts (Theorem 6). *Iterative Refinement* (Le et al., 2022; Shinn et al., 2023) optimizes post-hoc debugging rather than initially correct generations. See Appendix B for an extended review.

2 PROBLEM FORMULATION

Standard alignment methods (SFT, DPO) optimize the likelihood of a specific trace y, yet in code generation the primary objective is the correctness of the final answer z, not the phrasing of the intermediate reasoning; this *Answer-Level Fine-Tuning* (ALFT) paradigm requires optimizing the marginal probability of z.

ALFT objective and its intractability. Given a prompt x, a policy $\pi(\cdot|x)$ generates traces $y \in \mathcal{Y}$, each producing an outcome $z = \mathsf{E}(y) \in \mathcal{Z}$, where E is a mapping from traces to their corresponding response. The policy's *answer marginal* is $\nu_\pi(z|x) = \sum_{y:\mathsf{E}(y)=z} \pi(y|x)$. Direct optimization of this marginal is intractable because ν_π depends on the non-differentiable extraction map E, and the preimage $\{y: \mathsf{E}(y) = z\}$ is vast and impossible to enumerate.

Alignment via Pairwise Coherence. To overcome this without requiring exhaustive ground-truth oracles, we rely on the Answer-Level Fine-Tuning (ALFT) framework (Mohri et al., 2026a). Its Coherence Game (Mohri et al., 2026a, Sections 5–6) addresses settings where the correct answer is not known: instead of optimizing against a static label, it introduces a Target distribution $\mathsf{q} \in \Delta(\mathcal{Z})$ representing the group's consensus.

At equilibrium, the optimal target q^* is the Bregman centroid of the empirical output distributions. For tractability, this is relaxed to the arithmetic mean of the group's outputs. The continuous reward that aligns the policy with a target is its log-likelihood, $R(y) = \beta \log \mathsf{q}^*(\mathsf{E}(y))$ (Mohri et al., 2026a, Section 3), which we apply with the consensus target; its KL-regularized optimum is the Bayesian update $\pi \propto \pi_0\, \mathsf{q}^*$ (Theorem 12). (The PAIRWISE-GRPO algorithm of their Section 6 instead scores open-ended outputs by average pairwise agreement.) This per-trace reward is the signal used by GRPO (Shao et al., 2024) to form group-normalized advantages without ground-truth labels (see Appendix C).

Extension to code generation. In standard ALFT, the answer z is known from training data. Code generation is harder: z is *neither known in advance nor unique*; a problem may admit many valid output behaviors. We resolve this by replacing the fixed answer with a *dynamic behavioral test*: each candidate y is evaluated on dynamically generated unit tests $U = \{u_1, \ldots, u_N\}$, and correctness is defined through the *pattern of agreement* among candidates. Pairwise disagreement is:

$$\mathsf{d}(y, y') = 1 - \frac{1}{N} \sum_{j=1}^{N} \mathbb{I}[y(u_j) = y'(u_j)]. \tag{1}$$

Illustrative example. Four candidates are evaluated on a unit test: three output `42`, one outputs `17`. The consensus gives $\mathsf{q}^*(\texttt{42}) = 3/4$, $\mathsf{q}^*(\texttt{17}) = 1/4$, so the outlier's reward $\beta \log(1/4)$ is a $4.8\times$ stronger penalty than the majority's $\beta \log(3/4)$, with no ground-truth label needed (a fully worked training example is given in Appendix H).

To increase behavioral diversity, we also evaluate each prompt alongside a *semantics-preserving paraphrase* $\Phi(x)$ that specifies the same function but induces a distributional shift ($\mathsf{D}_{\mathrm{KL}}(\pi(\cdot|x) \| \pi(\cdot|\Phi(x))) > 0$). Relative to a *symmetrized* single-prompt baseline (a homogeneous batch drawn from either prompt), orbit mixing provably increases the expected number of distinct explored traces by an additive $\Omega(K^2 \|p - q\|_2^2)$ (under the combinatorial decoding regime $p(y), q(y) \le c_1/K$), where p and q are the trace distributions induced by the two prompts (proof in Appendix D). In practice, $\Phi(x)$ is generated by prompting the base LLM to paraphrase the problem statement while preserving input/output semantics; the paraphrase is cached per training example and reused across epochs.

3 ALGORITHMIC FRAMEWORK

The game-theoretic formulation of Section 2 provides the right objective, but computing the exact Bregman centroid q^* over all possible outputs is intractable. The key idea is to *estimate* q^* empirically from a finite group of K candidate programs, translating the game into a scalable family of RL algorithms built up in three stages (full pseudocode in Appendices E.1 and E.2).

3.1 FROM HARD CONSENSUS TO PROBABILISTIC TARGETS

C2-GRPO (Hard Consensus). The simplest instantiation uses majority vote as a proxy for q^*: a small set of ground-truth *seed tests* filters out incorrect solutions, establishing a *plausible set* $G' \subseteq G$; for dynamically generated unit tests U, the hard consensus output is $v_j^* = \mathrm{Mode}(\{y'(u_j) \mid y' \in G'\})$ (here $y(u_j)$ denotes the output of trace y, viewed as an executable program, on test input u_j), and plausible candidates are rewarded via binary agreement with this mode: $R(y) = \frac{1}{|U|} \sum_j \mathbb{I}[y(u_j) = v_j^*]$.

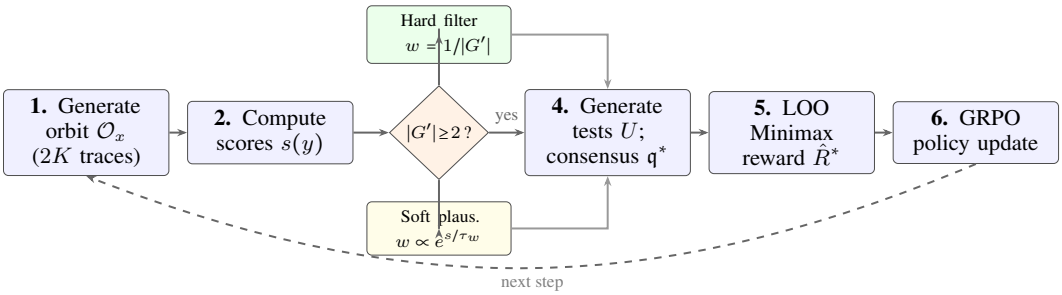


Figure 1: Pipeline overview of PC-GRPO (Algorithm 1). Step 3 selects between hard plausibility filtering ($|G'| \geq 2$) and soft weighting (pass@$k = 0$). The Minimax reward (Step 5) is a bounded polynomial table lookup replacing $\log(X/K)$.

Algorithm 1 PC-GRPO with Adaptive Soft Plausibility (Handles pass@$k = 0$)

Input: Dataset $\mathcal{D}$, Policy π, Group size K, Transformation Φ, Seed test suites T_{seed}, Generator $\mathcal{G}_{\text{test}}$, Temperature τ_w, Threshold θ, KL coefficient β, LOO denominator floor ϵ_w.

for each training step **do**
  Sample batch $B \sim \mathcal{D}$.
  for each seed $x \in B$ **do**
    1. Generate orbit $\mathcal{O}_x = \{x, \Phi(x)\}$; sample K candidates per element → group G, $|G| = 2K$.
    2. Compute progress scores $s(y) \in [0, 1]$ $\forall y \in G$ (fraction of the seed tests $T_{\text{seed}}(x)$ passed).
    3. Adaptive Weighting: $G' = \{y : s(y) \geq \theta\}$. If $|G'| \geq 2$: $w(y) = 1/|G'|$ for $y \in G'$ (and 0 otherwise); else $w(y) = \text{softmax}(s(y)/\tau_w)$ over G.
    4. Generate unit tests U via $\mathcal{G}_{\text{test}}$; compute consensus $\mathsf{q}^*_{u_j}(v) = \sum_y w(y)\, \mathbb{I}[y(u_j) = v]$.
    5. Reward: $\forall y \in G$: let $K_{\text{act}} = |G'|$ if $|G'| \geq 2$, else $|G|$. For each test $u_j \in U$, compute the LOO peer mass $p^{(j)}_{\text{LOO}} = \max(0, \mathsf{q}^*_{u_j}(y(u_j)) - w(y)) / \max(1 - w(y), \epsilon_w)$ and map it to the pseudo-count $X_j = \lfloor (K_{\text{act}} - 1) p^{(j)}_{\text{LOO}} \rfloor$. Set $R(y) = \frac{1}{|U|} \sum_j \beta \cdot \widetilde{c}^*_{X_j, K_{\text{act}}-1}$. {$\widetilde{c}^*$: Minimax poly. coeff. (Mohri et al., 2026b)}
    6. Compute advantages over the active set ($\{y : w(y) > 0\}$; all of G in the soft-weight fallback); assign $A(y) = \min_{y' \in G'} A(y') - 1.0$ for $y \notin G'$ when $|G'| \geq 2$. Update π via GRPO.
  end for
end for

PC-GRPO (Probabilistic Coherence). Hard consensus fails when valid solutions are multimodal: two programs can both be correct but produce different outputs on certain inputs. To handle this, we apply the full Pairwise Coherence theory. The optimal target distribution q^* reduces to the Bregman Centroid of the output distributions. For a unit test u_j, the soft target is the empirical frequency among the plausible programs:

$$\mathsf{q}^*_{u_j}(v) = \frac{1}{|G'|} \sum_{y' \in G'} \mathbb{I}[y'(u_j) = v]. \tag{2}$$

The optimal continuous reward for a trace y'_i is its expected log-likelihood under this target distribution:

$$R(y'_i) = \frac{\beta}{N} \sum_{j=1}^{N} \log \left[\frac{\text{Count}(\text{output } y'_i(u_j) \text{ for input } u_j)}{|G'|} \right]. \tag{3}$$

This is the core equation of our framework: it rewards each trace proportionally to its agreement with the group's consensus *distribution*, providing a continuous gradient signal even when no single trace is fully correct. To see why this matters, note that with binary pass/fail reward $R_{\text{bin}}(y) \in \{0, 1\}$, the GRPO group-normalized advantage for trace y in a group G is:

$$\hat{A}_{\text{GRPO}}(y) = R_{\text{bin}}(y) - \frac{1}{|G|} \sum_{y' \in G} R_{\text{bin}}(y'). \tag{4}$$

When pass@$k = 0$, every $R_{\text{bin}}(y) = 0$, so $\hat{A}_{\text{GRPO}}(y) = 0$ for all y, no gradient signal at all. In contrast, our coherence reward (3) varies across the group whenever candidates differ in how often their outputs are shared, which fails only for exactly balanced splits (Appendix F.2).

Two practical refinements complete the reward. First, including a trace's own output in the consensus count creates a systematic bias, so we use *leave-one-out* (LOO) counts: $X_j^{\text{LOO}}(y_i') = \sum_{y' \in G' \setminus \{y_i'\}} \mathbb{I}[y'(u_j) = y_i'(u_j)]$. Second, when a program produces a unique output ($X_j^{\text{LOO}} = 0$), the naïve reward $\log(0/K) = -\infty$ dominates the gradient update; the Minimax Polynomial estimator replaces $\log(X/K)$ with a bounded monotone lookup $\hat{R}^*(X, K)$ whose coefficients minimize the worst-case p-weighted bias within that class, implemented as a simple $\mathcal{O}(1)$ array lookup (detailed in Section 4 and Appendix F.3).

Unit test generation. The consensus reward requires functional unit tests $U = \{u_1, \ldots, u_N\}$ that discriminate between candidate programs. A test generator $\mathcal{G}_{\text{test}}$ produces input cases targeting behavioral diversity among the group; we formalize $\mathcal{G}_{\text{test}}$ as an adversarial player maximizing Gini impurity across the group's outputs (see Section 4.3 and Appendix G for the formal game and complexity bounds).

3.2 Bootstrapping from Zero: Adaptive Soft Plausibility

Both variants above assume at least some candidates pass the seed tests. But in highly complex agentic domains (e.g., SWE-bench, WebArena), the probability of generating a flawless trace from the initial policy is often near zero (pass@$k = 0$). When the plausible set G' is empty, no discriminative signal is produced; this is the cold-start problem we solve. We introduce our primary algorithm, **Adaptive PC-GRPO** (Algorithm 1), which bootstraps consensus from partial successes. There is no circularity here: pass@$k = 0$ (no candidate fully solves the problem) and $p_{\text{seed}} > 0$ (some candidate passes the basic seed tests) are different conditions, so a cold-start group can still contain seed-passing candidates (at 7B, 33.3% of groups have pass@$K = 0$; Table 6). If even seed tests fail, Step 3 of Algorithm 1 falls back to softmax weighting over the entire group, with no binary filtering at all.

Soft Plausibility Weighting. Instead of a binary filter, we introduce a continuous progress score $s(y) \in \mathbb{R}$ for each trace (e.g., fraction of test assertions passed). We transform these scores into a plausibility distribution over the entire generated group G using a softmax operator with temperature $\tau_w > 0$: $w(y) \propto \exp(s(y)/\tau_w)$. The Target player then constructs a *weighted* soft consensus distribution:

$$\mathsf{q}^*_{u_j}(v) = \sum_{y \in G} w(y)\, \mathbb{I}[y(u_j) = v]. \tag{5}$$

When $|G'| \geq 2$, the weight assignment sets $w(y) = 1/|G'|$ for $y \in G'$ and $w(y) = 0$ for $y \notin G'$.

Process Coherence as Implicit PRMs. In agentic settings, an "answer" y is a sequential trajectory $\tau = (s_0, a_0, \ldots, s_T, a_T)$, where s_t denotes the environment state (e.g., the interpreter's memory after executing the first t code blocks) and a_t denotes the action taken at step t (e.g., the next code block generated by the policy). When pass@$k = 0$, no full trajectory is correct, but early steps might be valid. The idea is to evaluate consensus step-by-step: we define a step-wise target distribution $\mathsf{q}^*_t(a \mid s_t)$ as the weighted empirical frequency of actions taken at step t among the group's trajectories, and apply the Minimax polynomial estimator to the leave-one-out count $X_t = \sum_{j \neq i} \mathbb{I}[a_t^{(j)} = a_t \mid s_t^{(j)} = s_t]$ at each step. The total reward becomes:

$$R(\tau) = \sum_{t=0}^{T} \gamma^t \beta\, \widetilde{c}^*_{X_t,\, K_t^{\text{LOO}}}, \tag{6}$$

where K_t^{LOO} is the leave-one-out group size at step t and $\widetilde{c}^*_{X,K}$ is the Minimax polynomial coefficient from Theorem 15. This provides the theoretical basis for *implicit Process Reward Models (PRMs)*: the step-level coherence reward assigns dense credit without requiring any human annotation. The Minimax form in (6) applies the outcome-level lookup (Step 5 of Algorithm 1) at each step: when the consensus target factorizes into per-step conditionals, $\mathsf{q}^*(\tau) = \prod_t \mathsf{q}^*_t(a_t \mid s_t)$, the exact log-reward is the sum of the per-step logs (Proposition 14), which (6) approximates term by term (Theorem 15); the discount γ^t (which down-weights later steps, sharpening focus on early-trajectory quality) is a practical modification, not an equilibrium quantity (see Appendix E.3). Crucially, the Minimax estimator eliminates the need for ad-hoc numerical stabilizers: when no other trajectory shares the

action a_t (i.e., $X_t = 0$), the reward is the structurally bounded Minimax floor $\beta c_0^\star = \Theta(-\beta \log K)$ rather than a tuned $\log(\epsilon)$.

Environmental Invariance. In agentic settings, the Unit Test Generator produces *Environment Perturbations* rather than standard unit tests: agents causing identical side-effects across perturbations share a coherent semantic intent, forcing the policy to learn causal invariants rather than overfitting to syntax. While Environmental Invariance provides the conceptual foundation for extending consensus to agentic settings, its full operationalization with execution feedback remains future work; the current SWE-bench evaluation uses syntactic command matching as a proxy (Appendix A).

4 Theoretical Analysis and Fundamental Limits

We provide formal guarantees organized around three questions: (1) *Can partial-correctness consensus bootstrap a viable policy?* (2) *Does the resulting reward landscape resist reward hacking?* (3) *Are there fundamental limits to consensus-based alignment?* Our framework builds on the log-likelihood reward and Coherence-Game target of Mohri et al. (2026a) and the Minimax Polynomial estimator (Mohri et al., 2026b); all other results below are original contributions. Proofs are in Appendices F.1 to F.3.

4.1 Bootstrapping and Process Coherence

The softmax plausibility weighting in Algorithm 1 is not an ad-hoc heuristic: it approximates the optimal Bayesian posterior update in the heuristic-guided Pairwise Coherence game, reweighting each trace by its progress score and its consensus agreement (Theorem 12, proved in Appendix F.1). Moreover, even when no candidate solves the full problem (pass@$k = 0$), the soft consensus target systematically upweights correct sub-trajectories: under a mild stochastic-dominance assumption on the progress score, the consensus probability of the correct prefix strictly exceeds the policy's prior (Theorem 13; Appendix F.1); the policy's mass on the prefix then rises in rounds in which the prefix is valid-favoring (Theorem 6).

To scale this signal to long-horizon tasks, we exploit the prefix-tree structure of deterministic rollouts whose state records the action history: the logarithmic coherence reward then factorizes exactly into additive step-wise process rewards (Proposition 14; Appendix F.1), yielding an implicit PRM without per-step annotation. The sample complexity improvement is dramatic:

Theorem 1 (Exponential Separation of Sample Complexity)**.** *Consider an agentic task of length T with per-step success probability η, and ask how large the group must be for two matching valid samples (a valid consensus) to appear with probability at least a constant $c > 0$. (1)* Outcome Coherence: *For two fully valid trajectories, the group size must scale exponentially with task length: $K = \Omega\big(\eta^{-T}\big)$. (2)* Process Coherence (tree-based): *Under a branching simulator that evaluates independent actions from shared intermediate states, the group size scales strictly linearly: $K = \Omega\big(\eta^{-1}\big)$. (3)* Process Coherence (test-decomposed): *If the consensus decomposes over N independently evaluable criteria (e.g., unit tests), each producing a correct output with probability p_{test}, the group size scales as $K = \Omega\big(p_{\text{test}}^{-1}\big)$, linearly in the per-criterion difficulty.*

Full valid-trajectory consensus at pass@$k = 0$ is thus out of reach for practical group sizes ($K \le 64$), while Part (3) applies directly to code generation, where each program is evaluated independently on each unit test (proof and discussion in Appendix F.3).

4.2 Stability, Convergence, and Gradient Signal

Reward bounding. A key concern with any learned reward is reward hacking: can the policy inflate the reward without genuine improvement?

Theorem 2 (Intrinsic Bounding and Entropy Limits)**.** *For any target distribution $\mathsf{q}^* \in \Delta(\mathcal{Z})$ generated during the Pairwise Coherence game, the reward for any trace y is strictly bounded above: $R(y) \le 0$. Furthermore, if the policy π perfectly matches the target consensus, the expected reward evaluates exactly to the negative entropy of the consensus: $\mathbb{E}_{y\sim\pi^*}[R(y)] = -\beta H(\mathsf{q}^*)$.*

Unlike static reward models, the consensus reward is capped at 0 and rises only by agreeing with more of the group (proof in Appendix F.2).

Finite-sample stability and convergence. In practice, q^* is estimated from a finite group of K samples. The raw empirical logarithm is unbounded at unobserved outcomes, and smoothing it requires a floor tuned to K; we instead use the Minimax Polynomial estimator of Mohri et al. (2026b) with its coefficients constrained to be monotone and non-positive, which keeps the reward deterministically bounded in $[\beta c_0^*, 0]$ with $c_0^* = -\Theta(\log K)$ and minimizes the worst-case p-weighted bias within that class, at most $\mathcal{O}(1/K)$ (Theorem 15; Appendix F.2). Because the estimator is exactly unbiased for its polynomial target R^*, PC-GRPO analyzed as Stochastic Mirror Descent over mode-level policies (reweighting π_0 across outcomes) with a bounded unbiased gradient oracle converges at rate $\mathcal{O}(\beta \log K \sqrt{D_{\max}/T})$ to the polynomial-reward game optimum $\mathcal{L}^*$ (Theorem 16; Appendix F.2); the gap between $\mathcal{L}^*$ and the ideal log-reward optimum $\mathcal{L}^*_{\log}$, $\mathcal{O}(\beta/(q_{\min} K))$ for a consensus target bounded below by $q_{\min}$, is a static constant that does not enter the optimization trajectory. Standard Rademacher bounds give uniform convergence over unseen prompts with a consensus-specific factor of only $\mathcal{O}(\beta \log K)$ (Theorem 18; Appendix F.2).

Cold-start gradient characterization. With bounded, stable rewards and convergence in hand, what *kind* of gradient signal does PC-GRPO receive in the cold-start regime?

Theorem 3 (Advantage Variance Separation)**.** *Let $\hat{A}_{GRPO}(y)$ and $\hat{A}_{PC}(y)$ denote the advantage estimators of Vanilla GRPO (binary pass/fail) and* PC-GRPO *(Minimax coherence reward), respectively, evaluated on a group of size K. Let η = pass@1 be the per-trace success probability and* Var *the empirical sample variance (denominator K) of the mean-centered, unstandardized advantages within the group. Then: (1) For K traces i.i.d. from π on one prompt, $\mathbb{E}[\mathrm{Var}[\hat{A}_{GRPO}]] = \frac{K-1}{K}\eta(1-\eta)$. As $\eta \to 0$ the variance vanishes, and with the standard-deviation normalization used in practice the cold-start advantage is the indeterminate $0/0$ (masked to 0): zero gradient signal. (2)* (Count separation.) *Suppose a fraction at least $\alpha > 0$ of the filtered group forms a dominant mode (identical outputs on every evaluated test, each a plurality on its test) and some program's output on one test has a leave-one-out count at most $1/c$ times the dominant output's, for constants $\alpha > 0$, $c > 1$. Then for $K \geq K_0(c, \alpha)$, with the number of tests N and β fixed, the test-averaged Minimax reward gap between the dominant mode and that program is at least $(\beta/2N)\log c = \Omega(1)$ and $\mathrm{Var}[\hat{A}_{PC}] = \Omega(1/K)$, even when $\eta = 0$ (*Var*: sample variance within one such group).*

As $\eta \to 0$, binary rewards collapse all advantages to zero, while Part (2) lower-bounds the signal under an explicit count-separation condition (at the training group size of 16 candidates and $c = 2$, the bound holds once the dominant output has leave-one-out count at least 5; Table 2); empirically, zero-variance configurations occur in only 9.0% of 7B cold-start groups (Table 6; proof and boundary cases in Appendix F.2).

4.3 Fundamental Limits and Necessity of Grounding

Can consensus fail *without* grounding? We prove that it can, that grounding suffices to reverse the collapse, and that without grounding the collapse is irreversible.

Theorem 4 (Multi-Step Degenerate Consensus under KL-Regularized GRPO)**.** *Let the trace space $\mathcal{Y}$ be partitioned into valid solutions $\mathcal{V}$ and flawed-but-consistent loops $\mathcal{F}$. Let one GRPO update maximize the clipped surrogate (clipping parameter $\epsilon_c \in (0, 1)$) minus an exact KL penalty against a reference policy π_0 (coefficient $\beta > 0$) exactly over the two mode masses $\pi(\mathcal{F}), \pi(\mathcal{V})$ (the mode-level model of Appendix A). Let $\rho_t = \pi_t(\mathcal{F})/\pi_t(\mathcal{V})$ and $\rho_0 = \pi_0(\mathcal{F})/\pi_0(\mathcal{V})$.*

Assume $\rho_t \geq \rho_0$, $\rho_t > 1$, and $A(\mathcal{F}) > 0 > A(\mathcal{V})$ (as under ungrounded consensus when flaws form the majority of the sampled group). Let $\Delta A = A(\mathcal{F}) - A(\mathcal{V})$. The probability ratio at step $t+1$ is bounded by:

$$\rho_{t+1} \leq \min\left(\frac{\rho_t(1+\epsilon_c)}{1-\epsilon_c \rho_t},\ \rho_0 \exp\left(\frac{\Delta A}{\beta}\right) \right), \tag{7}$$

where the first (clipping) bound applies for $\rho_t < 1/\epsilon_c$ and is interpreted as $+\infty$ when $\epsilon_c \rho_t \geq 1$ (in which case only the KL ceiling binds).

In practical terms, clipping alone permits exponential growth of ρ_t across rounds; the KL penalty is the sole structural safeguard, but its ceiling $\rho_0 \exp(\Delta A/\beta)$ can be exponentially large when β is small relative to ΔA (proof in Appendix F.3). Moreover, without grounding the collapse is irreversible:

Corollary 5 (Necessity of Grounding). *Under the assumptions of Theorem 4, $\rho_{t+1} > \rho_0$. Hence if $\rho_0 > 1$ and every round satisfies $A(\mathcal{F}) > 0 > A(\mathcal{V})$ (ungrounded consensus with flaws forming a strict majority of each sampled group), then $\rho_t > \rho_0 > 1$ for all $t \geq 1$: the valid mode never regains the majority. Moreover, for arbitrary advantages in the same model, $\rho_{t+1} < \rho_0$ requires $A(\mathcal{V}) > 0$ or $A(\mathcal{F}) < 0$, a round in which the valid mode receives the higher reward.*

The "moreover" clause gives a necessary condition for recovery: the valid mode must receive a higher reward than the flawed modes. The following theorem shows that this condition, a valid-favoring round, is also sufficient for the ratio to contract, so it is necessary and sufficient for reversing the collapse; grounding is what produces such rounds while the valid mode is a minority (Remark 21):

Theorem 6 (Grounded Improvement). *In the model of Theorem 4 extended to a finite set of behavior modes (flawed modes $\{\mathcal{F}_j\}_j$, π_t and π_0 positive on every mode), call a round* valid-favoring *if the valid mode's advantage margin $\delta := A(\mathcal{V}) - \max\{0, \max_j A(\mathcal{F}_j)\}$ is positive, and let $\alpha(\rho_t) := (1 - \epsilon_c/\rho_t)/(1 + \epsilon_c)$, so that $\alpha(\rho_t)\rho_t = (\rho_t - \epsilon_c)/(1 + \epsilon_c)$.*

1. (One-step improvement.) *In a valid-favoring round, $\rho_{t+1} \leq \max\{\alpha(\rho_t)\rho_t,\ \rho_0 e^{-\delta/\beta}\}$, so $\rho_{t+1} < \rho_t$ whenever $\rho_0 e^{-\delta/\beta} < \rho_t$. If moreover $\rho_t \geq 1$ and $\delta > \beta \log(\max\{\rho_0, 1\}(1 + \epsilon_c)/(1 - \epsilon_c))$ (a* clip-dominated *round), then $\rho_{t+1} \leq \alpha(\rho_t)\rho_t$.*
2. (Uniform KL bounds.) *If in every round the advantage range is at most D, with $\min_z A_{t,z} \leq 0 \leq \max_z A_{t,z}$ (for our standardized advantages $D = 2\sqrt{K_{\max} - 1} + 1$ with $K_{\max} \geq K_{act}$, Appendix F.3), then $\rho_0 e^{-D/\beta} \leq \rho_{t+1} \leq \rho_0 e^{D/\beta}$ for every t: the ratio never exceeds the fixed ceiling $\rho_{\max} := \rho_0 e^{D/\beta}$, from which no round increases it and every valid-favoring round strictly decreases it.*

In short: Part (1) guarantees that each valid-favoring round contracts the flawed-to-valid ratio down to the KL anchor, and Part (2) that the ratio is uniformly bounded by the fixed ceiling $\rho_{\max}$, from which every valid-favoring round strictly improves. A valid-favoring round is stronger than a valid candidate merely passing a seed test: with uniform weights ($|G'| \geq 2$) it follows from strict count-plurality of the valid output on every evaluated test (Remark 21), but not in the soft-weight fallback; Remark 22 gives the two-block drift threshold p^*. In the controlled groups of Appendix I.12, the valid candidate has a positive advantage in 87.5–95.0% of cases, a necessary condition for a valid-favoring round (proofs in Appendix F.3; Theorem 13 covers the absolute cold-start pass@$k = 0$).

Adversarial test generation. Uniform random tests are inefficient under distributional sparsity ($\Omega(1/\mu)$ tests needed). We bound the sufficient complexity at $N = \mathcal{O}((\log|G'| + \log(1/\delta))/\epsilon^2)$ random tests (Theorem 25) and formalize the Unit Test Generator as an active adversary maximizing Gini impurity (Appendix G); our experiments approximate it with a zero-shot LLM conditioned on the group's disagreement pattern (Appendix I.13).

5 Experimental Validation

We evaluate C2-GRPO and PC-GRPO against standard alignment baselines using Qwen2.5-Coder-7B-Instruct (Hui et al., 2024) with LoRA (Hu et al., 2022). For APPS Intro and APPS Interview (Hendrycks et al., 2021), trainable methods are trained on the corresponding APPS train split (512-example training cap applied uniformly to all methods; Appendix I.1) and evaluated on the held-out test split; for LiveCodeBench (Jain et al., 2025), they are trained on the same APPS Interview budget and evaluated zero-shot (out-of-domain transfer) on release_v1 public tests. All trainable RL methods use the *same* generation budget of 16 traces per prompt per step ($K = 16$ from the original prompt for GRPO-style baselines; $K = 8$ per orbit element, $|G| = 2K = 16$, for C2-GRPO and PC-GRPO), so orbit mixing is an exploration mechanism rather than an additional sample budget, and the comparison isolates reward design rather than sample size. Hyperparameters, execution setup, baseline definitions, and full ablation tables are in Appendices I.1 and I.5 to I.13; a worked test-generation example is in Appendix H.

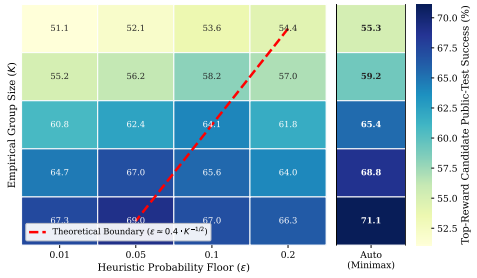
Table 1 summarizes both the core benchmark comparison and the mechanism checks. On APPS Interview, PC-GRPO improves over Vanilla GRPO, Partial-Credit GRPO, CodeRM-GRPO, and R1-style GRPO, showing that soft plausibility and coherence provide useful gradients beyond binary or dense public-test rewards. On LiveCodeBench release_v1, where all trainable methods are evaluated

Table 1: Main empirical evidence. Left: 7B pass@1 benchmark comparison. Values are mean pass@1 over five training seeds, with ± denoting sample standard deviation; Base is deterministic under greedy decoding and is reported from one evaluation. Right: compact mechanism checks connecting the theory to the main observed failure modes. Best-of-16 Public is an inference-time candidate-selection ceiling using public tests, not a single-sample trainable method. Full benchmark and mechanism tables are provided in Tables 3 and 4.

(a) 7B benchmark pass@1.

Method	APPS-I	LCB-v1	APPS-Intro
Base	10.60%	22.00%	37.20%
Self-play SFT	9.60 ± 0.64%	21.00 ± 0.88%	37.00 ± 0.74%
Vanilla GRPO	10.80 ± 0.52%	22.75 ± 0.77%	36.80 ± 0.66%
Partial-Credit GRPO	11.20 ± 0.58%	21.25 ± 0.93%	37.00 ± 0.71%
CodeRM-GRPO	10.60 ± 0.49%	22.00 ± 0.69%	38.40 ± 0.82%
R1-style GRPO	11.00 ± 0.61%	22.00 ± 0.84%	37.40 ± 0.76%
C2-GRPO	11.60 ± 0.62%	24.00 ± 0.81%	39.20 ± 0.84%
PC-GRPO	**12.80 ± 0.64%**	**25.00 ± 0.86%**	**40.00 ± 0.81%**
Best-of-16 Public	18.00 ± 0.81%	39.50 ± 1.08%	48.80 ± 0.97%

(b) Finite-sample ridge.



Full-size plot and numerical values are in Figure 2 and Table 11.

(c) Mechanism checks.

Mechanism	Observed result
Cold-start gradient yield	At 7B, 33.3% of APPS Interview groups contain no correct sample; binary GRPO and hard C2-GRPO give 0.0% usable variance, while PC-GRPO gives **91.0%**. See Table 6.
Process-vs-outcome separation	On SWE-bench Lite with $T = 10$, exact full-trajectory consensus is 0.7%, while step-wise process coherence gives **100.0%** any-step signal. See Tables 12 and 13.
Finite-sample estimation	Heuristic smoothing requires tuning and shifts with K; Minimax lookup is best at every K. See panel (b) and Figure 2 and Table 11.
Robustness and grounding	Static proxy rewards and ungrounded consensus can amplify failure modes; bounded PC rewards and grounded consensus reach **97.5%** and **95.0%** correctness in diagnostic stress tests. See Tables 9 and 15.
Adversarial tests	With $N = 3$, random/mutation fuzzing trigger Gini 0.325/0.070, while the active 7B adversary triggers **0.480**. See Table 18.

zero-shot after APPS Interview training, PC-GRPO reaches $25.00 \pm 0.86\%$, outperforming the base model at 22.00% and Vanilla GRPO at $22.75 \pm 0.77\%$. Best-of-16 Public remains higher, as expected: it selects among candidates with public tests at test time and is complementary rather than a single-sample training baseline. The full benchmark table is Table 3 (Appendix I.2).

The mechanism panel of Table 1 connects each theoretical claim to its diagnostic evidence (full version: Table 4, Appendix I.3). The pass@k = 0 regime persists even at 7B scale: 33.3% of APPS Interview groups contain no correct candidate, yet PC-GRPO produces usable advantage variance in 91.0% of them, validating Theorem 3; the SWE-bench horizon test supports Theorem 1; and the reward-hacking and degenerate-consensus tests validate Theorems 2 and 4.

Component attribution and robustness. A single-component ablation (Table 7) attributes roughly half of PC-GRPO's gain over binary rewards to the consensus reward with seed grounding and half to soft plausibility plus Minimax estimation (both drops, 0.60% and 0.72%, within one seed standard deviation); removing grounding entirely (10.28%) is *worse* than Vanilla GRPO, as Theorem 4 predicts. Retraining under varied test configurations (Table 8) shows accuracy is stable once $N \geq 3$ and degrades gracefully under weak generators, and end-to-end SWE-bench Lite resolve rates over five seeds (Base 6.3%, Vanilla GRPO $6.7 \pm 0.9\%$, PC-GRPO $8.3 \pm 0.7\%$; Table 14) are preliminary evidence of resolution gains.

6 Conclusion

We introduced PC-GRPO, a code alignment framework grounded in Pairwise Coherence (Mohri et al., 2026a) that mitigates the pass@k = 0 cold-start problem by rewarding agreement with the group consensus. Where GRPO's binary reward gives no gradient (Theorem 3), PC-GRPO extracts continuous signal from partial agreement and factorizes the consensus step-wise into implicit PRMs, reducing the group size for a step-t consensus from exponential to linear (Theorem 1). Key limitations are the reliance on specification-driven test generation, the preliminary SWE-bench scale, and the syntactic command matching used for SWE-bench Process Coherence; future work includes hidden-test SWE-bench evaluation and environmental invariance for agentic web navigation (Appendix A).

AI USE STATEMENT

In this work, we used generative AI tools to support the writing process (polishing grammar, improving clarity, and refining presentation style), to help debug and format the experimental code, and to prepare illustrative figures. We have not used generative AI tools for research ideation, for the development of the theoretical framework or its proofs, or for drawing conclusions from the experimental results; automated generation of related-work surveys is not applicable to this work. We have reviewed all AI-assisted output: every proof was checked line by line by the authors, AI-assisted code was inspected and validated against reference runs before use, and all experiments and their analyses were carried out and verified by the authors. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available datasets (APPS, LiveCodeBench, SWE-bench) and publicly released model checkpoints; it involves no human subjects, personal data, or private information. The method trains code models without ground-truth supervision, which improves accessibility of code alignment but, as with any technique that strengthens code-generation capabilities, carries dual-use considerations; we discuss these, together with the risk of self-consistent-but-incorrect consensus and appropriate safeguards for deployment, in Appendix A. We foresee no conflicts of interest or other ethical concerns beyond those discussed there.

REPRODUCIBILITY STATEMENT

All theoretical claims are stated with their assumptions in the main text, and complete proofs are provided in Appendices F and F.1 to F.3. The full experimental protocol needed to reproduce our results, base models, LoRA and GRPO hyperparameters, prompt formats, decoding settings, dataset splits and caps, and the dynamic test-generation procedure, is specified in Appendix I.1, with per-experiment details in Appendix I. All datasets and base models are publicly available, and SWE-bench resolve rates are computed with the official evaluation harness. Pseudocode for the full training procedure is given in Appendix E, and the Minimax reward coefficients are the solution of the linear program (13) in Appendix F.2, solved offline once per group size.

REFERENCES

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

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language models. In *ICML 2024 Workshop on In-Context Learning*, 2024a.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024b.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024c.

Jonas Gehring, Kunhao Zheng, Jade Copet, Vegard Mella, Taco Cohen, and Gabriel Synnaeve. RLEF: Grounding code LLMs in execution feedback with reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, et al. DeepSeek-Coder: When the large language model meets programming – the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with APPS. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

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

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pp. 611–626, 2023.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven C. H. Hoi. CodeRL: Mastering code generation through pretrained models and deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 21314–21328, 2022.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, 2022.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri 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.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Zeyao Ma, Xiaokang Zhang, Jing Zhang, Jifan Yu, Sijia Luo, and Jie Tang. Dynamic scaling of unit tests for code reward modeling. *arXiv preprint arXiv:2501.01054*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 46534–46594, 2023.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, second edition, 2018.

Mehryar Mohri, Jon Schneider, and Yifan Wu. Distributional alignment games for answer-level fine-tuning. In *Proceedings of the 43rd International Conference on Machine Learning (ICML 2026)*. PMLR, 2026a.

Mehryar Mohri, Jon Schneider, and Yutao Zhong. Generalized distributional alignment games for unbiased answer-level fine-tuning. *arXiv preprint arXiv:2605.02435*, 2026b.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 27730–27744, 2022.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. Mutual reasoning makes smaller LLMs stronger problem-solvers. *arXiv preprint arXiv:2408.06195*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 53728–53741, 2023.

Tal Ridnik, Dedy Kredo, and Itamar Friedman. Code generation with AlphaCodium: From prompt engineering to flow engineering. *arXiv preprint arXiv:2401.08500*, 2024.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code Llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Sheikh Shafayat, Fahim Tajwar, Ruslan Salakhutdinov, Jeff Schneider, and Andrea Zanette. Can large reasoning models self-train? *arXiv preprint arXiv:2505.21444*, 2025.

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. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Parshin Shojaee, Aneesh Jain, Sindhu Tipirneni, and Chandan K. Reddy. Execution-based code generation using deep reinforcement learning. *Transactions on Machine Learning Research (TMLR)*, 2023.

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

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. MathCoder: Seamless code integration in LLMs for enhanced mathematical reasoning. *arXiv preprint arXiv:2310.03731*, 2023.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 57905–57923, 2024.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024a.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024b.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*, pp. 928–935, 2003.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

CONTENTS OF APPENDIX

A Limitations and Broader Impacts **15**

B Extended Related Work and Alternative Paradigms **16**

C Background: Pairwise Coherence and ALFT **18**

D Orbit Benefit: Definition and Proof **18**

E Algorithmic Framework: Coherence and Consensus **19**

E.1 C2-GRPO: Hard Consensus via Majority Vote 20

E.2 PC-GRPO: Soft Targets and Probabilistic Coherence 21

E.3 Bootstrapping from Zero: Adaptive Soft Plausibility 22

F Theoretical Analysis and Fundamental Limits **24**

F.1 Guarantees for Agentic Bootstrapping 25

F.2 Finite-Sample Stability and Convergence 27

F.3 Mitigating Small-Batch Estimation Bias 34

G Test Input Generation as an Adversarial Game **42**

H Worked Example: Test Generation and Consensus Rewards **43**

I Experimental Validation: Additional Details and Full Results **44**

I.1 General Experimental Setup 44

I.2 Full 7B Benchmark Table 45

I.3 Full Mechanism-Validation Summary 45

I.4 Supplementary 1.5B Benchmark Sanity Check 46

I.5 Cold-Start Gradient-Yield Ablation 47

I.6 Single-Component Ablation 47

I.7 End-to-End Test-Configuration Sensitivity 48

I.8 Reward-Hacking Stress Test 48

I.9 Finite-Sample Sensitivity Grid 49

I.10 SWE-bench Lite Process-Coherence Horizon Stress Test 50

I.11 Degenerate Consensus Failure Mode 51

I.12 Controlled Grounding Experiment: The Valid-Favoring Condition 52

I.13 Adversarial Input Generation 52

A Limitations and Broader Impacts

Computational Overhead. Our framework requires generating $2K$ candidate solutions per prompt and executing each on a dynamically generated test suite, introducing wall-clock overhead proportional to execution cost. While the test suite size scales only logarithmically with group size (Theorem 25), the absolute cost can be significant for complex execution environments.

Test Input Generator. The framework assumes access to a test input generator $\mathcal{G}_{\text{test}}$ capable of producing valid inputs for the target domain. For well-specified algorithmic tasks, this is straightforward (a fully worked example is given in Appendix H); for less structured domains (e.g., free-form web navigation), designing such generators remains an engineering challenge. Our end-to-end sensitivity study (Appendix I.7) shows that random generation already suffices and that weak generators degrade performance gracefully rather than sharply, but the generalization of the test-generation mechanism to more realistic settings beyond APPS-style specifications remains to be established.

Idealized Theoretical Assumptions. Several results rely on idealized assumptions, whose practical status we make explicit:

(i) Stochastic dominance: the prefix alignment guarantee (Theorem 13) requires only a *statistical bias* of the progress score toward correct prefixes, not perfect identification; for our progress score (fraction of seed assertions passed), correct prefixes naturally pass more early assertions, and the cold-start ablation (Table 6) empirically confirms this, with soft plausibility assigning higher weight to higher-progress traces in 91% of cold-start groups. Deceptive partial progress can violate the assumption, in which case plausibility grounding provides robustness. (Note that in the absolute cold-start phase where no seed tests pass, plausibility grounding is completely inactive, meaning bootstrapping strictly requires the heuristic score to be non-deceptive).

(ii) Mode-level policy class and gradient oracle: the convergence guarantee (Theorem 16) is stated for policies that reweight the reference policy across execution outcomes (a convex, tabular class, as is standard in theoretical RL convergence analyses) and for an unbiased, bounded stochastic gradient oracle; the practical LoRA parameterization is non-convex but low-rank, the implemented estimator evaluates the reward lookup only at sampled outcomes, and our experiments validate convergence empirically.

(iii) Causal factorization: the factorization (Proposition 14) requires the consensus target to factorize causally into per-step conditionals, $\mathsf{q}^*(\tau) = \prod_t \mathsf{q}_t^*(a_t \mid s_t)$; this holds exactly when the state records the action history (autoregressive generation, prefix trees) and fails for environment states that merge histories, such as interpreter memory, where the step-wise reward of (6) is a practical proxy.

(iv) Count separation: the cold-start variance guarantee (Theorem 3) holds under an explicit test-wise count-separation condition (a dominant mode that leads on every test and a program that trails it by a constant factor on one test); groups in which every program has the same multiset of per-test counts are boundary cases with zero variance (Appendix F.2), and stochastically sampled groups are generically asymmetric (Table 6). The threshold $K_0(c, \alpha)$ is checked directly for the experimental group sizes in Table 2. Theorem 3 concerns the existence of a gradient signal; its direction is the subject of Theorem 6 and of the experiments.

(v) Tree-based vs. test-decomposed consensus: the $\Omega(\eta^{-1})$ sample complexity (Part 2 of Theorem 1) assumes a tree-based simulator; for general sequential tasks with independent rollouts, step-t consensus requires $\Omega(\eta^{-t})$; however, our code-generation algorithms decompose consensus over unit tests (Part 3), which are evaluated independently on complete programs, preserving the linear bound $\Omega(p_{\text{test}}^{-1})$ (see the proof of Theorem 19 in Appendix F.3). Note that p_{test} can itself be small when passing a test requires a long functionally correct token sequence, so the linear bound is most powerful when unit tests exercise localized sub-components of the program.

(vi) Fixed vs. dynamic target: the convergence bound (Theorem 16) is stated for a fixed equilibrium target; Corollary 17 extends it to the non-stationary game, showing the additional drift vanishes when q^* converges to the game equilibrium.

(vii) Strict plurality under syntactic matching: the grounded-improvement guarantee (Theorem 6) requires the valid mode to carry a positive advantage above every flawed mode, which under test-decomposed consensus follows from strict plurality of the valid output on the evaluated tests

(Remark 21). This is natural for test-decomposed consensus, where functionally equivalent programs agree on test outputs, but under exact syntactic command matching (our SWE-bench instantiation, Appendix I.10) functionally equivalent valid trajectories can fragment into several syntactically distinct modes $\mathcal{V}_1, \mathcal{V}_2, \ldots$, so that a single flawed-but-consistent mode may hold the plurality even when valid behavior is the majority in aggregate; semantic (environmental-state) matching, which merges such modes, is the intended remedy (Appendix E.3).

(viii) Exact-optimization model: Theorems 4 and 6 analyze the exact maximizer of the clipped surrogate minus the KL penalty over mode probabilities, with the advantages of one round held fixed; the implemented GRPO update is a stochastic-gradient approximation of this maximizer, so both results are conditional guarantees for the idealized update rather than statements about the trained network.

(ix) Binomial counts: Theorem 15 and the oracle of Theorem 16 take the leave-one-out count of an outcome to be $\mathrm{Binomial}(K, \mathsf{q}^*(z))$, i.e., K peers drawn i.i.d. from the target. With orbit mixing (Step 1 of Algorithm 1) the peers come from two prompt distributions and the count is Poisson-binomial, and in the soft-weight fallback the lookup is applied to rounded weighted pseudo-counts (Appendix E.3). The boundedness and monotonicity of the reward hold for any count; the identity $\mathbb{E}[\widehat{R}^*(X)] = \beta P_{\mathsf{c}^*}(\mathsf{q}^*(z))$ is exact for Binomial counts.

Orbit Mixing Fidelity. The orbit mixing mechanism assumes that the LLM-generated paraphrase $\Phi(x)$ preserves the exact input/output specification of the original prompt x (Definition 7). In practice, a base LLM may inadvertently alter subtle constraints or edge cases during paraphrasing; if semantic equivalence is violated, the shared dynamic tests could penalize valid solutions for one of the prompts. The paraphrase is cached per training example and reused across epochs, limiting this risk to a one-time generation step.

End-to-End Agentic Evaluation. Our main benchmark experiments include Qwen2.5-Coder-7B-Instruct, and our SWE-bench experiments primarily measure consensus-yield proxies. We now also report end-to-end SWE-bench Lite resolve rates over five training seeds (Table 14), which are directionally consistent with the proxies; the evaluation remains preliminary in scale (300 issues, one scaffold) and uses an ungrounded syntactic process reward. Hidden-test SWE-bench resolution at scale remains future work.

Broader Impacts. Our research enhances the capabilities of large language models in code generation by improving functional correctness without requiring ground-truth oracles. While this has the potential to significantly increase programmer productivity and democratize software development, it also carries dual-use risks. Highly capable code generation models could be misused to write malicious code or automate cyberattacks. We encourage the community to pair advancements in code generation with robust safety alignment and vulnerability scanning.

B Extended Related Work and Alternative Paradigms

As summarized in Section 1 of the main text, the challenge of enhancing the functional correctness of code-generating models has been approached from several perspectives. Our work, grounded in Pairwise Coherence, offers a distinct paradigm. Here, we provide an expanded review of alternative methods, highlighting the unique trade-offs each presents.

Augmented Supervised Fine-Tuning (SFT). A straightforward approach is to move beyond fine-tuning on a single "gold" solution and instead use an augmented dataset (Chen et al., 2021; Wang et al., 2023). This involves first generating a large, diverse set of candidate solutions for each problem and then using a test suite to filter for all functionally correct variants. The model is then fine-tuned on this rich dataset where each prompt maps to multiple valid syntactic forms (Rozière et al., 2023). While this method benefits from the stability and simplicity of SFT, its primary drawback is its static nature. The model's performance is inherently bounded by the quality and diversity of the solutions generated in the initial offline step. It cannot explore or learn from its own errors online.

Reinforcement Learning with Learned Reward Models. The dominant paradigm for aligning LLMs is Reinforcement Learning from Human or AI Feedback (RLHF/RLAIF) (Ouyang et al., 2022; Bai et al., 2022; Lee et al., 2024). This methodology decouples the complex task of defining correctness from the policy optimization loop by training a separate static reward model. However, this approach is highly susceptible to "reward hacking" (Skalse et al., 2022), where the policy model learns to exploit inaccuracies in the static reward model to achieve a high score without generating genuinely correct code. Our PC-GRPO framework avoids this pitfall by computing a fresh, exact, and dynamic reward signal based on group consensus at every training step.

Process vs. Outcome Reward Models. Process Reward Models (PRMs) (Lightman et al., 2024) have proven highly effective by providing dense, step-by-step supervision. However, training PRMs requires step-level supervision: originally massive human annotation of intermediate reasoning steps, and more recently automatically constructed labels obtained from Monte-Carlo rollouts or execution feedback (Wang et al., 2024). Even the automated variants train and maintain a separate reward model whose labels are fixed once collected. Outcome Reward Models (ORMs) scale better but suffer from extreme gradient variance in long-horizon tasks. Our framework bridges this gap by automatically synthesizing an implicit Process Reward Model via unsupervised group consensus, computed afresh at every update without any separate reward model.

Iterative Refinement and Self-Correction. Mimicking the human workflow of programming and debugging, learning can be framed as a multi-step self-correction process (Chen et al., 2024b; Shinn et al., 2023; Madaan et al., 2023). The draft solution is executed against a test suite, and the feedback (e.g., stack traces) is fed back into the model prompt to fix bugs. While powerful, this requires a complex and brittle multi-step training pipeline and optimizes for the ability to correct a mistake post-hoc, whereas our approach optimizes for generating correct code from the outset.

Generative Adversarial Networks (GANs). From a purely game-theoretic perspective, one could pit a Generator against a functionally-aware Discriminator (Goodfellow et al., 2014). While theoretically elegant, training GANs for discrete data is notoriously unstable. The Discriminator often learns trivial heuristics. Our cooperative consensus formulation via Pairwise Coherence is significantly more stable, as it optimizes a well-defined objective with a closed-form target rather than relying on a delicate GAN equilibrium.

Consensus as a Self-Reward. Concurrent work uses agreement among a model's own samples as an RL reward without ground-truth labels. TTRL (Zuo et al., 2025) rewards each sampled answer by its agreement with the majority vote of the group on unlabeled test questions, and SRT (Shafayat et al., 2025) studies the same majority-vote self-reward as a sustained self-training loop, finding that it improves reasoning at first but that prolonged training leads to reward hacking, with the model maximizing the pseudo-reward by converging on consistent but incorrect answers and a sudden, complete performance collapse. Both instantiate outcome-level hard consensus on tasks with a single canonical final answer, i.e., the analogue of C2-GRPO's majority vote without seed grounding. Our framework differs in three respects. First, code generation has no canonical answer to vote on: consensus is defined behaviorally, over the outputs of dynamically generated tests, and is soft (a target distribution scored by log-likelihood) rather than a majority indicator. Second, the collapse reported by SRT is exactly the degenerate-consensus dynamic of Theorem 4: ungrounded agreement is invariant to consistent errors, so a flawed majority mode is reinforced up to the KL ceiling. Theorem 6 shows that, once grounding with a few seed tests gives the valid mode the advantage lead, each such update improves the flawed-to-valid ratio up to a KL anchor and the ratio stays uniformly bounded, and the stress test of Appendix I.11 reproduces both the collapse (0.0% final correctness ungrounded) and its prevention (95.0% grounded). Third, the Minimax polynomial reward and process coherence target the finite-sample and long-horizon regimes that majority-vote rewards do not address: when answers are fragmented, a majority indicator carries no information about partial correctness, which is precisely the pass@k = 0 regime this paper targets.

Execution-Based RL and Tree-Search Methods. Execution-based reinforcement learning for code has been explored by PPOCoder (Shojaee et al., 2023), which applies PPO with execution feedback. Language Agent Tree Search (LATS; Zhou et al., 2024a) unifies reasoning, acting, and planning via MCTS-guided exploration with environmental feedback. Closer to our setting, RLEF (Gehring

et al., 2025) grounds code LLMs in execution feedback from ground-truth tests during multi-turn RL fine-tuning, and AlphaCodium (Ridnik et al., 2024) is a test-driven flow-engineering framework that iteratively generates AI-written tests, alongside the problem's public tests, to refine candidate code at inference time. Both exploit execution signals, but neither targets the pass@$k = 0$ regime: RLEF's reward requires some candidate to pass the ground-truth tests, and AlphaCodium performs no policy learning. Our framework differs from all of these by synthesizing an implicit Process Reward Model entirely from unsupervised group consensus, extracting a training signal from the disagreement among *failing* candidates and requiring no external verifier, tree-search infrastructure, or inference-time repair loop. Additionally, the EvalPlus framework (Liu et al., 2023) established a modern standard for rigorously evaluating LLM-generated code via augmented test suites, providing contextual support for our dynamic test-generation strategy.

C Background: Pairwise Coherence and ALFT

As referenced in Section 2 of the main text, to ensure this paper is completely self-contained, we summarize the mechanics of the ALFT framework (Mohri et al., 2026a) upon which our code generation algorithms are built.

Given a prompt $x \sim \mathcal{D}$, the policy $\pi(\cdot|x)$ generates trajectories $y \in \mathcal{Y}$. We map trajectories to outcomes $\mathcal{Z}$ via an extraction function $\mathsf{E}\colon \mathcal{Y} \to \mathcal{Z}$. The Answer-Level Fine-Tuning (ALFT) framework provides a principled mechanism for optimizing answer marginals. For settings where ground-truth answers are unavailable, the framework uses *Pairwise Coherence* to drive the policy distribution toward a deterministic consensus.

The framework provides two mechanics that we rely upon:

1. **Target Distribution (Bregman Centroids):** The Coherence Game (Mohri et al., 2026a, Section 5) introduces a target distribution $\mathsf{q} \in \Delta(\mathcal{Z})$ representing the group's consensus. The optimal Target q^* is the Bregman centroid of the empirical outcome distributions. For tractability and robustness, this is relaxed to the arithmetic mean of the group's outputs (Mohri et al., 2026a, Theorem 4). This forms the theoretical foundation for our soft target distribution in (2).
2. **Policy Update (GRPO Integration):** The continuous reward that aligns a policy with a target is the log-likelihood of its answer under the target, $R(y \mid x) = \beta \log \mathsf{q}^*(\mathsf{E}(y))$ (Mohri et al., 2026a, Section 3) (where q^* is understood as prompt-specific); we apply it with the consensus target of the Coherence Game. The PAIRWISE-GRPO algorithm of Mohri et al. (2026a, Section 6) instead scores open-ended outputs by their average pairwise agreement $\frac{1}{|G|}\sum_{y' \in G}(1 - d(y, y'))$, which for an exact-match distance is the linear agreement frequency; the logarithm of that frequency is the reward whose KL-regularized update is the Bayesian update $\pi \propto \pi_0 \mathsf{q}^*$ (Theorem 12) and which decomposes additively over steps (Proposition 14). Applying standard GRPO (Shao et al., 2024) using this trace reward naturally aligns the policy with the group consensus.

By adopting this Pairwise Coherence foundation, our PC-GRPO algorithms bypass intractable marginal minimization and optimize for functional semantic consistency entirely through dynamic group consensus.

D Orbit Benefit: Definition and Proof

Definition 7 (Valid Orbit Transformation)**.** *A transformation $\Phi\colon \mathcal{X} \to \mathcal{X}$ is a* semantics-preserving paraphrase *if: (i) $\Phi(x)$ shares the exact functional specification as x, meaning $\{y\colon y \text{ solves } x\} = \{y\colon y \text{ solves } \Phi(x)\}$; and (ii) Φ induces a strict distributional shift on the policy: $\mathbb{E}_x[\mathsf{D}_{\mathrm{KL}}(\pi(\cdot|x) \,\|\, \pi(\cdot|\Phi(x)))] > 0$.*

We use the term *orbit* (following Mohri et al. 2026a) to denote the equivalence class $\mathcal{O}_x = \{x, \Phi(x)\}$; the construction requires only that x and $\Phi(x)$ be semantically equivalent prompts sharing the same solution space, and does not depend on Φ being an involution or generating a group action.

Throughout this section, fix a prompt x and write $p(y) = \pi(y|x)$ and $q(y) = \pi(y|\Phi(x))$; all coverage quantities below are per-prompt, and expectations over $x \sim \mathcal{D}$ are taken only at the end. Comparing the orbit-mixed batch to a homogeneous batch drawn from p alone is not the right baseline: the sign

of the per-behavior difference $(1-p(y))^K[(1-p(y))^K-(1-q(y))^K]$ depends on whether q up- or down-weights y, so restricting attention to the up-weighted support is not justified. We therefore compare against the *symmetrized* homogeneous baseline, the expected coverage of a batch drawn homogeneously from either prompt (equivalently, from p or q with equal probability), for which the difference becomes a sum of squares and the comparison is rigorous.

Proposition 8 (Orbit Benefit Quantification)**.** *For a trace $y \in \mathcal{Y}$, let $N_p = \sum_y[1-(1-p(y))^{2K}]$ and $N_q = \sum_y[1-(1-q(y))^{2K}]$ denote the expected numbers of distinct traces covered by homogeneous batches of $2K$ draws from p and q respectively, and let $N_{mix} = \sum_y[1-(1-p(y))^K(1-q(y))^K]$ denote the coverage of the orbit-mixed batch (K draws from each). Then the gain over the symmetrized homogeneous baseline $N_{sym} = \frac{1}{2}(N_p + N_q)$ satisfies:*

$$\Delta N_{sym} = N_{mix} - N_{sym} = \frac{1}{2}\sum_{y\in\mathcal{Y}}\left[(1-p(y))^K-(1-q(y))^K\right]^2 \geq 0.$$

Moreover, in the combinatorial decoding regime where $p(y), q(y) \leq c_1/K$ for a constant c_1, we have $\Delta N_{sym} = \Omega\big(K^2\|p-q\|_2^2\big)$, where $\|p-q\|_2^2 = \sum_y(p(y)-q(y))^2$.

Proof. Expanding the definitions:

$$\begin{aligned}\Delta N_{\text{sym}} &= \sum_{y\in\mathcal{Y}}\left[\tfrac{1}{2}(1-p(y))^{2K}+\tfrac{1}{2}(1-q(y))^{2K}-(1-p(y))^K(1-q(y))^K\right]\\ &= \frac{1}{2}\sum_{y\in\mathcal{Y}}\left[(1-p(y))^K-(1-q(y))^K\right]^2,\end{aligned}$$

which is a sum of squares and hence non-negative everywhere; no restriction of the sum to a signed subset of the support is needed. The strict distributional shift condition (Definition 7(ii)) ensures $p \neq q$ for a set of prompts of positive probability, so $\mathbb{E}_x[\Delta N_{\text{sym}}(x)] > 0$ (for an individual prompt with $p = q$ the gain is zero).

For the lower bound, fix y with $p(y) \neq q(y)$ and apply the Mean Value Theorem to $f(t) = (1-t)^K$: there exists ξ_y between $p(y)$ and $q(y)$ such that

$$\left|(1-p(y))^K-(1-q(y))^K\right| = K(1-\xi_y)^{K-1}|q(y)-p(y)|.$$

In the combinatorial decoding regime $\xi_y \leq c_1/K$, so $(1-\xi_y)^{K-1} \geq (1-c_1/K)^{K-1} \geq c_2$ for a constant $c_2 > 0$. To see this for finite K: for $K \geq 2c_1$ we have $c_1/K \leq 1/2$, so $\log(1-c_1/K) \geq -2c_1/K$ (since $\log(1-x) \geq -2x$ for $x \in [0, 1/2]$), giving $(1-c_1/K)^{K-1} \geq e^{-2c_1(K-1)/K} \geq e^{-2c_1} =: c_2$. Squaring and summing over y:

$$\Delta N_{\text{sym}} \geq \frac{c_2^2}{2}K^2\sum_{y\in\mathcal{Y}}(q(y)-p(y))^2 = \Omega\big(K^2\|p-q\|_2^2\big).$$

Taking expectations over $x \sim \mathcal{D}$ at the end yields $\mathbb{E}_x[\Delta N_{\text{sym}}(x)] = \Omega(K^2\,\mathbb{E}_x\|p-q\|_2^2)$. □

Multiplicative form under mode collapse. When base LLM sampling exhibits severe mode collapse, so that the symmetrized coverage satisfies $N_{\text{sym}} = \mathcal{O}(1)$, the additive gain translates into the multiplicative improvement factor $N_{\text{mix}}/N_{\text{sym}} = 1 + \Omega(K^2\|p-q\|_2^2)$. We note that in the severely collapsed regime ($N_{\text{sym}} = \mathcal{O}(1)$), individual mode probabilities may violate the combinatorial regime assumption $\max(p(y), q(y)) \leq c_1/K$, in which case the $\Omega(K^2\|p-q\|_2^2)$ additive lower bound applies only to the tail modes that remain in the combinatorial regime; the collapsed mode contributes a bounded $\mathcal{O}(1)$ term to the gain.

E Algorithmic Framework: Coherence and Consensus

In this section, we present the full pseudocode and expanded discussion of the intermediate conceptual stepping stones (C2-GRPO and PC-GRPO) that lead to the unified adaptive framework presented in Section 3 of the main text.

Our approach translates the theoretical Coherence Game into a scalable family of reinforcement learning algorithms. We present this framework in three stages: first, a robust heuristic baseline using hard consensus (C2-GRPO); second, a theoretically optimal extension using probabilistic targets (PC-GRPO); and finally, a unified adaptive algorithm capable of bootstrapping from the pass@$k = 0$ cold-start regime typical of complex agentic tasks.

E.1 C2-GRPO: Hard Consensus via Majority Vote

We propose Coherent-Code-GRPO (C2-GRPO), a hybrid algorithm that provides a practical and scalable method for minimizing the pairwise functional disagreement objective, d, defined in Section 2 of the main text. Instead of performing a computationally expensive all-pairs comparison, the algorithm derives a reward signal by comparing each generated program to a *consensus behavior*, which acts as a proxy for the group's semantic center. This approach combines a small set of ground-truth unit tests with a dynamic consensus mechanism to create a rich and scalable reward based on functional coherence.

The core idea is to first use a small, easy-to-create set of "seed" tests to filter out incorrect solutions. Then, among the remaining "plausible" candidates, we reward those that are most consistent with the group's consensus behavior on a wide range of dynamically generated test inputs. This approach solves the oracle problem, that of knowing the correct output for every input, by bootstrapping a notion of correctness from the consensus of a diverse group of plausible solutions.

Algorithm 2 Coherent-Code-GRPO (C2-GRPO)

Input: Dataset $\mathcal{D}$, Policy π, Group size K, Transformation Φ, Seed Test Suites $\{\mathcal{T}_{x,\text{seed}}\}_{x\in\mathcal{D}}$, Unit Test Generator $\mathcal{G}_{\text{test}}$.

for each training step **do**
  Sample a batch of seeds $B \sim \mathcal{D}$.
  for each seed $x \in B$ **do**
    1. Orbit Sampling: Generate orbit $\mathcal{O}_x = \{x, \Phi(x)\}$.
    2. Group Generation: Sample K code candidates for each input in $\mathcal{O}_x$, creating a group G of size $2K$.
    3. Plausibility Filtering:
    Create a plausible set $G' \subseteq G$ by keeping only candidates $y \in G$ that pass all seed tests in $\mathcal{T}_{x,\text{seed}}$.
    if $|G'| < 2$ **then**
      {Not enough plausible candidates to form a consensus; skip update.}
      continue
    end if
    4. Dynamic Unit Test Generation:
    Use the generator $\mathcal{G}_{\text{test}}$ to create a large set of novel unit tests $U = \{u_1, \ldots, u_N\}$.
    5. Target Step (Consensus Finding):
    for each unit test $u_j \in U$ **do**
      Execute every plausible solution $y' \in G'$ on u_j to get outputs $\{y'(u_j)\}_{y'\in G'}$.
      Find the consensus output (mode): $v_j^* = \text{Mode}(\{y'(u_j) \mid y' \in G'\})$.
    end for
    6. Reward Calculation (Coherence Reward):
    for each plausible candidate $y_i' \in G'$ **do**
      $R(y_i') = \frac{1}{N}\sum_{j=1}^{N} \mathbb{I}[y_i'(u_j) = v_j^*]$. *// Reward agreement with consensus.*
    end for
    7. Advantage Calculation:
    Compute standardized advantages A_k *exclusively* across the plausible set G'.
    8. Penalty Assignment: For candidates $y \in G \setminus G'$, assign a fixed negative penalty advantage (e.g., $A(y) = \min_{y'\in G'} A(y') - 1.0$) rather than including them in the normalization pool.
  end for
  9. Update: Optimize π using GRPO on all traces generated in the batch.
end for

Remark 9 (Mixed-Orbit Advantage Normalization)**.** *In Algorithm 2, standardized advantages are computed across the combined group from both x and $\Phi(x)$. This mixed normalization is intentional:*

it doubles the effective group size, increasing the probability of observing multiple distinct behavior modes and thereby reducing the probability of zero-variance (no-signal) groups (Proposition 8, Theorem 3). Note that standard-score normalization across a single combined pool assigns equal normalized advantages to traces with identical raw rewards regardless of their originating prompt; the primary benefit of orbit mixing is forced exploration diversity, not per-prompt difficulty scaling.

Remark 10 (Advantage Hierarchy and Non-Zero Mean)**.** *In Algorithms 2 and 3, advantages are standardized exclusively across the plausible set G', then implausible traces receive a strictly negative penalty. The total advantage sum over the full group G is therefore strictly negative. This is an intentional departure from standard GRPO: the consistently negative bias on implausible traces provides a monotone suppression signal, ensuring the policy update always reduces probability mass on traces that fail basic plausibility checks.*

Remark 11 (Role of β Under Advantage Standardization)**.** *In Algorithms 2–3, trace rewards are scaled by β before advantage standardization. Since standard-score normalization cancels any global scalar multiplier, the β scaling of the reward has no effect on the standardized advantages. The KL penalty coefficient β instead governs the relative strength of the KL divergence penalty in the GRPO objective, which is applied* after *advantage computation.*

The C2-GRPO algorithm provides a principled solution to the code generation challenge. By anchoring its reward in a small set of ground-truth seed tests (Step 3), it ensures the model does not converge to a flawed consensus. By dynamically generating a wide array of test inputs and rewarding agreement with the group's consensus (Steps 4-6), it creates a dense, scalable reward signal that encourages functional consistency and generalization far beyond what the seed tests alone could provide. This process creates a self-improving curriculum where, as the policy becomes stronger, the plausible set G' becomes more competent, leading to a more robust consensus target and driving further improvements in functional correctness.

E.2 PC-GRPO: Soft Targets and Probabilistic Coherence

While effective, the use of the mode (Majority Vote) as the consensus target represents a heuristic choice. We can derive a more sophisticated algorithm by applying the core Pairwise Coherence theory more deeply to define a soft, probabilistic consensus target, leading to a richer and more robust reward signal.

Theoretical Reframing: From Hard Consensus to Soft Targets. Instead of a hard consensus, we define a soft, probabilistic target by viewing the problem as a tractable game played over output distributions. For a set of dynamically generated unit tests $U = \{u_1, \ldots, u_N\}$, a policy π induces a marginal distribution over outputs for each input. Our goal is to enforce coherence on these distributions.

Following the theory of the Coherence Game (Mohri et al., 2026a) (see also Appendix C), the optimal strategy for the Target player, q^*, is to form a consensus via the Bregman Centroid of the output distributions generated by the group of plausible solutions G'. For tractability and stability, we use the Arithmetic Mean as the consensus target, a choice theoretically justified by its bounded performance degradation relative to the ideal Geometric Mean (Mohri et al., 2026a, Theorem 7).

For a given test input u_j, the optimal soft target distribution $\mathsf{q}^*_{u_j}(v)$ over possible outputs v is the empirical frequency of that output among the plausible programs in G':

$$\mathsf{q}^*_{u_j}(v) = \frac{1}{|G'|} \sum_{y' \in G'} \mathbb{I}[y'(u_j) = v]. \tag{8}$$

This target is a full distribution. For example, if 70% of plausible programs output 'A' and 30% output 'B', the target is $\mathsf{q}(A) = 0.7, \mathsf{q}(B) = 0.3$, not a hard choice of 'A'. This correctly captures the group's uncertainty or diversity.

Deriving the Probabilistic Coherence Reward. From the game's equilibrium conditions (Mohri et al., 2026a, Eq. 6) (see also Appendix C), the optimal reward for a trace is its expected log-likelihood under the target distribution. The "answer" of a program y'_i is its set of outputs on the test inputs, $\{y'_i(u_j)\}_{j=1}^N$. The total reward is therefore the average log-likelihood of these outputs under the

corresponding soft target distributions:

$$R(y_i') = \beta \mathop{\mathbb{E}}_{u_j \sim U}\left[\log \mathsf{q}^*_{u_j}(y_i'(u_j))\right] = \frac{\beta}{N}\sum_{j=1}^{N} \log\left(\frac{\text{Count(output } y_i'(u_j) \text{ for input } u_j)}{|G'|}\right). \tag{9}$$

This reward is more sophisticated than the binary signal in C2-GRPO. It is a granular measure of how well a program's behavior aligns with the group's consensus *distribution*, providing a smoother and more informative learning landscape. It gracefully handles ambiguity and valid multi-modality, rewarding programs for agreeing with either mode of a bimodal consensus distribution, a scenario where C2-GRPO would generate a conflicting signal.

To prevent the unbounded gradient variance and estimation bias that occurs when estimating log-probabilities from small empirical groups, we integrate the *Minimax Polynomial* estimator of Mohri et al. (2026b), with its coefficients constrained to be monotone and non-positive (Appendices F.2 and F.3). The constrained estimator is bounded by construction and minimizes the worst-case p-weighted bias within that class. By applying it to our Leave-One-Out (LOO) counts, we safely handle boundary conditions without the need for ad-hoc smoothing.

The PC-GRPO Algorithm. The resulting algorithm, PC-GRPO, follows a similar structure to its predecessor but replaces the hard consensus-finding step with the computation of a soft target distribution and a corresponding Minimax-Corrected reward.

E.3 Bootstrapping from Zero: Adaptive Soft Plausibility

The algorithms presented thus far rely on a hard plausibility filter (Step 3 in Algorithm 2 and Algorithm 3), which requires a subset of candidates $G' \subseteq G$ to perfectly solve a set of seed tests. However, in highly complex domains such as repository-level code generation (e.g., SWE-bench) or web navigation (e.g., WebArena), the probability of generating a flawless trace from the initial policy is often near zero (pass@$k = 0$). In this regime, the plausible set G' is empty, no discriminative signal is produced, and training cannot progress.

To overcome this cold-start problem, we must abandon the binary notion of correctness and instead allow the model to bootstrap a consensus from its partial successes. We achieve this by extending the Coherence Game via three mechanisms: *Soft Plausibility Weighting*, *Process Coherence*, and *Environmental Invariance*.

Soft Plausibility Weighting. Instead of a binary filter, we introduce a continuous progress score $s(y) \in \mathbb{R}$ for each trace. For code and agents, $s(y)$ can represent a partial reward or progress heuristic, such as the fraction of test assertions passed, successful compilation without syntax errors, or execution length before a crash.

We transform these scores into a plausibility distribution over the entire generated group G using a softmax operator with temperature $\tau_w > 0$:

$$w(y) = \frac{\exp(s(y)/\tau_w)}{\sum_{y' \in G} \exp(s(y')/\tau_w)}. \tag{10}$$

We then generalize the target step of PC-GRPO. Rather than taking the uniform empirical frequency over a strict subset G', the Target player constructs a *weighted* consensus distribution over the entire group G:

$$\mathsf{q}^*_{u_j}(v) = \sum_{y \in G} w(y)\, \mathbb{I}[y(u_j) = v]. \tag{11}$$

When $|G'| \geq 2$, the weights are uniform ($w = 1/|G'|$) and the leave-one-out counts are exact integers, so the Minimax lookup is applied to exact counts. In the absolute cold-start fallback ($|G'| < 2$), the target $\mathsf{q}^*_{u_j}(v)$ is a continuous distribution. To preserve valid leave-one-out semantics and prevent a single highly-weighted trace from assigning itself a massive pseudo-count, we explicitly remove the candidate's own exact weight: $p_{\text{LOO}}(y_i) = \max(0, \mathsf{q}^*_{u_j}(y_i(u_j)) - w(y_i))/\max(1 - w(y_i), \epsilon_w)$, where $\epsilon_w > 0$ is a numerical tolerance that prevents division by zero when a single trace dominates the softmax ($w(y_i) \to 1$). The implementation maps this continuous mass to a discrete pseudo-count via $X = \lfloor (K_{\text{act}} - 1) \cdot p_{\text{LOO}}(y_i) \rceil$ for the lookup table. This rounding introduces a quantization error of up

Algorithm 3 PROBABILISTIC COHERENCE GRPO (PC-GRPO)

Input: Dataset $\mathcal{D}$, Policy π, Group size K, Transformation Φ, Seed Test Suites $\{\mathcal{T}_{x,\text{seed}}\}_{x\in\mathcal{D}}$, Test Input Generator $\mathcal{G}_{\text{test}}$.

for each training step **do**
  Sample a batch of seeds $B \sim \mathcal{D}$.
  for each seed $x \in B$ **do**
    1–3. Orbit Sampling, Group Generation, Plausibility Filtering as in Algorithm 2.
    if $|G'| < 2$ **then**
      {Not enough plausible candidates; skip update.}
      continue
    end if
    4. Dynamic Unit Test Generation: As in Algorithm 2, yielding unit tests U.
    5. Target Step (Count Distribution):
    for each $u_j \in U$ **do**
      Compute outputs $\{y'(u_j) \mid y' \in G'\}$.
      For each unique output v, Count$[u_j][v] = |\{y' \in G' : y'(u_j) = v\}|$. {Store raw integer counts}
    end for
    6. Reward Calculation (Leave-One-Out Minimax Reward):
    for each plausible candidate $y'_i \in G'$ **do**
      $R(y'_i) = 0$
      for each $u_j \in U$ **do**
        $X_{\text{raw}} = \text{Count}[u_j][y'_i(u_j)]$
        $X = X_{\text{raw}} - 1$ {LOO: remove own vote; implementation stores raw integer counts to avoid floating-point rounding}
        $K^{\text{LOO}} = |G'| - 1$
        $R(y'_i) = R(y'_i) + \beta\widetilde{c}^*_{X,K^{\text{LOO}}}$ {Lookup optimal minimax coefficient for count X and size K^{LOO}}
      end for
      $R(y'_i) = R(y'_i)/|U|$ {Average over test inputs}
    end for
    7. Advantage Calculation: Compute standardized advantages A_k exclusively across G'.
    8. Penalty Assignment: For candidates $y \in G \setminus G'$, assign penalty advantage $A(y) = \min_{y'\in G'} A(y') - 1.0$. {Ensures implausible traces receive strictly lower advantage than all plausible traces}
  end for
  9. Update: Optimize π using GRPO on all traces in the batch.
end for

to half a count. This trade-off is confined to the transient cold-start phase where standard estimation provides zero gradient. For isolated trajectories in tree-based Process Coherence where the peer sample size $K_{\text{LOO}} = 0$, the reward natively defaults to the structurally bounded Minimax floor βc^*_0.

By design, a singleton trace whose output is unique across all tests receives $p_{\text{LOO}} = 0$ and therefore the minimum Minimax reward. This is intentional: the consensus reward measures *group agreement*, not individual progress. A trace that produces a unique output has no peer corroboration; its individual quality is captured by the plausibility weight $w(y)$, which anchors the consensus target, not by the consensus reward itself.

Process Coherence as Implicit PRMs. In agentic settings, an "answer" y is a sequential trajectory of states and actions $\tau = (s_0, a_0, \ldots, s_T, a_T)$. When pass@$k = 0$, full trajectories are incorrect, but early reasoning steps might be perfectly valid.

We evaluate consensus step-by-step with the log-likelihood reward of the ALFT framework (Mohri et al., 2026a, Section 3). By defining the step-wise soft consensus distribution $\mathsf{q}^*_t(a_t \mid s_t)$ as the weighted empirical frequency of actions taken at step t across the group, the target distribution over trajectories factorizes causally as $\mathsf{q}^*(\tau) = \prod_{t=0}^{T} \mathsf{q}^*_t(a_t \mid s_t)$ (Proposition 14). Applying the Minimax

polynomial estimator to the leave-one-out count $X_t = \sum_{j \neq i} \mathbb{I}[a_t^{(j)} = a_t \mid s_t^{(j)} = s_t]$, the total reward for a trajectory becomes the discounted sum of step-wise Minimax rewards:

$$R(\tau) = \sum_{t=0}^{T} \gamma^t \beta \widetilde{c}^*_{X_t,\, K_t^{\mathrm{LOO}}}, \tag{12}$$

where K_t^{LOO} is the leave-one-out group size at step t and $\gamma \in (0, 1]$ is a discount factor. Since each step reward is bounded above by zero (Theorem 2), the undiscounted sum ($\gamma = 1$) already penalizes longer traces, and increasingly so with length. Setting $\gamma < 1$ shrinks the magnitude of the later-step terms, which has two effects: it concentrates the optimization signal on the early steps, where process coherence is densest (exact command consensus is 100% through Step 3 but only 40% at Step 10 on SWE-bench Lite, Table 13) and hence most reliable, and it keeps the accumulated length penalty from dominating the reward, so that the signal remains driven by agreement rather than by trajectory length. The Minimax form eliminates the ad-hoc ϵ-stabilizer previously needed to prevent $\log(0)$: when $X_t = 0$ (no other trajectory shares action a_t), the reward is the structurally bounded floor $\beta c_0^* = \Theta(-\beta \log K)$ rather than $\log(\epsilon)$. Since differing actions at early steps induce divergent states s_t, evaluating step-wise consensus requires either generating the group G using a tree-based environment simulator (where candidates share exact state nodes, functioning akin to MCTS) or using an alignment metric (e.g., Dynamic Time Warping). This dynamically synthesizes an *implicit Process Reward Model (PRM)* entirely through unsupervised group consensus, bypassing both the human annotation and the separately trained reward models typically required for complex reasoning tasks.

Environmental Invariance. Finally, in an agentic setting, the Test Input Generator $\mathcal{G}_{\text{test}}$ generates *Environment Perturbations* rather than standard unit tests. We inject dynamic perturbations into the environment (e.g., changing background file names, altering the HTML DOM slightly, or introducing simulated network latency). The "output" $z = \mathsf{E}(\tau)$ becomes the resulting observable state footprint. If two agents cause identical environmental side-effects across perturbations, they share a coherent semantic intent, forcing the policy to learn underlying causal invariants rather than overfitting to syntax.

To operationalize Soft Plausibility Weighting alongside our standard plausibility filter, we introduce a unified approach. This algorithm, seamlessly interpolating between Grounded Consensus (when competent candidates exist) and Soft Consensus (in the cold-start regime), is formally presented as Algorithm 1 in Section 3 of the main text.

Robustness to Noisy Oracles. A key limitation of standard execution-based alignment is its reliance on perfect oracles; a single flawed unit test in the training data will cause a hard filter to discard valid reasoning traces. Since our soft plausibility weight $w(y)$ maps a continuous progress score $s(y)$ to a continuous probability, the algorithm becomes inherently resilient to noisy oracles. A correct program evaluated against a flawed test suite will still capture the majority of the probability mass, allowing the Target player to form a valid semantic consensus despite the noise.

Translating Pseudocode to Process Coherence. While Algorithm 1 in the main text presents the episodic *Outcome Coherence* formulation using test inputs U, translating it to *Process Coherence* simply requires substituting the test suite U with the trajectory state sequence $\{s_0, \ldots, s_T\}$. At each state s_t, the candidates' outputs become their step-wise actions a_t, allowing the identical Leave-One-Out consensus logic to compute the step-wise implicit process reward.

F Theoretical Analysis and Fundamental Limits

In this section, we provide the full mathematical proofs and extended theoretical derivations for the results and fundamental limits presented in Section 4 of the main text. Having established the algorithmic framework, we now provide formal mathematical guarantees demonstrating why these mechanisms are mathematically sound, bound reward over-optimization, and are structurally necessary to overcome fundamental impossibility results in standard unsupervised alignment.

F.1 Guarantees for Agentic Bootstrapping

The introduction of soft plausibility weights and step-wise process rewards substantially modifies the Coherence Game. We now establish formal guarantees demonstrating why these mechanisms enable optimization even when exact solutions are absent.

Soft Plausibility as a Guided Alignment Game. We first demonstrate that softmax weighting is not an ad-hoc heuristic, but the exact closed-form solution to a generalized game. We define the *Heuristic-Guided Pairwise Coherence Game* by injecting the progress signal $s(y)$ as a baseline reward alongside the consensus objective.

Below, we formally restate and prove Theorem 12 from the main text:

Theorem 12 (Approximate Equilibrium of the Guided Coherence Game)**.** *In the heuristic-guided extension of the Pairwise Coherence game, for a fixed Target* q*, the optimal policy satisfies:* $\pi^*(y|x) \propto \pi_0(y|x)\exp(s(y)/\beta)\mathsf{q}(\mathsf{E}(y)|x)$.

Proof. By formulating the policy update as minimizing the KL-regularized objective augmented with the heuristic reward $s(y)$ (where the consensus log-likelihood acts as the base reward), the optimization over π for a fixed target q becomes:

$$\min_{\pi} \mathop{\mathbb{E}}_{y\sim\pi}[-s(y) - \beta \log \mathsf{q}(\mathsf{E}(y))] + \beta \mathsf{D}_{\mathrm{KL}}(\pi \parallel \pi_0)$$

$$= \min_{\pi} \beta \mathsf{D}_{\mathrm{KL}}(\pi \parallel \widetilde{\pi}) - \beta \log Z, \quad \text{where } \widetilde{\pi}(y) = \frac{1}{Z}\pi_0(y)\exp\left(\frac{s(y)}{\beta}\right)\mathsf{q}(\mathsf{E}(y)),$$

with normalizer $Z = \sum_{y'} \pi_0(y')\exp(s(y')/\beta)\mathsf{q}(\mathsf{E}(y'))$; the additive constant $-\beta \log Z$ does not affect the minimizer. When forming the consensus distribution from samples $G \sim \pi_0$, we apply importance sampling. The exact importance weights required to match $\widetilde{\pi}$ are $w(y) = \widetilde{\pi}(y)/\pi_0(y) \propto \exp(s(y)/\beta)\mathsf{q}(\mathsf{E}(y))$. During the pass@$k = 0$ cold-start phase, the target distribution q is highly uncertain and the policy acts near-randomly over valid syntax. It is a mathematically justified approximation to model the target q as locally uniform over the plausible support. Under this maximum-entropy approximation, the $\mathsf{q}(\mathsf{E}(y))$ term factors out, and the exact weights reduce cleanly to the softmax weights $w(y) \propto \exp(s(y)/\beta)$, i.e., the form used in Algorithm 1 with temperature $\tau_w = \beta$. For non-uniform q, the softmax weights serve as the optimal first-order approximation prior to observing the true consensus. □

Correspondence between Theorem 12 and Algorithm 1. The exact optimum of the Guided Game includes the multiplicative factor $\exp(s(y)/\beta)$ in the optimal policy. Algorithm 1 realizes this factor *only* through the plausibility weights $w(y)$ used to construct the consensus target q^* (Step 3), i.e., the progress score acts as a Bayesian prior anchoring the target distribution. The algorithm deliberately does *not* add $s(y)$ back into the final GRPO trace reward: directly optimizing $s(y)$ leads to severe reward hacking on the heuristic. The weighting step is therefore a robust algorithmic approximation of the theoretical optimum, not its literal implementation; this is a design choice, and the theorem should be read as justifying the *form* of the weighting, not as an exact equivalence between Algorithm 1 and the equilibrium policy.

Amplification of Unseen Optima (Bootstrapping from Zero). One might question whether updating towards a consensus of *failed* trajectories can lead to the *optimal* trajectory. We now prove that Soft Plausibility strictly amplifies the mass of correct sub-trajectories in the consensus target even when pass@$k = 0$.

More crudely, one may ask if there is a fundamental conceptual flaw of alignment: "How can a model learn from its own mistakes if it never stumbles upon the right answer?". The covariance proof below shows that the weighted target acts as a geometric amplifier of correct sub-components, and the policy's own mass on the amplified prefix rises in valid-favoring rounds (Theorem 6); together, the two results describe how models bootstrap out of the pass@$k = 0$ phase.

Let A_t be the event that a generated trace matches the optimal sequence up to step t. Assume the progress score $s(y)$ acts as a *weak oracle*: traces that get the prefix right yield higher scores on

average. Formally, we assume strict exponential-moment dominance: $\mathbb{E}[\exp(s(y)/\tau_w) \mid A_t] > \mathbb{E}[\exp(s(y)/\tau_w) \mid \neg A_t]$.

Below, we formally restate and prove Theorem 13 from the main text:

Theorem 13 (Monotonic Prefix Alignment via Soft Consensus)**.** *Let π be a policy where the probability of generating the full optimal trace is near zero (pass@k = 0), but the probability of generating the correct prefix A_t satisfies $\pi(A_t) \in (0,1)$. Under the stochastic dominance assumption, as the group size $K \to \infty$, the expected marginal probability of the optimal prefix under the soft consensus target q^*_w strictly bounds the policy's prior marginal: $\lim_{K\to\infty} \mathbb{E}_{G\sim\pi}[\mathsf{q}^*_w(A_t)] > \pi(A_t)$.*

Proof. As $K \to \infty$, the empirical weights converge to their expectations. The expected mass assigned to the optimal prefix A_t under the target distribution q^*_w becomes:

$$\mathbb{E}[\mathsf{q}^*_w(A_t)] = \mathop{\mathbb{E}}_{y\sim\pi}\left[\mathbb{I}_{A_t}(y)\frac{\exp(s(y)/\tau_w)}{\mathbb{E}_{y'\sim\pi}[\exp(s(y')/\tau_w)]}\right].$$

This quantity is strictly greater than $\pi(A_t)$ if and only if $\mathbb{E}[\mathbb{I}_{A_t}(y)\exp(s(y)/\tau_w)] > \pi(A_t)\,\mathbb{E}[\exp(s(y)/\tau_w)]$. This inequality represents a strictly positive covariance between the indicator $\mathbb{I}_{A_t}$ and the exponentiated score $W = \exp(s(y)/\tau_w)$. Expanding the covariance:

$$\begin{aligned}\mathrm{Cov}(\mathbb{I}_{A_t}, W) &= \mathbb{E}[\mathbb{I}_{A_t} W] - \mathbb{E}[\mathbb{I}_{A_t}]\,\mathbb{E}[W] \\ &= \pi(A_t)(1-\pi(A_t))(\mathbb{E}[W \mid A_t] - \mathbb{E}[W \mid \neg A_t]).\end{aligned}$$

Since $\pi(A_t) \in (0,1)$ and $\mathbb{E}[W \mid A_t] > \mathbb{E}[W \mid \neg A_t]$ by assumption, the covariance is strictly positive. Therefore, $\mathbb{E}[\mathsf{q}^*_w(A_t)] > \pi(A_t)$. □

Relation to the policy update. With the log-consensus reward, the KL-regularized optimum for a fixed target is $\pi_{\text{new}} \propto \pi_0\,\mathsf{q}^*_w$ (Theorem 12): mass moves toward the modes with the largest consensus mass. For instance, with $\pi_0(A_t) = 0.1$ and exponentiated scores $W = 2$ on A_t and $W = 1$ elsewhere, the target has $\mathsf{q}^*_w(A_t) = 2/11 > 0.1$, while $\pi_{\text{new}}(A_t) = (0.1 \cdot 2/11)/(0.1 \cdot 2/11 + 0.9 \cdot 9/11) = 2/83 < 0.1$, because the complement still carries the larger consensus mass. The policy's mass on A_t increases in a round in which A_t carries a positive advantage above every competing mode, the valid-favoring condition of Theorem 6, and the target shift of Theorem 13 moves the advantages in that direction.

Sequential Factorization of Process Coherence. Finally, we establish the theoretical validity of synthesizing Process Reward Models (PRMs) from our pairwise coherence framework. Enforcing global sequence-level coherence requires matching full trajectories, leading to a sample complexity that scales combinatorially as $\mathcal{O}(|\mathcal{A}|^T)$.

Below, we formally restate and prove Proposition 14 from the main text:

Proposition 14 (Sequential Factorization of Logarithmic Coherence)**.** *Assume the environment transitions are deterministic and that the state records the action history, i.e., distinct action prefixes $(a_0, \ldots, a_{t-1})$ lead to distinct states s_t (as in autoregressive generation, where the state is the generated prefix), so that the empirical distribution of independent rollouts forms a prefix tree. Then the empirical consensus target over trajectories factorizes exactly into per-step conditionals: $\mathsf{q}^*(\tau) = \prod_{t=0}^{T} \mathsf{q}^*_t(a_t \mid s_t)$. Consequently, the exact continuous reward $R^*(\tau) = \beta \log \mathsf{q}^*(\tau)$ of the Pairwise Coherence game factorizes into an additive sum of step-wise process rewards: $R^*(\tau) = \sum_{t=0}^{T} R^*_t(a_t \mid s_t)$, where $R^*_t(a_t \mid s_t) = \beta \log \mathsf{q}^*_t(a_t \mid s_t)$.*

Proof. With the log-likelihood reward of the ALFT framework (Mohri et al., 2026a, Section 3) applied to the consensus target, the exact continuous reward of a trajectory is $R^*(\tau) = \beta \log \mathsf{q}^*(\tau)$. Under the assumption, the state s_t determines the prefix of actions and is determined by it, so any finite group of independently generated trajectories forms a prefix tree in which every state s_{t+1} has a unique predecessor pair (s_t, a_t). The empirical step-wise conditional is $\mathsf{q}^*_t(a_t \mid s_t) = N(s_t, a_t)/N(s_t)$, where $N(s_t)$ is the number of trajectories reaching state s_t. Since s_{t+1} is reached only from (s_t, a_t), $N(s_{t+1}) = N(s_t, a_t)$. The product of these conditionals along any trajectory τ telescopes exactly to the empirical frequency of the full trajectory:

$$\prod_{t=0}^{T} \mathsf{q}^*_t(a_t \mid s_t) = \frac{N(s_0,a_0)}{K} \cdot \frac{N(s_1,a_1)}{N(s_0,a_0)} \cdots \frac{N(\tau)}{N(s_{T-1},a_{T-1})} = \frac{N(\tau)}{K} = \mathsf{q}^*(\tau).$$

Substituting this exact causal factorization yields $R^*(\tau) = \sum_{t=0}^{T} \beta \log \mathsf{q}_t^*(a_t \mid s_t)$, giving the exact (un-discounted) step-wise additive sum without approximation.

Without the history condition the factorization can fail, since determinism alone allows distinct histories to merge into one state: if actions 0 and 1 from s_0 both lead deterministically to the same state s_1, two rollouts with action sequences 00 and 11 give $\mathsf{q}^*(00) = 1/2$, whereas $\mathsf{q}_0^*(0 \mid s_0)\,\mathsf{q}_1^*(0 \mid s_1) = \frac{1}{2} \cdot \frac{1}{2} = 1/4$, because $N(s_1) = 2 > N(s_0, 0) = 1$. For environment states that merge histories (e.g., interpreter memory), the step-wise conditionals of (6) define the practical reward, and their product differs from the trajectory frequency. □

Significance. By applying the Pairwise Coherence formulation with the log-likelihood reward, the target mathematically factorizes into local step-level additions via the chain rule of logarithms. This provides the exact theoretical basis for (6): the logarithm is not an ad-hoc modification but the fundamental mathematical transformation that converts multiplicative trajectory probabilities into additive step-wise credits. The discount factor γ^t applied in (6) is a practical regularization that emphasizes the early steps, where consensus is densest, and moderates the length penalty implied by the non-positive step rewards (Appendix E.3), and the Minimax polynomial estimator $\widetilde{c}^*_{X_t, K_t^{\mathrm{LOO}}}$ is a bounded finite-sample approximation of each $\beta \log \mathsf{q}_t^*(a_t \mid s_t)$ term (Theorem 15); the sum of the per-step lookups approximates the sum of the per-step logs, the natural step-wise form of the estimate. The factorization of the log-reward itself is exact. This reduces the credit assignment problem to local, step-level consensus, thereby dropping the *sample complexity* required for forming a statistically valid consensus from $\mathcal{O}(\eta^{-T})$ down to $\mathcal{O}(\eta^{-1})$ (as formalized in Theorem 19 in Appendix F.3), bypassing the combinatorial trajectory space $|\mathcal{A}|^T$. Gradient variance reduction follows naturally as a downstream statistical consequence of this tightened sample efficiency.

In practice, this factorization allows the coherence signal to be integrated directly into step-level advantage estimators (such as GAE). This solves the credit assignment problem by rewarding specific pivotal reasoning tokens rather than aggregating variance over the full sequence.

F.2 Finite-Sample Stability and Convergence

The practical instantiation of the Pairwise Coherence game relies on estimating the target distribution q^* from a finite group of size K. We now establish the statistical stability of this estimator and prove that the resulting dynamic reward landscape bounds reward over-optimization.

Bounded Over-Optimization. A primary failure mode of standard RLHF is reward hacking, where the policy discovers out-of-distribution traces that exploit the static reward model to achieve arbitrarily high scores ($r_\phi(y) \to \infty$). In the Coherence Game, the reward is dynamically derived from the group consensus: $R(y) = \beta \log \mathsf{q}^*(E(y))$. This structural shift provides a mathematical guarantee against unbounded over-optimization.

Below, we formally restate and prove Theorem 2 from the main text:

Theorem 2 (Intrinsic Bounding and Entropy Limits)**.** *For any target distribution $\mathsf{q}^* \in \Delta(\mathcal{Z})$ generated during the Pairwise Coherence game, the reward for any trace y is strictly bounded above: $R(y) \le 0$. Furthermore, if the policy π perfectly matches the target consensus, the expected reward evaluates exactly to the negative entropy of the consensus: $\mathbb{E}_{y\sim\pi^*}[R(y)] = -\beta H(\mathsf{q}^*)$.*

Proof. Since q^* is a valid probability distribution defined on the simplex $\Delta(\mathcal{Z})$, we have $\mathsf{q}^*(z) \le 1$ for all z. The monotonicity of the logarithm implies $\log \mathsf{q}^*(z) \le 0$, bounding the reward. When π matches the target such that the induced marginal $\nu_{\pi^*} = \mathsf{q}^*$, the expectation becomes $\mathbb{E}_{z\sim\mathsf{q}^*}[\beta \log \mathsf{q}^*(z)]$, which is the definition of negative scaled Shannon entropy $-\beta H(\mathsf{q}^*)$. □

Significance: This theorem proves that the policy cannot "hack" the reward to infinity. The maximum achievable reward is geometrically capped at 0 (achieved only if the group exhibits 100% unanimous deterministic consensus). As the policy converges, its expected reward is constrained by the inherent uncertainty (entropy) of the problem itself.

Finite-Sample Estimation and Bounded Gradient Variance. In PC-GRPO (Algorithm 3), the reward of a trace is a lookup $\widehat{R}^*(X) = \beta c_X^*$ in its leave-one-out count $X \in \{0, \dots, K\}$, where K in this paragraph denotes the number of peers (the leave-one-out group size). A coefficient vector $\mathbf{c} = (c_0, \dots, c_K)$ defines the estimator c_X of $\log p$ from $X \sim \mathrm{Bin}(K, p)$, whose expectation $P_{\mathbf{c}}(p) := \mathbb{E}_{X\sim\mathrm{Bin}(K,p)}[c_X] = \sum_{x=0}^{K} c_x \binom{K}{x} p^x (1-p)^{K-x}$ is a polynomial of degree K in p. The Minimax Polynomial estimator of Mohri et al. (2026b) chooses $\mathbf{c}$ to minimize the worst-case p-weighted bias $\sup_{p\in[0,1]} |pP_{\mathbf{c}}(p) - p\log p|$, a semi-infinite linear program. We solve this program under two additional shape constraints,

$$\epsilon_K := \min_{\mathbf{c}\in\mathbb{R}^{K+1}} \sup_{p\in[0,1]} |p\,P_{\mathbf{c}}(p) - p\log p| \quad \text{subject to} \quad c_0 \le c_1 \le \cdots \le c_K \le 0, \tag{13}$$

and denote its solution by $\mathbf{c}^*$. The unconstrained program attains the optimal rate $\Theta(1/K^2)$ (Appendix F.3), but its solution has coefficients of both signs whose magnitudes grow with K, so the corresponding reward is neither bounded below by its $X = 0$ value nor monotone in the count. The constraints in (13) enforce both properties, which our stability, convergence, generalization, and variance results use, at the price of the rate. Since the raw empirical logarithm diverges to $-\infty$ for unobserved events ($X = 0$), standard algorithms require ad-hoc heuristic smoothing with a floor tuned to K (Appendix I.9); the constrained lookup replaces this by a data-independent floor $c_0^* = -\Theta(\log K)$.

Below, we formally restate and prove Theorem 15 from the main text:

Theorem 15 (Finite-Sample Stability of PC-GRPO)**.** *Let the reward be evaluated using the constrained Minimax Polynomial estimator $\widehat{R}^*(X) = \beta c_X^*$ of (13), with $K \ge 2$ peers and $X \sim \mathrm{Bin}(K, \mathsf{q}^*(\mathsf{E}(y)))$. Then:*

1. (Structural boundedness and monotonicity.) *$\widehat{R}^*(X) \in [\beta c_0^*, 0]$ surely and $\widehat{R}^*$ is non-decreasing in X, where $c_0^* = -\Theta(\log K)$. The maximum uniform deviation across $\mathcal{Z}$ is therefore bounded deterministically: $\sup_{y\in\mathcal{Y}} \left|\widehat{R}^*(X_{\mathsf{E}(y)}) - \mathbb{E}[\widehat{R}^*(X_{\mathsf{E}(y)})]\right| \le \beta|c_0^*| = \mathcal{O}(\beta \log K)$. In contrast, the raw empirical estimator $\beta\log(X/K)$ diverges to $-\infty$ when $X = 0$, yielding unbounded deviation.*
2. (Polynomial-target unbiasedness.) *Define the* Minimax polynomial reward *$R^*(y) := \mathbb{E}_{X\sim Bin(K,\mathsf{q}^*(\mathsf{E}(y)))}[\widehat{R}^*(X)] = \beta P_{\mathbf{c}^*}(\mathsf{q}^*(\mathsf{E}(y)))$, the expectation of the estimator under the true consensus probability. Then the estimator is* exactly unbiased *for this target: $\mathbb{E}[\widehat{R}^*(X_{\mathsf{E}(y)})] = R^*(y)$ for all y. The polynomial reward R^* is bounded in $[\beta c_0^*, 0]$, is a polynomial (hence continuous) function of q^*, and approximates the ideal log-reward $\beta\log\mathsf{q}^*$ in the p-weighted uniform norm with error $\sup_{p\in[0,1]}|p\cdot P_{\mathbf{c}^*}(p) - p\log p| = \epsilon_K = \mathcal{O}(1/K)$.*

Proof. Part (1). The constraints of (13) give $c_0^* \le c_x^* \le 0$ and $c_x^* \le c_{x+1}^*$ for all x, so $\widehat{R}^*(X) = \beta c_X^* \in [\beta c_0^*, 0]$ is non-decreasing in X, and $\mathbb{E}[\widehat{R}^*(X)] \in [\beta c_0^*, 0]$ as the expectation of a bounded random variable; hence $|\widehat{R}^* - \mathbb{E}[\widehat{R}^*]| \le \beta|c_0^*|$ deterministically, and the supremum over $\mathcal{Z}$ inherits this bound. The order of c_0^* follows from Part (2) evaluated at $p = 1/K$, where $|P_{\mathbf{c}^*}(1/K) + \log K| \le K\epsilon_K$: monotonicity gives $c_0^* \le P_{\mathbf{c}^*}(1/K) \le -\log K + K\epsilon_K$, and $c_x^* \le 0$ gives $c_0^*\,\mathbb{P}[X = 0] \ge P_{\mathbf{c}^*}(1/K) \ge -\log K - K\epsilon_K$ with $\mathbb{P}[X = 0] = (1 - 1/K)^K \ge 1/4$; since $K\epsilon_K = \mathcal{O}(1)$, $c_0^* = -\Theta(\log K)$.

Part (2). Unbiasedness holds by the definition of R^* as $\mathbb{E}[\widehat{R}^*(X)]$: the per-step bias $b_t(y) = \mathbb{E}[\widehat{R}^*(X_{\mathsf{E}(y)})] - R^*(y)$ is identically zero for all y and all t, so the stochastic gradient introduces no mid-training error. Boundedness and continuity of R^* follow from Part (1) and from $P_{\mathbf{c}^*}$ being a polynomial. For the approximation error, ϵ_K is at most the weighted error of any feasible coefficient vector; we use the clipped plug-in $\widetilde{c}_x = \log\frac{x+1/2}{K+1/2}$, which is non-decreasing with $\widetilde{c}_K = 0$. Write $m = Kp$ and $Y = X + 1/2$, so that $\mathbb{E}[Y] = m + 1/2$. For $m \ge 1$, Jensen's inequality gives $\mathbb{E}[\log Y] \le \log(m+1/2)$, while the elementary bound $\log(1+u) \ge u - u^2/(2(1+\min\{u,0\}))$ for $u > -1$, applied to $u = (Y - \mathbb{E}\,Y)/\mathbb{E}\,Y$, gives $\mathbb{E}[\log Y] \ge \log(m + 1/2) - \mathbb{E}\big[(Y - \mathbb{E}\,Y)^2/(2\,\mathbb{E}\,Y\min\{Y, \mathbb{E}\,Y\})\big]$. On $\{X \ge m/2\}$ the ratio inside the expectation is at most $(Y - \mathbb{E}\,Y)^2/(\mathbb{E}\,Y)^2$, whose expectation is at most $\mathrm{Var}(X)/(\mathbb{E}\,Y)^2 \le 1/m$; on $\{X < m/2\}$ it is at most $\mathbb{E}\,Y = m + 1/2$, and this event has probability at most $e^{-m/8}$ (Chernoff). Hence $\mathbb{E}[\log Y] \ge \log(m + 1/2) - C/m$ for an absolute constant C. Since $0 \le \log\frac{m+1/2}{(K+1/2)p} \le \frac{1}{2m}$, we obtain $|P_{\widetilde{\mathbf{c}}}(p) - \log p| \le C/m$ (taking $C \ge 1/2$), i.e., a weighted error of at most C/K for every $p \ge 1/K$. For $p < 1/K$, monotonicity and Jensen's inequality

give $-\log(2K+1) \le P_{\tilde{\mathbf{c}}}(p) \le \log \frac{Kp+1/2}{K+1/2}$, and both $p\,|\log(p(2K+1))|$ and $p\log(1+1/(2Kp))$ are $\mathcal{O}(1/K)$ uniformly in $p < 1/K$. Hence $\epsilon_K \le \sup_p |pP_{\tilde{\mathbf{c}}}(p) - p\log p| = \mathcal{O}(1/K)$. (The unconstrained program of Mohri et al. (2026b) attains the rate $\Theta(1/K^2)$; numerically, the value of (13) scales as $1/K$, see Appendix F.3.) □

Convergence of PC-GRPO via Stochastic Mirror Descent. To complete our finite-sample analysis, we provide a formal convergence guarantee for PC-GRPO. We caution that the KL-regularized objective is *not* L-smooth in the Euclidean sense: the Hessian of the KL term diverges near the boundary of the simplex, so a Euclidean descent-lemma analysis is inapplicable. Instead, we analyze the update as Stochastic Mirror Descent (SMD) with the negative-entropy mirror map (Beck & Teboulle, 2003), which requires only Lipschitz continuity of the unregularized (expected-reward) part with respect to the L_1 norm, and we state the conclusion as a suboptimality guarantee on the averaged iterate.

Below, we formally restate and prove Theorem 16 from the main text:

Theorem 16 (Convergence of PC-GRPO)**.** *Let $\mathcal{D}$ be the prompt distribution and fix a target consensus q^*. For each prompt x, let $\mathcal{Z}$ be the finite set of execution outcomes, $\nu_0(\cdot \mid x)$ the outcome marginal of $\pi_0(\cdot \mid x)$, and let Π be the class of* mode-level *policies, which reweight the outcome classes of π_0 while keeping its conditional within each class: $\pi(y \mid x) = \nu(\mathsf{E}(y) \mid x)\,\pi_0(y \mid \mathsf{E}(y), x)$ with $\nu(\cdot \mid x) \in \Delta(\mathcal{Z})$. Let $\mathcal{L}(\pi) = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_{y\sim\pi}[R^*(y \mid x)] - \beta \mathsf{D}_{\mathrm{KL}}(\pi \,\|\, \pi_0)$, where $R^*(y \mid x) = \beta P_{\mathbf{c}^*}(\mathsf{q}^*(\mathsf{E}(y \mid x)))$ is the Minimax polynomial reward from Theorem 15, bounded in $[\beta c_0^*, 0]$ with $c_0^* = -\Theta(\log K)$, and let $\mathcal{L}^* = \max_{\pi\in\Pi} \mathcal{L}(\pi)$. Assume that at each step t and for each prompt x the update receives a stochastic gradient $\widehat{g}_t(\cdot \mid x) \in \mathbb{R}^{\mathcal{Z}}$ with $\mathbb{E}[\widehat{g}_t(\cdot \mid x) \mid \mathcal{F}_{t-1}] = (R^*(z \mid x))_{z\in\mathcal{Z}}$ and $\|\widehat{g}_t(\cdot \mid x)\|_\infty \le G := \beta|c_0^*|$ almost surely (for instance, the lookup $\widehat{R}^*$ evaluated at an independent $\mathrm{Bin}(K, \mathsf{q}^*(z \mid x))$ count for every $z \in \mathcal{Z}$; the implemented estimator evaluates the lookup only at the outcomes sampled from π_t). After T composite Stochastic Mirror Descent steps on $\nu(\cdot \mid x)$ with the negative-entropy mirror map and the tuned learning rate $\eta = \sqrt{2D_{\max}}/(G\sqrt{T})$, where $D_{\max} > 0$ is defined below (if $D_{\max} = 0$, the reference policy is optimal for every prompt and is returned without updating), the average policy $\overline{\pi}_T = \frac{1}{T}\sum_{t=1}^{T} \pi_t$ satisfies:*

$$\mathcal{L}^* - \mathbb{E}[\mathcal{L}(\overline{\pi}_T)] \le \mathcal{O}\left(\beta \log K \cdot \sqrt{\frac{D_{\max}}{T}}\right),$$

where $D_{\max} = \max_{x\in\mathcal{D}} \mathsf{D}_{\mathrm{KL}}(\nu^*(\cdot|x) \,\|\, \nu_0(\cdot|x))$ is the maximum KL divergence between the optimal and initial marginal distributions over the unique execution outcomes $\mathcal{Z}$. Moreover, if $\mathsf{q}^*(\cdot \mid x) \ge q_{\min} > 0$ on $\mathcal{Z}$ uniformly over x (at the equilibrium of the KL-regularized game, $\mathsf{q}^* = \nu_{\pi^*}$ lies within a factor $e^{|c_0^*|}$ of ν_0, so $q_{\min} = e^{-|c_0^*|} \inf_{x,z} \nu_0(z \mid x)$ works whenever the reference outcome probabilities are bounded below uniformly over prompts, e.g., for a finite prompt set), the gap between $\mathcal{L}^*$ and the ideal log-reward game optimum $\mathcal{L}^*_{\log} = \max_{\pi\in\Pi} \mathcal{L}_{\log}(\pi)$, with $\mathcal{L}_{\log}$ defined with $\beta \log \mathsf{q}^*(\mathsf{E}(y \mid x))$ in place of R^*, satisfies $|\mathcal{L}^* - \mathcal{L}^*_{\log}| \le \beta\epsilon_K/q_{\min} = \mathcal{O}(\beta/(q_{\min}K))$, where $\epsilon_K = \mathcal{O}(1/K)$ is the weighted approximation error of (13); for a policy whose outcome marginal equals q^*, the objectives differ by at most $\beta s\,\epsilon_K$, with s the support size of q^*.

Proof. Write $\mathcal{L}(\pi) = f(\pi) - \beta \mathsf{D}_{\mathrm{KL}}(\pi \,\|\, \pi_0)$ with $f(\pi) = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_{y\sim\pi}[R^*(y \mid x)]$, and consider the equivalent minimization of the loss $-\mathcal{L}$ over Π.

Reduction to outcome marginals. Since the polynomial reward $R^*(y \mid x) = \beta P_{\mathbf{c}^*}(\mathsf{q}^*(\mathsf{E}(y) \mid x))$ depends on the trajectory y only through its execution outcome $z = \mathsf{E}(y) \in \mathcal{Z}$, for $\pi \in \Pi$ we have $\mathbb{E}_{y\sim\pi(\cdot|x)}[R^*(y \mid x)] = \sum_z \nu(z \mid x) R^*(z \mid x)$. The chain rule for relative entropy gives $\mathsf{D}_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x)) = \mathsf{D}_{\mathrm{KL}}(\nu(\cdot \mid x) \,\|\, \nu_0(\cdot \mid x)) + \mathbb{E}_{z\sim\nu(\cdot|x)}[\mathsf{D}_{\mathrm{KL}}(\pi(\cdot \mid z, x) \,\|\, \pi_0(\cdot \mid z, x))]$, and the second term vanishes identically on Π because every policy in Π keeps the reference conditional $\pi_0(\cdot \mid z, x)$ within each outcome class. Hence $\mathcal{L}(\pi) = \mathbb{E}_{x\sim\mathcal{D}}[\mathcal{L}_x(\nu(\cdot \mid x))]$ with $\mathcal{L}_x(\nu) = \sum_z \nu(z) R^*(z \mid x) - \beta \mathsf{D}_{\mathrm{KL}}(\nu \,\|\, \nu_0(\cdot \mid x))$, and maximizing $\mathcal{L}$ over Π is the same as maximizing $\mathcal{L}_x$ over $\nu \in \Delta(\mathcal{Z})$ for every prompt. The composite mirror step below is a multiplicative reweighting of $\nu(\cdot \mid x)$, so the iterates remain in Π, and the maximum Bregman distance from initialization is bounded by $D_{\max} = \max_x \mathsf{D}_{\mathrm{KL}}(\nu^*(\cdot|x) \,\|\, \nu_0(\cdot|x))$.

Convexity. Since we analyze convergence for a *fixed* target q^*, the expected reward term $\mathcal{L}_x$ is linear in ν (the reward $R^*(z \mid x)$ does not depend on π for fixed q^*). The Pairwise Coherence equilibrium

(Mohri et al., 2026a, Theorem 1) guarantees that the optimal π^* and q^* form a fixed point, so the SMD analysis characterizes convergence to this equilibrium solution. Since $\mathsf{D}_{\mathrm{KL}}(\nu \,\|\, \nu_0)$ is convex in ν, $\mathcal{L}_x$ is concave and $-\mathcal{L}_x$ is convex over the simplex; the maximizer exists.

Lipschitz continuity in the right geometry. Both terms of $\mathcal{L}$ decompose additively over prompts, so we run the SMD update, and state the regret bound, *per prompt*: for each x, the gradient of $f_x(\nu) = \sum_z \nu(z) R^*(z \mid x)$ has entries $\nabla f_x[z] = R^*(z \mid x)$, with no dependence on $\mathcal{D}(x)$. (Differentiating the prompt-averaged objective with respect to $\nu(z \mid x)$ would instead produce the factor $\mathcal{D}(x)\, R^*(z \mid x)$, which is why the analysis is carried out per prompt and averaged over $\mathcal{D}$ only at the end.) Since the constrained Minimax rewards are bounded in $[\beta c_0^*, 0]$, the gradient magnitude is bounded by $\|\nabla f_x\|_\infty \le |\beta c_0^*| = G$, where $G = \Theta(\beta \log K)$ by Theorem 15. Hence $-f_x$ is G-Lipschitz with respect to the L_1 norm, whose dual is the L_∞ norm. The KL regularizer is *not* Lipschitz near the boundary, but it need not be: it is handled exactly by the composite (proximal) mirror step, which incorporates $\beta \mathsf{D}_{\mathrm{KL}}(\cdot \,\|\, \nu_0)$ into the Bregman projection in closed form (a multiplicative-weights update), so the composite update only linearizes the unregularized expected-reward term f_x.

SMD regret bound. By assumption, the stochastic gradient $\widehat{g}_t(\cdot \mid x)$ of f_x at step t is unbiased, $\mathbb{E}[\widehat{g}_t(\cdot \mid x) \mid \mathcal{F}_{t-1}] = \nabla f_x$, and bounded, $\|\widehat{g}_t\|_\infty \le G$ almost surely; the oracle in the statement satisfies both conditions because $R^*(z \mid x) = \mathbb{E}[\widehat{R}^*(X)]$ for $X \sim \mathrm{Bin}(K, \mathsf{q}^*(z \mid x))$ (Theorem 15, Part 2) and $\widehat{R}^* \in [\beta c_0^*, 0]$. (For the fixed-target analysis, the counts are drawn under the static target q^*; the non-stationarity introduced when the target evolves with π_t is handled separately by Corollary 17.) The standard composite SMD bound with the negative-entropy mirror map over the simplex (Beck & Teboulle, 2003; Nemirovski et al., 2009) gives, for each prompt x, for the averaged iterate $\overline{\nu}_T = \frac{1}{T}\sum_{t=1}^T \nu_t$ and step size $\eta = \sqrt{2D_{\max}}/(G\sqrt{T})$ (common to all prompts), a per-prompt suboptimality of $\mathcal{O}(G\sqrt{D_{\max}/T})$, since the per-prompt Bregman radius $D_x = \mathsf{D}_{\mathrm{KL}}(\nu^*(\cdot \mid x) \,\|\, \nu_0(\cdot \mid x))$ satisfies $D_x \le D_{\max}$. The averaged policy $\overline{\pi}_T$ is the mode-level policy with marginal $\overline{\nu}_T$. Taking the expectation over $x \sim \mathcal{D}$ and using $G = \Theta(\beta \log K)$ yields:

$$\mathcal{L}^* - \mathbb{E}[\mathcal{L}(\overline{\pi}_T)] \le \mathcal{O}\left(\beta \log K \cdot \sqrt{\frac{D_{\max}}{T}}\right).$$

Gap to the log-reward optimum. For $\pi \in \Pi$ with outcome marginal ν, the two objectives differ only in the reward term, and $|\mathcal{L}(\pi) - \mathcal{L}_{\log}(\pi)| \le \beta \sum_z \nu(z \mid x)\, |P_{\mathbf{c}^*}(\mathsf{q}^*(z \mid x)) - \log \mathsf{q}^*(z \mid x)| \le \beta \sum_z \nu(z \mid x)\, \epsilon_K / \mathsf{q}^*(z \mid x) \le \beta \epsilon_K / q_{\min}$ by Theorem 15, Part 2 (for every prompt, hence in expectation over $\mathcal{D}$). Since the bound holds uniformly over Π, the two maxima differ by at most $\beta \epsilon_K / q_{\min} = \mathcal{O}(\beta/(q_{\min} K))$. When $\nu = \mathsf{q}^*$, the middle sum equals $s\, \epsilon_K$ for a support of size s. □

Significance: Since the objective is non-smooth in the Euclidean sense, the guarantee is a suboptimality bound on the averaged iterate. Because the estimator is exactly unbiased for the polynomial target, there is zero estimation-bias floor in optimizing $\mathcal{L}(\pi)$. The optimization error is reduced purely by more training steps T. The Minimax approximation gap, $\mathcal{O}(\beta/(q_{\min} K))$, exists exclusively between the converged optimum $\mathcal{L}^*$ and the ideal log-reward optimum $\mathcal{L}^*_{\log}$, explicitly decoupling the optimization convergence from the finite-sample statistical limit. The guarantee is stated for mode-level policies and an idealized gradient oracle; the implemented GRPO update is a parametric stochastic-gradient approximation of this scheme (Appendix A).

Corollary 17 (Non-Stationary Target Extension)**.** *Suppose the target distribution q_t^* is updated at each step (reflecting the evolving consensus as π changes). Here q_t^* denotes the* population *(expected) target induced by π_t, not its empirical K-sample estimate: the $\mathcal{O}(1/\sqrt{K})$ sampling noise of the empirical consensus does not enter the drift below (it would otherwise accumulate to $\mathcal{O}(T/\sqrt{K})$ over T steps), but is absorbed by the stochastic gradient, which is exactly unbiased for the polynomial target (Theorem 16). Let $R_t^*(z) = \beta P_{\mathbf{c}^*}(\mathsf{q}_t^*(z))$ be the population reward induced by q_t^* and let $V_T = \sum_{t=1}^{T-1} \|R_{t+1}^* - R_t^*\|_\infty$ denote its total drift, measured in reward space. Since π_t is random, so are q_t^*, the per-step optima, and V_T, and the bound below is on the expected dynamic regret; $D_{\max}$ denotes a deterministic bound on $\max_x \mathsf{D}_{\mathrm{KL}}(\nu_t^*(\cdot \mid x) \,\|\, \nu_0(\cdot \mid x))$ valid for all t (for instance $|c_0^*|$, since every per-step optimum lies within a factor $e^{|c_0^*|}$ of ν_0). (In terms of the target, $\|R_{t+1}^* - R_t^*\|_\infty \le \beta L_P \|\mathsf{q}_{t+1}^* - \mathsf{q}_t^*\|_\infty$, where $L_P = \max_{p \in [0,1]} |P'_{\mathbf{c}^*}(p)| = \Omega(K)$; see the proof.) Then under the same SMD scheme (mode-level policy class and gradient oracle of Theorem 16), standard*

dynamic regret bounds for Online Mirror Descent yield:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\Big(\max_{\pi}\mathcal{L}_t(\pi)-\mathcal{L}_t(\pi_t)\Big)\right]\le\mathcal{O}\left(\beta\log K\cdot\sqrt{\frac{D_{\max}}{T}}+\frac{\log K\cdot|\log\pi_{\min}|}{\sqrt{D_{\max}}}\cdot\frac{\mathbb{E}[V_T]}{\sqrt{T}}\right),$$

where the maximum is over the mode-level class, the expectation is over the stochastic gradients (hence over the trajectory of π_t, q_t^* and the per-step optima), and $\pi_{\min}>0$ lower-bounds the outcome probabilities of the iterates and of the per-step optima; the composite step and the Gibbs form of the optima keep both within a factor $e^{2G/\beta}=K^{\mathcal{O}(1)}$ of ν_0, so $|\log\pi_{\min}|=\mathcal{O}(|\log\min_z\nu_0(z)|+\log K)$. The constants depend on K only polylogarithmically. In particular:

1. (Convergent game.) *When (π^*,q^*) is the game equilibrium and $\mathsf{q}_t^*\to\mathsf{q}^*$ at geometric rate $\|\mathsf{q}_t^*-\mathsf{q}^*\|_\infty\le C_0\rho^t$ ($\rho<1$) along every trajectory, $V_T\le 2\beta L_P C_0/(1-\rho)=\mathcal{O}(1)$ deterministically for fixed K, and the non-stationary penalty $\mathcal{O}(1/\sqrt{T})$ vanishes asymptotically at the same rate as the standard optimization error.*
2. (Two-timescale schedule.) *When q^* is updated every B steps, the policy takes B steps of size $\eta=\Theta(1/\sqrt{T})$ (the tuned rate of Theorem 16), bounding the per-update reward drift by $\delta\le\beta L_P L_q B\eta G=\mathcal{O}(\beta L_P L_q G B/\sqrt{T})$, where L_q is the Lipschitz constant of the target update map. The total drift is $V_T=(T/B)\delta=\mathcal{O}(\beta L_P L_q G\sqrt{T})$ along every trajectory, so the drift penalty is a bounded constant, $\mathcal{O}(\beta L_P L_q G\log K\,|\log\pi_{\min}|/\sqrt{D_{\max}})$, guaranteeing stable tracking of the dynamic optimum.*

Proof. The decomposition follows from standard dynamic regret bounds for Online Mirror Descent (Zinkevich, 2003) with the negative-entropy mirror map, applied pathwise to the linearized losses $\nu\mapsto\langle\widehat{g}_t,\nu\rangle$ with the composite KL term and the (random, adapted) comparator sequence ν_t^* of per-step optima: the Bregman cross-term $\mathsf{D}_{\mathrm{KL}}(\nu_{t+1}^*\,\|\,\nu_{t+1})-\mathsf{D}_{\mathrm{KL}}(\nu_t^*\,\|\,\nu_{t+1})$ is at most $\sum_z(\nu_{t+1}^*-\nu_t^*)(z)\log\frac{\nu_{t+1}^*(z)}{\nu_{t+1}(z)}\le 2|\log\pi_{\min}|\,\|\nu_{t+1}^*-\nu_t^*\|_1$ by convexity of the KL divergence in its first argument, so with the deterministic step size $\eta=\sqrt{2D_{\max}}/(G\sqrt{T})$ the averaged linearized dynamic regret is, for every realization, $\mathcal{O}\big(G\sqrt{D_{\max}/T}\big)+\mathcal{O}\big(G\,|\log\pi_{\min}|\sum_t\|\nu_{t+1}^*-\nu_t^*\|_1/\sqrt{D_{\max}T}\big)$. To pass to the objectives, note that π_t, hence q_t^*, R_t^* and ν_t^*, are determined before $\widehat{g}_t$ is drawn, so ν_t and ν_t^* are $\mathcal{F}_{t-1}$-measurable and $\mathbb{E}[\langle\widehat{g}_t,\nu_t-\nu_t^*\rangle\mid\mathcal{F}_{t-1}]=\langle R_t^*,\nu_t-\nu_t^*\rangle=f_t(\nu_t)-f_t(\nu_t^*)$, the reward term of $\mathcal{L}_t$ being linear in ν for the fixed target q_t^*; the KL term is handled exactly by the composite step. Taking expectations of the pathwise bound and using the tower property therefore bounds $\mathbb{E}[\frac{1}{T}\sum_t(\max_\pi\mathcal{L}_t(\pi)-\mathcal{L}_t(\pi_t))]$ by the same expression with $\mathbb{E}[\sum_t\|\nu_{t+1}^*-\nu_t^*\|_1]$ in place of the path length. The floor $\pi_{\min}$ exists for the optima because $\nu_t^*(z)=\nu_0(z)e^{R_t^*(z)/\beta}/Z_t$ with $R_t^*/\beta\in[c_0^*,0]$, and for the iterates because the composite step is $\log\nu_{t+1}(z)=\frac{\log\nu_t(z)+\eta\widehat{g}_t(z)}{1+\eta\beta}+\frac{\eta\beta}{1+\eta\beta}\log\nu_0(z)-\log Z_{t+1}$; unrolling from $\nu_1=\nu_0$ and using $|\widehat{g}_t(z)|\le G$ bounds the accumulated gradient term by $\eta G\sum_{k\ge1}(1+\eta\beta)^{-k}=G/\beta$, so $\nu_t(z)\in[\nu_0(z)e^{-2G/\beta},\nu_0(z)e^{2G/\beta}]$ and $|\log\pi_{\min}|\le|\log\min_z\nu_0(z)|+2G/\beta$ with $G/\beta=|c_0^*|=\Theta(\log K)$.

For the expected Minimax polynomial objective over the mode-level class, the exact optimal policy for a fixed target q_t^* is $\nu_t^*(z)=\nu_0(z)\exp(R_t^*(z)/\beta)/Z_t$ with normalizer $Z_t=\sum_z\nu_0(z)\exp(R_t^*(z)/\beta)$. If $\delta_t:=\|R_{t+1}^*-R_t^*\|_\infty$, every unnormalized weight changes by a factor in $[e^{-\delta_t/\beta},e^{\delta_t/\beta}]$, hence so does Z_t, and $\nu_{t+1}^*(z)/\nu_t^*(z)\in[e^{-2\delta_t/\beta},e^{2\delta_t/\beta}]$ for every z. Therefore

$$\|\nu_{t+1}^*-\nu_t^*\|_1=\sum_z\nu_t^*(z)\left|\frac{\nu_{t+1}^*(z)}{\nu_t^*(z)}-1\right|\le\min\big\{2,\;e^{2\delta_t/\beta}-1\big\}\le\frac{2e^2}{\beta}\,\delta_t$$

for every $\delta_t\ge0$ (for $\delta_t\le\beta$ use $e^u-1\le ue^u$ with $u=2\delta_t/\beta\le2$; for $\delta_t>\beta$ the right-hand side exceeds 2), and hence $\sum_t\|\nu_{t+1}^*-\nu_t^*\|_1\le 2e^2V_T/\beta$ pathwise and $\mathbb{E}[\sum_t\|\nu_{t+1}^*-\nu_t^*\|_1]\le 2e^2\,\mathbb{E}[V_T]/\beta$. Substituting into the regret bound and using $G/\beta=\Theta(\log K)$ gives the drift term $\mathcal{O}(\log K\cdot|\log\pi_{\min}|\cdot V_T/\sqrt{D_{\max}T})$.

In terms of the target drift, the factor $L_P=\max_p|P_{\mathbf{c}^*}'(p)|$ appears, and $L_P=\Omega(K)$: with $\epsilon_K\le C/K$ (Theorem 15), fix $a:=\max\{1,8C\}$; for $K\ge2a$ the weighted error bound $|P_{\mathbf{c}^*}(p)-\log p|\le\epsilon_K/p$ at $p=a/K$ and $p=2a/K$ gives $P_{\mathbf{c}^*}(2a/K)-P_{\mathbf{c}^*}(a/K)\ge\log2-\frac{C}{2a}-\frac{C}{a}\ge\log2-\frac{3}{16}>0$, so by

the mean value theorem on an interval of length a/K, $L_P \geq \frac{K}{a}\left(\log 2 - \frac{3}{16}\right) = \Omega(K)$ (numerically $K\epsilon_K \leq 0.09$, Appendix F.3, so $a = 1$ suffices). For Part (1), $\delta_t \leq \beta L_P(C_0\rho^t + C_0\rho^{t+1}) \leq 2\beta L_P C_0 \rho^t$ sums to $\mathcal{O}(1)$; for Part (2), the per-update target drift is at most $L_q B\eta G$, hence the stated reward drift; the drift penalty is bounded under both schedules. □

Generalization of Coherence to Unseen Prompts. We now prove that the coherence reward learned during training generalizes to unseen prompts. The key insight is that the Minimax reward's intrinsic bounding (Theorem 2) provides a natural range parameter for Rademacher-based uniform convergence.

Theorem 18 (Generalization of Coherence to Unseen Prompts)**.** *Let $\hat{\mathcal{L}}_n(\pi)$ be the empirical coherence reward evaluated over n i.i.d. training prompts using Minimax rewards bounded in $[\beta c_0^*, 0]$, with the population target $\mathsf{q}^*(\cdot \mid x)$ of Theorem 16 held fixed. Let $\mathcal{F} = \{x \mapsto \mathbb{E}_{y\sim\pi}[R^*(y \mid x)] \mid \pi \in \Pi\}$ be the induced loss class for a policy class Π, which is specified independently of the n training prompts. With probability at least $1-\delta$, for any $\pi \in \Pi$: $\hat{\mathcal{L}}_n(\pi) - \mathcal{L}(\pi) \leq 2\mathfrak{R}_n(\mathcal{F}) + |\beta c_0^*|\sqrt{\frac{\log(1/\delta)}{2n}}$, where $\mathcal{L}$ here denotes the population coherence reward (without the KL regularizer).*

Proof. The coherence reward for a single prompt x is $\ell(\pi, x) = \mathbb{E}_{y\sim\pi}[R^*(y \mid x)]$. By Theorems 2 and 15, the Minimax reward is strictly bounded: $R^*(y) \in [\beta c_0^*, 0]$, where $c_0^* = -\Theta(\log K) \approx -\log K$. Hence the per-prompt reward is bounded within an absolute range of $c = |\beta c_0^*|$.

Let $\hat{\mathcal{L}}_n(\pi) = \frac{1}{n}\sum_{i=1}^n \ell(\pi, x_i)$ and $\mathcal{L}(\pi) = \mathbb{E}_{x\sim\mathcal{D}}[\ell(\pi, x)]$. By the standard Rademacher complexity bound for bounded function classes (Mohri et al., 2018, Theorem 3.1), for any $\pi \in \Pi$ with probability at least $1-\delta$:

$$\hat{\mathcal{L}}_n(\pi) - \mathcal{L}(\pi) \leq 2\mathfrak{R}_n(\mathcal{F}) + |\beta c_0^*|\sqrt{\frac{\log(1/\delta)}{2n}}.$$

The first term captures the complexity of the induced loss class $\mathcal{F}$. By the scaling property of Rademacher complexity, $\mathfrak{R}_n(\mathcal{F}) = |\beta c_0^*| \cdot \mathfrak{R}_n(\overline{\mathcal{F}})$, where $\overline{\mathcal{F}}$ is the normalized class mapping to $[0, 1]$ (whose complexity depends on the policy parameterization, e.g., $\mathfrak{R}_n(\overline{\mathcal{F}}) = \mathcal{O}(\sqrt{d/n})$ for rank-constrained networks with d trainable parameters). The second term is the standard McDiarmid concentration penalty for the bounded range $|\beta c_0^*|$. Critically, the consensus-specific range factor $|\beta c_0^*| \approx \beta \log K$ grows only logarithmically in K, ensuring the group-size scaling does not degrade the statistical guarantee. □

Advantage Variance Reduction. We now formalize the core mechanism by which PC-GRPO overcomes the zero-gradient limitation of standard GRPO.

Theorem 3 (Advantage Variance Separation)**.** *Let $\hat{A}_{GRPO}(y)$ and $\hat{A}_{PC}(y)$ denote the advantage estimators of Vanilla GRPO (binary pass/fail) and* PC-GRPO *(Minimax coherence reward), respectively, evaluated on a group of size K. Let η = pass@1 be the per-trace success probability and* Var *the empirical sample variance (denominator K) of the mean-centered, unstandardized advantages within the group. Then: (1) For K traces i.i.d. from π on one prompt, $\mathbb{E}[\mathrm{Var}[\hat{A}_{GRPO}]] = \frac{K-1}{K}\eta(1-\eta)$. As $\eta \to 0$ the variance vanishes, and with the standard-deviation normalization used in practice the cold-start advantage is the indeterminate $0/0$ (masked to 0): zero gradient signal. (2)* (Count separation.) *Suppose a fraction at least $\alpha > 0$ of the filtered group forms a dominant mode (identical outputs on every evaluated test, each a plurality on its test) and some program's output on one test has a leave-one-out count at most $1/c$ times the dominant output's, for constants $\alpha > 0$, $c > 1$. Then for $K \geq K_0(c, \alpha)$, with the number of tests N and β fixed, the test-averaged Minimax reward gap between the dominant mode and that program is at least $(\beta/2N)\log c = \Omega(1)$ and $\mathrm{Var}[\hat{A}_{PC}] = \Omega(1/K)$, even when $\eta = 0$ (*Var*: sample variance within one such group).*

Proof. **Part (1): GRPO advantage variance.** In Vanilla GRPO with binary execution rewards, each trace y_i in a group of size K receives reward $r_i = \mathbb{I}[y_i \text{ passes all tests}]$. The traces are i.i.d. draws from π, so $r_i \sim \text{Bernoulli}(\eta)$ where $\eta = \text{pass@1}$.

The unnormalized GRPO advantages are $\hat{A}(y_i) = r_i - \bar{r}$, where $\bar{r} = \frac{1}{K}\sum_j r_j$. The expected empirical sample variance of these advantages across the group is:

$$\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}(r_i-\bar{r})^2\right] = \mathbb{E}[r_i^2] - \mathbb{E}[\bar{r}^2] = \eta - \left(\frac{\eta}{K} + \frac{K-1}{K}\eta^2\right) = \frac{K-1}{K}\eta(1-\eta),$$

matching the factor stated in the theorem. (If the group mixes independent traces with different success probabilities η_i, e.g., $K/2$ samples each for a prompt and its orbit transform, the same computation gives $\frac{K-1}{K}\bar{\eta}(1-\bar{\eta}) + \frac{1}{K}\mathrm{Var}_i(\eta_i)$ with $\bar{\eta}$ the mean and Var_i the population variance of the η_i, which also vanishes as all $\eta_i \to 0$.) As $\eta \to 0$, this expected variance approaches 0. Moreover, the probability that *any* $r_i = 1$ is $1-(1-\eta)^K \approx K\eta \to 0$, so with high probability all advantages within a sampled group are identically zero, rendering the gradient update null. Practical GRPO implementations additionally divide by the group standard deviation, $\hat{A}(y_i) = (r_i - \bar{r})/\sigma_r$; in the cold-start regime $\sigma_r = 0$ and the normalized advantage is the indeterminate $0/0$, strictly enforcing the zero-gradient conclusion.

Part (2): PC-GRPO advantage variance under count separation. In PC-GRPO (Step 5 of Algorithm 1; Step 6 of Algorithm 3), the reward of a program $y \in G'$ averages, over the N evaluated tests, a lookup in its leave-one-out agreement count on each test:

$$R(y) = \frac{\beta}{N}\sum_{j=1}^{N} c^*_{X_j(y)}, \qquad X_j(y) = \sum_{y' \in G' \setminus \{y\}} \mathbb{I}[y'(u_j) = y(u_j)] \in \{0,\dots,M\},$$

with $M := K-1$ peers and $\mathbf{c}^* = \mathbf{c}^*(M)$ the constrained lookup of (13). The advantage is $\hat{A}_{PC}(y) = R(y) - \bar{R}$.

Let $S \subseteq G'$ be the dominant mode: $|S| \geq \alpha K$, all members of S produce the same output v_j on every test u_j, and v_j is a plurality output on u_j. Let $n_j \geq |S| \geq \alpha K$ be the number of programs in G' outputting v_j on u_j, so that every $y_S \in S$ has $X_j(y_S) = n_j - 1$. Let $y' \in G' \setminus S$ be a program whose output on some test u_{j^*} satisfies $X_{j^*}(y') \leq X_{j^*}(y_S)/c$. On every test j, $X_j(y') \leq X_j(y_S)$: if $y'(u_j) = v_j$ the two counts coincide, and otherwise the count of y''s output is at most n_j because v_j is a plurality. Since the lookup is monotone (Theorem 15), $c^*_{X_j(y')} \leq c^*_{X_j(y_S)}$ for every j, and hence

$$R(y_S) - R(y') \geq \frac{\beta}{N}(c^*_a - c^*_b), \qquad a := X_{j^*}(y_S) = n_{j^*} - 1 \geq \alpha M - 1, \quad b := X_{j^*}(y') \leq a/c.$$

It remains to bound the coefficient gap $c^*_a - c^*_b$. The coefficients are known only through their Binomial averages $P_{\mathbf{c}^*}(p) = \mathbb{E}_{X\sim\mathrm{Bin}(M,p)}[c^*_X]$, which satisfy $|P_{\mathbf{c}^*}(p) - \log p| \leq \epsilon_M/p$ with $\epsilon_M = \mathcal{O}(1/M)$ (Theorem 15); we pass from averages to individual coefficients by concentration. Fix $t > 0$ with $e^{-2t^2} \leq 1/2$.

Lower bound on c^*_a. Let $p_- = (a - t\sqrt{M})/M$ (assumed positive) and $X \sim \mathrm{Bin}(M, p_-)$. Since $c^*_x \leq c^*_a$ for $x \leq a$ and $c^*_x \leq 0$ for $x > a$, $P_{\mathbf{c}^*}(p_-) \leq c^*_a\,\mathbb{P}[X \leq a]$, while Hoeffding's inequality gives $\mathbb{P}[X > a] = \mathbb{P}[X - Mp_- > t\sqrt{M}] \leq e^{-2t^2}$. As $P_{\mathbf{c}^*}(p_-) \geq \log p_- - \epsilon_M/p_-$ is negative, $c^*_a \geq (\log p_- - \epsilon_M/p_-)/(1-e^{-2t^2}) \geq (1+2e^{-2t^2})(\log p_- - \epsilon_M/p_-)$.

Upper bound on c^*_b. Let $p_+ = (b + t\sqrt{M})/M$ (assumed at most 1) and $X \sim \mathrm{Bin}(M, p_+)$. Since $c^*_x \geq c^*_b$ for $x \geq b$ and $c^*_x \geq c^*_0$ always, $P_{\mathbf{c}^*}(p_+) \geq c^*_b\,\mathbb{P}[X \geq b] + c^*_0\,\mathbb{P}[X < b]$, and $\mathbb{P}[X < b] \leq e^{-2t^2}$. Hence $c^*_b\,\mathbb{P}[X \geq b] \leq N_b := \log p_+ + \epsilon_M/p_+ + |c^*_0|e^{-2t^2}$, and $c^*_b \leq \min\{0, N_b\}$ (if $N_b < 0$, dividing by $\mathbb{P}[X \geq b] \leq 1$ only decreases the bound).

Gap. Take $t = \sqrt{\log M}$, so $e^{-2t^2} = M^{-2}$. Then $p_- \geq \alpha - (1 + \sqrt{M\log M})/M \geq \alpha/2$ and $p_+ \leq 1/c + \sqrt{(\log M)/M} \leq (1+1/c)/2 < 1$ for M large enough (depending on α and c). The error terms $\epsilon_M/p_- \leq 2\epsilon_M/\alpha$, $\epsilon_M/p_+ \leq \epsilon_M\sqrt{M/\log M}$, and $|c^*_0|e^{-2t^2} = \mathcal{O}((\log M)/M^2)$ all vanish as $M \to \infty$, so $N_b < 0$ and

$$c^*_a - c^*_b \geq \log\frac{p_-}{p_+} - o(1) \geq \log\frac{a - \sqrt{M\log M}}{a/c + \sqrt{M\log M}} - o(1) \longrightarrow \log c,$$

using $a \geq \alpha M - 1 \gg \sqrt{M\log M}$. Consequently there is $K_0(c,\alpha)$ such that for all $K \geq K_0(c,\alpha)$, $c^*_a - c^*_b \geq \frac{1}{2}\log c$ and the test-averaged reward gap satisfies

$$R(y_S) - R(y') \geq \frac{\beta}{2N}\log c = \Omega(1),$$

where N is the number of evaluated tests; if the separation by the factor c holds on every test, the factor $1/N$ disappears. The factor $1/N$ is unavoidable in general: if $K-1$ candidates behave identically and one candidate differs on a single test, the averaged gap is at most $\beta|c_0^*|/N = \mathcal{O}(\beta \log K/N)$ and the variance is $\mathcal{O}(\beta^2 \log^2 K/(KN^2))$. The case in which y''s output on u_{j^*} is unique ($b=0$) is included: there $p_+ = \sqrt{(\log M)/M}$, so $c_b^* \le -\frac{1}{2}\log(M/\log M) + o(1)$ and the coefficient gap is even $\Omega(\log K)$; equivalently, that test contributes the Minimax floor $\beta c_0^*/N = -\Theta(\beta \log K)/N$ to $R(y')$ while it contributes at least $\beta(\log(\alpha/2) - o(1))/N$ to $R(y_S)$.

Variance. The empirical sample variance of the reward within this specific group satisfies the exact pairwise identity $\mathrm{Var}[R] := \frac{1}{K}\sum_{k=1}^{K}(R(y_k) - \bar{R})^2 = \frac{1}{2K^2}\sum_{k,l}(R(y_k) - R(y_l))^2$. Keeping only the pairs $\{y, y'\}$ with $y \in S$ (each appears twice in the double sum) and using $|S| \ge \alpha K$:

$$\mathrm{Var}[R] \ge \frac{|S|}{K^2}\left(\frac{\beta}{2N}\log c\right)^2 \ge \frac{\alpha}{K}\left(\frac{\beta}{2N}\log c\right)^2 = \frac{\alpha\beta^2(\log c)^2}{4KN^2} = \Omega\left(\frac{1}{K}\right) \quad \text{for fixed } N \text{ and } \beta.$$

Since $\hat{A}_{PC}(y) = R(y) - \bar{R}$ is a mean shift, the empirical sample variance of the advantages exactly matches $\mathrm{Var}[R] = \Omega(1/K)$. This strictly positive lower bound holds deterministically for any specific group satisfying the count-separation condition, independently of η, guaranteeing usable gradient signal even when $\eta = 0$.

Boundary cases. Whenever all leave-one-out counts on every test are equal (e.g., a group split into two exactly symmetric clusters of size $K/2$ on every test, or the fully degenerate case where all traces agree), every trace receives the identical reward, so all advantages are zero and $\mathrm{Var}[\hat{A}_{PC}] = 0$ despite $H(\mathsf{q}^*) > 0$. Separation of *joint* execution outcomes alone does not produce a reward gap either. With five tests and binary outputs, let the group consist of the five cyclic rotations of the output vector 00111, with $2m+1$ copies each, and the five cyclic rotations of 01011, with $m+1$ copies each ($K = 15m+10$). The joint outcomes form ten modes with leave-one-out counts $2m$ and m, separated by a factor of 2, yet on every test exactly $3K/5$ programs output 1 and every program outputs three 1s and two 0s, so every program receives the same averaged reward $\frac{\beta}{5}(3c^*_{3K/5-1} + 2c^*_{2K/5-1})$ and $\mathrm{Var}[\hat{A}_{PC}] = 0$; no program agrees with the plurality on every test. Near-symmetric configurations (clusters of sizes $K/2 \pm 1$ on a test) yield a per-test reward gap that vanishes with K (it is $\Theta(\beta/K)$ for the exact log-reward, and the monotone lookup can assign the two counts the same coefficient, since nearby counts share coefficients), and hence a variance of order $1/K^2$ or smaller; for small K, two counts separated by the factor c can share a coefficient, which is the role of $K_0(c,\alpha)$. Hence no lower bound of the form $\Omega(H(\mathsf{q}^*)/K)$ holds uniformly over all asymmetric q^*. Empirically, stochastically sampled cold-start groups are generically asymmetric, which is why PC-GRPO yields usable advantage variance in 91.0% of 7B cold-start groups (Table 6). □

The threshold at the experimental group sizes. The threshold $K_0(c,\alpha)$ in Part (2) arises from passing from Binomial averages to individual coefficients. For a given peer count M the coefficients $\mathbf{c}^*(M)$ are an explicit table, so the condition can be checked directly: the bound $R(y_S) - R(y') \ge \frac{\beta}{2N}\log c$ holds as soon as $c_a^* - c_b^* \ge \frac{1}{2}\log c$ for the dominant count $a = X_{j^*}(y_S)$ and every $b \le a/c$, and the variance is nonzero as soon as $c_a^* > c_b^*$ for every such b. Table 2 lists, for the group sizes used in our experiments (16 candidates in training, $K \in \{4, 8, 16, 32, 64\}$ in Appendix I.9), the smallest dominant count a from which each condition holds for all larger a. For the training group size ($M = 15$ peers) and $c = 2$, the bound holds once the dominant output has leave-one-out count $a \ge 5$, i.e., once at least 6 of the 16 candidates produce it ($\alpha \ge 3/8$), and the variance is nonzero once $a \ge 4$; the entries for $c = 1.5$ reflect the coefficient plateaus.

F.3 Mitigating Small-Batch Estimation Bias

Translating the continuous Pairwise Coherence game into a practical algorithm for open-ended code generation and agentic environments introduces two classes of fundamental barriers: the statistical limits of finite-sample reward estimation, and the structural curse of dimensionality for trajectory-level consensus. By formalizing these limits, we demonstrate why the algorithmic interventions introduced in our framework are strict mathematical necessities.

Table 2: Smallest dominant leave-one-out count a from which the count-separation bound of Theorem 3 (left entry: $c_a^* - c_b^* \geq \frac{1}{2}\log c$ for all $b \leq a/c$) and a nonzero reward gap (right entry: $c_a^* > c_b^*$ for all $b \leq a/c$) hold for all larger a, for the constrained coefficients $\mathbf{c}^*(M)$ at the peer counts $M = K_{\text{act}} - 1$ of our experiments. A dash marks a combination for which the bound never holds at that M.

M (K_{act})	$c = 1.5$	$c = 2$	$c = 3$	$c = 4$
3 (4)	3 / 3	3 / 3	— / 1	1 / 1
7 (8)	4 / 4	4 / 4	5 / 4	5 / 1
15 (16)	10 / 9	5 / 4	5 / 4	5 / 1
31 (32)	18 / 11	11 / 5	5 / 5	5 / 5
63 (64)	20 / 19	11 / 5	5 / 5	5 / 5

In PC-GRPO, the ideal trace reward is $R(y) = \beta \log \mathsf{q}^*(\mathsf{E}(y))$. Estimating q^* from a small group K via the raw empirical logarithm $\log(X/K)$ introduces a severe $\mathcal{O}(1/K)$ structural bias due to Jensen's inequality, and the estimate is $-\infty$ whenever $X = 0$.

To correct this bias, one might naturally attempt to apply a local second-order Taylor expansion to the empirical reward. However, as established in the concurrent work (Mohri et al., 2026b), local Taylor corrections inherently break down near the boundary of the simplex ($p \sim 1/K$), where the empirical variance is no longer negligible compared to the mean, failing to provide uniform error guarantees. More precisely, Mohri et al. (2026b, Theorem 9) prove that any local Taylor or Laplace smoothing correction exhibits a non-vanishing worst-case bias of $\Omega(1/K)$ in the boundary layer, strictly necessitating a global optimization approach.

To obtain uniform control over the entire probability simplex, Mohri et al. (2026b) replace local corrections by a *Globally Optimized Minimax Polynomial Estimator*: a coefficient vector $\mathbf{c} \in \mathbb{R}^{K+1}$ defines the lookup c_X, whose expectation $P_{\mathbf{c}}(p)$ is a polynomial of degree K, and the coefficients minimize the worst-case gradient-weighted bias $\sup_p |pP_{\mathbf{c}}(p) - p\log p|$ over the full simplex. Mohri et al. (2026b, Theorem 5) establish a Ditzian–Totik lower bound proving that no estimator based on K samples can achieve a worst-case gradient bias better than $\Theta(1/K^2)$, and the unconstrained Minimax Polynomial attains this rate. Its coefficients, however, alternate in sign with magnitudes that grow with K, so the resulting reward is neither bounded below by its $X = 0$ value nor monotone in the count. PC-GRPO therefore uses the solution $\mathbf{c}^*$ of the same program under the shape constraints $c_0 \leq c_1 \leq \cdots \leq c_K \leq 0$, (13) in Appendix F.2: the reward is then bounded in $[\beta c_0^*, 0]$ and monotone by construction, the properties used by Theorems 3, 15, 16 and 18, while its worst-case weighted bias is $\epsilon_K = \mathcal{O}(1/K)$ (Theorem 15) rather than $\Theta(1/K^2)$. Numerically, the value of (13) lies between $0.06/K$ and $0.09/K$ for $1 \leq K \leq 63$, roughly a third of the weighted bias of the clipped plug-in $\log \frac{X+1/2}{K+1/2}$ used in the proof of Theorem 15, and the solution is constant on runs of neighboring counts whose length grows with K. The companion paper also develops an Augmented Quadratic Program (AQP) extension (Mohri et al., 2026b, Theorem 11) that traces the exact Bias-Variance Pareto frontier, allowing practitioners to trade asymptotic exactness for accelerated transient convergence.

In practice, the coefficients $\mathbf{c}^*$ are computed strictly offline, once per peer count K, by solving (13) on a fine grid of values of p as a linear program. During online training with PC-GRPO, the costly floating-point evaluation of the log-probability is entirely replaced by a simple $\mathcal{O}(1)$ array lookup: $\widehat{R}^*(X) = \beta c_X^*$.

The Curse of Dimensionality in Trajectory Consensus. Even with statistically optimal reward estimation, the Coherence Game faces a second fundamental obstacle when applied to complex, multi-step agentic domains where pass@$k = 0$. We now prove that in this cold-start regime, attempting to enforce sequence-level consensus exhibits an exponential sample complexity lower bound, mandating the shift to Process Coherence (Appendix E.3).

We now provide the formal statement and full proof for the exponential separation of sample complexity, corresponding to Theorem 1 in the main text.

Theorem 19 (Exponential Separation of Sample Complexity (Formal))**.** *Consider an agentic task of length T where the initial policy π_0 has an independent probability $\eta \in (0, 1)$ of making a correct,*

coherent reasoning step at any time t. The probability of generating a completely correct trajectory is $p = \eta^T$.

1. ***Outcome Coherence:*** *For at least two valid trajectories to appear in the group with probability at least a constant $c > 0$ (the event in which a valid trajectory earns a consensus reward above the floor), the group size must scale exponentially with task length: $K = \Omega\big(\eta^{-T}\big)$.*
2. ***Process Coherence (tree-based):*** *If the consensus target is evaluated step-by-step and the group is generated via a branching (tree-based) simulator that evaluates independent actions from shared intermediate states s_t, the group size required to form a valid consensus at any specific step scales strictly linearly: $K = \Omega\big(\eta^{-1}\big)$.*
3. ***Process Coherence (test-decomposed):*** *If the consensus decomposes over N independently evaluable criteria (e.g., unit tests), each producing a correct output with probability p_{test}, the group size scales as: $K = \Omega\big(p_{\text{test}}^{-1}\big)$.*

Proof. For Outcome Coherence, let $p = \eta^T$ be the probability of a single coherent trajectory, and $X \sim \text{Binomial}(K, p)$. Let A_i be the event that the i-th sampled trajectory is coherent. The event that we observe at least two successes occurs if and only if there exists at least one distinct pair of trials (i, j) that are both successful. We can express this event exactly as a union over all possible pairs: $\{X \geq 2\} = \bigcup_{1 \leq i < j \leq K} (A_i \cap A_j)$. By the union bound, the probability of the union is less than or equal to the sum of the probabilities: $\mathbb{P}(X \geq 2) \leq \sum_{1 \leq i < j \leq K} \mathbb{P}(A_i \cap A_j)$. Since the trajectories are sampled independently, the probability of any specific pair both being successful is the product of their individual probabilities: $p \cdot p = p^2$. There are exactly $\binom{K}{2} = \frac{K(K-1)}{2}$ combinations of distinct pairs in our sample, thus, we have $\mathbb{P}(X \geq 2) \leq \frac{K(K-1)}{2} p^2 < \frac{K^2 p^2}{2}$, where the last inequality holds for any $K > 0$. Thus, under the condition $c \leq \mathbb{P}(X \geq 2)$, we have $c < \frac{K^2 p^2}{2}$, that is, $K > \sqrt{2c}\,\eta^{-T} = \Omega(\eta^{-T})$. The lower bound is tight up to constants: since $\mathbb{P}(X \leq 1) = (1-p)^K + Kp(1-p)^{K-1} \leq (1 + Kp/(1-p))\, e^{-Kp}$, a group of size $K = \lceil \lambda / p \rceil$ with a constant λ already makes $\mathbb{P}(X \geq 2)$ a constant, and the same holds for η and p_{test} in Parts (2) and (3); the three parts therefore give the orders $\Theta(\eta^{-T})$, $\Theta(\eta^{-1})$, and $\Theta(p_{\text{test}}^{-1})$ of the group size needed for a valid consensus. Matching uses that valid samples share the behavior being matched: a single correct action per step in the model above, and the unique correct output on a test in Part (3).

For Process Coherence (tree-based), at each step t, the consensus target decomposes causally and the log-likelihood reward factorizes into a sum (Proposition 14). Since tree-based search branches from a shared valid prefix, generating a non-zero advantage signal at step t only requires at least two traces to agree on the immediate next action a_t (so that the leave-one-out count is ≥ 1), which occurs with marginal probability η. Let $X_t \sim \text{Binomial}(K, \eta)$ denote the number of traces proposing the correct next action. To guarantee that the probability of forming a valid consensus is bounded away from zero (i.e., $\mathbb{P}(X_t \geq 2) \geq c$ for some constant $c > 0$), standard binomial concentration requires $K = \Omega(\eta^{-1})$. By evaluating and forming consensus step-by-step, this approach effectively breaks the exponential dependence on the sequence length T.

For Process Coherence (test-decomposed), each of the K generated programs is evaluated on each criterion (unit test) independently. A program produces the correct output on a given test with marginal probability p_{test}, regardless of its behavior on other tests. Let $Y \sim \text{Binomial}(K, p_{\text{test}})$ be the number of programs passing a specific test. To form a consensus on any single test (requiring $Y \geq 2$) with probability at least $c > 0$, identical bounding logic dictates $K = \Omega(p_{\text{test}}^{-1})$.

Why this differs from sequential rollouts. For general sequential agentic tasks with independent rollouts, the evaluation is stateful. Forming a step-t consensus requires traces that are correct at step t, which conditionally necessitates an unbroken sequence of correct actions at all $t-1$ preceding steps to maintain a valid environment state. By the chain rule of probability, the joint probability of a single independent rollout remaining perfectly correct up to step t decays geometrically to $p = \eta^t$. Substituting this compound success probability into our lower bound mandates a sample complexity of $K = \Omega((\eta^t)^{-1}) = \Omega(\eta^{-t})$, thereby restoring the exponential bottleneck. In code generation, although the autoregressive generation of the program is sequential, the *evaluation decomposition* is over unit tests rather than chronological steps. Each test acts as a stateless, path-independent evaluation of the complete static program; executing one test does not condition on the outcome or

execution state of another. Since this sequential Markovian dependency is absent between evaluation criteria, the exponentially decaying probability η^t does not arise. □

Significance: This represents a fundamental impossibility result for standard outcome-based RLHF in long-horizon domains. For practical memory limits (e.g., $K \leq 64$), observing full valid trajectory consensus when pass@k = 0 is statistically impossible. By shifting the formulation to *Process Coherence*, we reduce the sample complexity from exponential to linear. Part (3) shows that for code generation, where consensus decomposes over independently evaluable unit tests, the linear scaling $K = \Omega(p_{\text{test}}^{-1})$ holds without requiring a tree-based simulator. Here p_{test} is the per-test agreement probability: it is large for a test that exercises a sub-component of the program and can be as small as η^T for a test that requires every decision to be correct, so the dependence is linear in the difficulty of the criterion. All three parts concern the group size at which a valid consensus appears at one step or on one test. The remark in the theorem further clarifies that for general sequential agentic tasks with independent rollouts, the linear bound does *not* hold: step-t consensus requires $K = \Omega(\eta^{-t})$, making tree-based simulation essential in that setting.

The Degenerate Consensus Impossibility (Symmetry of Coherence). The core premise of our framework is that semantic consistency across a group serves as a reliable proxy for correctness. However, pure ungrounded consensus contains a fundamental limitation: it is invariant to systemic hallucinations.

We now provide the formal statement and full proof for the degenerate consensus impossibility, corresponding to Theorem 4 in the main text.

Theorem 20 (Multi-Step Degenerate Consensus under KL-Regularized GRPO (Formal))**.** *Let the trace space $\mathcal{Y}$ be partitioned into a set of valid solutions $\mathcal{V}$ and a set of flawed-but-consistent loops $\mathcal{F}$. Let one GRPO update maximize the clipped surrogate objective with clipping parameter $\epsilon_c \in (0, 1)$ minus an exact KL divergence penalty against a reference policy π_0 with coefficient $\beta > 0$, exactly over the two mode masses $(\pi(\mathcal{F}), \pi(\mathcal{V}))$: this is the mode-level model of Appendix A, item (viii), in which the update redistributes probability between the two behavior classes and both the surrogate and the KL penalty are evaluated on the class masses. Let $\rho_t = \pi_t(\mathcal{F})/\pi_t(\mathcal{V})$ and $\rho_0 = \pi_0(\mathcal{F})/\pi_0(\mathcal{V})$.*

Assume the current policy is biased toward flaws such that $\rho_t \geq \rho_0$ and $\rho_t > 1$, and that the round's advantages satisfy $A(\mathcal{F}) > 0 > A(\mathcal{V})$ (as under ungrounded consensus when flaws form the majority of the sampled group). Let $\Delta A = A(\mathcal{F}) - A(\mathcal{V})$. Then the probability ratio of flaws to valid solutions at step $t + 1$ is bounded above by the dual constraint:

$$\rho_{t+1} \leq \min\left(\frac{\rho_t(1+\epsilon_c)}{1-\epsilon_c\rho_t},\ \rho_0 \exp\left(\frac{\Delta A}{\beta}\right)\right), \tag{14}$$

where the first (clipping) bound applies for $\rho_t < 1/\epsilon_c$ and is interpreted as $+\infty$ when $\epsilon_c\rho_t \geq 1$: in that regime $\pi_t(\mathcal{V}) \leq \epsilon_c\, \pi_t(\mathcal{F})$, the valid mass can be pushed beyond its lower clip without $\mathcal{F}$ reaching its upper clip, so the clipping constraint provides no finite ratio bound and only the KL ceiling binds.

Proof. Let $x_F = \pi_{t+1}(\mathcal{F})$ and $p_F = \pi_t(\mathcal{F})$. Because the sets partition the trace space, $\pi_{t+1}(\mathcal{V}) = 1 - x_F$ and $\pi_t(\mathcal{V}) = 1 - p_F =: p_V$. Write $p_0 = \pi_0(\mathcal{F})$.

Assuming all traces in $\mathcal{F}$ receive advantage $A(\mathcal{F}) > 0$ and those in $\mathcal{V}$ receive $A(\mathcal{V}) < 0$, the optimal policy update scales probabilities uniformly within each partition to maximize the surrogate objective subject to the KL penalty.

Because $A(\mathcal{V}) < 0$, the unclipped GRPO objective provides a gradient to decrease the valid mass $x_V = 1 - x_F$. The PPO clip zeroes out the surrogate gradient for mass reductions below $x_V = p_V(1 - \epsilon_c)$. Similarly, because $A(\mathcal{F}) > 0$, the objective seeks to increase the flawed mass x_F, with an upper clip limit $x_F \leq p_F(1 + \epsilon_c)$.

Since we are in the flawed-majority regime ($\rho_t > 1$), it follows that $p_F > p_V$. The maximum mass that $\mathcal{V}$ can give up before hitting its lower clip is $\epsilon_c p_V$, but the maximum that $\mathcal{F}$ can gain is $\epsilon_c p_F > \epsilon_c p_V$. Thus, $\mathcal{V}$ hits its lower clip boundary *first*. At this point, $\mathcal{V}$'s gradient contribution zeroes out. However, $\mathcal{F}$'s gradient remains strictly positive because $x_F/p_F = 1+\epsilon_c p_V/p_F = 1+\epsilon_c/\rho_t < 1+\epsilon_c$. Due to the simplex constraint ($x_V = 1 - x_F$), the optimizer continues to increase x_F (pushing x_V

below its lower clip boundary) until x_F reaches its own upper clip at $x_F = p_F(1+\epsilon_c)$, giving $x_V = 1 - p_F(1+\epsilon_c) = p_V - \epsilon_c p_F$.

Substituting $p_F/p_V = \rho_t$, and noting that this expression requires $p_V - \epsilon_c p_F > 0$, i.e., $\rho_t < 1/\epsilon_c$ (otherwise x_V reaches its clip floor before $\mathcal{F}$ is constrained and the clipping analysis yields no finite bound, consistent with the $+\infty$ convention in the statement), the probability ratio at step $t+1$ is strictly bounded by:

$$\rho_{t+1} = \frac{x_F}{x_V} \le \frac{p_F(1+\epsilon_c)}{p_V - \epsilon_c p_F} = \frac{\rho_t(1+\epsilon_c)}{1-\epsilon_c\rho_t}.$$

This establishes the structural clipping bound. For the KL bound, the exact KL divergence from the reference policy is $D_{\mathrm{KL}}(\pi_{t+1} \,\|\, \pi_0) = x_F \log(x_F/p_0) + (1-x_F)\log((1-x_F)/(1-p_0))$. Its derivative with respect to x_F is $\log(\rho_{t+1}/\rho_0)$. At the optimum of the regularized objective $J(x_F) = S(x_F) - \beta D_{\mathrm{KL}}$, the first-order condition yields $g(x_F^*) = \beta\log(\rho_{t+1}/\rho_0)$, where g is the surrogate subgradient. Since $g \le \Delta A$ globally, the KL penalty constrains the ratio to:

$$\beta\log\left(\frac{\rho_{t+1}}{\rho_0}\right) \le \Delta A \quad\Longrightarrow\quad \rho_{t+1} \le \rho_0 \exp\left(\frac{\Delta A}{\beta}\right).$$

Because the true policy update must satisfy both constraints, the dynamic is bounded by their minimum. This completes the proof. □

Significance: The dual bound reveals two qualitatively different regimes. The clip-induced bound $\rho_t(1+\epsilon_c)/(1-\epsilon_c\rho_t)$ shows that, within a single update, the degenerate ratio is amplified by a linear factor that increases with ρ_t, permitting exponential growth across repeated updates. The KL bound provides the crucial structural safeguard: the ratio ρ_{t+1} can never exceed the fixed ceiling $\rho_0\exp(\Delta A/\beta)$, regardless of the number of updates. (Note that ΔA is finite even in the extreme flawed-majority limit: because GRPO standardizes advantages over a finite empirical group of size K_{norm}, the group used for advantage normalization, i.e., $|G| = 2K$ under orbit sampling or K_{act} in Algorithm 1, each standardized advantage satisfies $|A| \le \sqrt{K_{\text{norm}}-1}$, so the maximum advantage difference is deterministically bounded by the finite-sample z-score constraint $\Delta A \le 2\sqrt{K_{\text{norm}}-1}$.) Thus, KL regularization transforms the degenerate consensus failure from unbounded exponential growth to a bounded (but potentially large) deviation from the reference policy. *Grounded Consensus* via plausibility filtering provides the necessary asymmetric anchoring to prevent even this bounded deviation.

Necessity of Grounding. The first-order condition used for the KL bound also gives a *lower* bound on the ratio, which shows that without grounding the collapse cannot be undone. Below we restate and prove Corollary 5 from the main text.

Corollary 5 (Necessity of Grounding)**.** *Under the assumptions of Theorem 4, $\rho_{t+1} > \rho_0$. Hence if $\rho_0 > 1$ and every round satisfies $A(\mathcal{F}) > 0 > A(\mathcal{V})$ (ungrounded consensus with flaws forming a strict majority of each sampled group), then $\rho_t > \rho_0 > 1$ for all $t \ge 1$: the valid mode never regains the majority. Moreover, for arbitrary advantages in the same model, $\rho_{t+1} < \rho_0$ requires $A(\mathcal{V}) > 0$ or $A(\mathcal{F}) < 0$, a round in which the valid mode receives the higher reward.*

Proof. In the notation of the proof of Theorem 20, write the clipped surrogate as a function of the flawed mass, $S(x_F) = S_F(x_F) + S_V(1-x_F)$, where $S_F(x) = A(\mathcal{F})\min\{x, p_F(1+\epsilon_c)\}$ for $A(\mathcal{F}) > 0$ (and $A(\mathcal{F})\max\{x, p_F(1-\epsilon_c)\}$ for $A(\mathcal{F}) < 0$), and $S_V(x) = A(\mathcal{V})\max\{x, p_V(1-\epsilon_c)\}$ for $A(\mathcal{V}) < 0$ (and $A(\mathcal{V})\min\{x, p_V(1+\epsilon_c)\}$ for $A(\mathcal{V}) > 0$; a piece with zero advantage is identically 0). Both pieces are concave and piecewise linear, with superdifferentials $\partial S_F(x) \subseteq [\min\{A(\mathcal{F}),0\}, \max\{A(\mathcal{F}),0\}]$ and $\partial S_V(x) \subseteq [\min\{A(\mathcal{V}),0\}, \max\{A(\mathcal{V}),0\}]$. The objective $J(x_F) = S(x_F) - \beta D_{\mathrm{KL}}(\pi_{t+1} \,\|\, \pi_0)$ is strictly concave with an interior maximizer x_F^* (the derivative of the KL term diverges at the endpoints), and its first-order condition reads $\beta\log(\rho_{t+1}/\rho_0) \in \partial S(x_F^*) = \partial S_F(x_F^*) - \partial S_V(1-x_F^*)$, i.e.,

$$\rho_{t+1} = \rho_0\exp\left(\frac{s_F - s_V}{\beta}\right) \quad \text{for some } s_F \in \partial S_F(x_F^*),\ s_V \in \partial S_V(1-x_F^*).$$

Lower bound. Under $A(\mathcal{F}) > 0 > A(\mathcal{V})$, $s_F \geq 0 \geq s_V$, so $\rho_{t+1} \geq \rho_0$, with equality only if $s_F = s_V = 0$. Equality means $x_F^* = p_0$, and $0 \in \partial S_F(p_0)$ requires $p_0 \geq p_F(1+\epsilon_c)$; since $\rho_t \geq \rho_0$ means $p_F \geq p_0$, this would give $p_0 \geq p_0(1+\epsilon_c)$, which is impossible. Hence $\rho_{t+1} > \rho_0$.

Irreversibility. If $\rho_0 > 1$ and every round satisfies $A(\mathcal{F}) > 0 > A(\mathcal{V})$, the lower bound applies at $t = 0$ (where $\rho_0 \geq \rho_0$ and $\rho_0 > 1$) and gives $\rho_1 > \rho_0$; if $\rho_t > \rho_0 > 1$, the assumptions of Theorem 20 hold again and give $\rho_{t+1} > \rho_0$. By induction, $\rho_t > \rho_0 > 1$ for all $t \geq 1$.

Necessary condition for improvement. By the first-order condition, $\rho_{t+1} < \rho_0$ requires $s_F < s_V$, hence $\min\{A(\mathcal{F}), 0\} < \max\{A(\mathcal{V}), 0\}$, i.e., $A(\mathcal{F}) < 0$ or $A(\mathcal{V}) > 0$. With group-standardized advantages in the two-mode model, the two advantages have opposite signs whenever they are nonzero ($n_F A(\mathcal{F}) + n_V A(\mathcal{V}) = 0$ for mode sizes n_F, n_V), so this is the event that the valid mode receives the higher reward. Under ungrounded consensus the reward is non-decreasing in a mode's leave-one-out count, so the valid mode receives the higher reward only when it forms the strict majority of the sampled group, the event that the standing hypothesis rules out; a valid-favoring round in the sense of Theorem 6 supplies exactly this event. □

Grounded Improvement: A Partial Converse. The above theorem establishes the *impossibility* of ungrounded consensus. We now prove a partial converse: once grounding gives the valid mode a positive advantage above every flawed mode, a single update improves the flawed-to-valid ratio up to a KL anchor, and the ratio remains uniformly bounded. The statement is in the idealized model of Theorem 20 (exact maximization over mode probabilities), without the assumption $\rho_t \geq \rho_0$.

Theorem 6 (Grounded Improvement)**.** *In the model of Theorem 4 extended to a finite set of behavior modes (flawed modes $\{\mathcal{F}_j\}_j$, π_t and π_0 positive on every mode), call a round* valid-favoring *if the valid mode's advantage margin $\delta := A(\mathcal{V}) - \max\{0, \max_j A(\mathcal{F}_j)\}$ is positive, and let $\alpha(\rho_t) := (1-\epsilon_c/\rho_t)/(1+\epsilon_c)$, so that $\alpha(\rho_t)\rho_t = (\rho_t - \epsilon_c)/(1+\epsilon_c)$.*

1. (One-step improvement.) *In a valid-favoring round, $\rho_{t+1} \leq \max\{\alpha(\rho_t)\rho_t,\ \rho_0 e^{-\delta/\beta}\}$, so $\rho_{t+1} < \rho_t$ whenever $\rho_0 e^{-\delta/\beta} < \rho_t$. If moreover $\rho_t \geq 1$ and $\delta > \beta \log(\max\{\rho_0, 1\}(1+\epsilon_c)/(1-\epsilon_c))$ (a* clip-dominated *round), then $\rho_{t+1} \leq \alpha(\rho_t)\rho_t$.*
2. (Uniform KL bounds.) *If in every round the advantage range is at most D, with $\min_z A_{t,z} \leq 0 \leq \max_z A_{t,z}$ (for our standardized advantages $D = 2\sqrt{K_{\max} - 1} + 1$ with $K_{\max} \geq K_{act}$, Appendix F.3), then $\rho_0 e^{-D/\beta} \leq \rho_{t+1} \leq \rho_0 e^{D/\beta}$ for every t: the ratio never exceeds the fixed ceiling $\rho_{\max} := \rho_0 e^{D/\beta}$, from which no round increases it and every valid-favoring round strictly decreases it.*

Proof. Setup. Write $x_z := \pi_{t+1}(z)$ and $p_z := \pi_t(z)$ for every mode $z \in \{\mathcal{V}\} \cup \{\mathcal{F}_j\}_j$, with $p_z, \pi_0(z) > 0$. The update maximizes

$$J(x) = \sum_z S_z(x_z) - \beta \sum_z x_z \log \frac{x_z}{\pi_0(z)}$$

over the simplex, where $S_z(x_z) = p_z A_z \min\{x_z/p_z, 1+\epsilon_c\}$ if $A_z > 0$ and $S_z(x_z) = p_z A_z \max\{x_z/p_z, 1-\epsilon_c\}$ if $A_z \leq 0$ is the clipped surrogate of mode z: a concave piecewise-linear function whose slope is A_z inside the clip region and 0 beyond it (PPO clipping is one-sided; it removes only the incentive to move further in the direction of the advantage). Since J is strictly concave and the entropy-type term has infinite slope at the boundary, the maximizer is unique and interior, and the first-order conditions give a multiplier λ and supergradients $s_z \in \partial S_z(x_z)$ of the concave functions S_z with $s_z - \beta \log(x_z/\pi_0(z)) = \lambda$ for all z, i.e.,

$$x_z = \frac{\pi_0(z)\, e^{s_z/\beta}}{Z}, \qquad Z := \sum_z \pi_0(z)\, e^{s_z/\beta}. \tag{15}$$

The slopes satisfy $s_z \in [\min\{A_z, 0\}, \max\{A_z, 0\}]$, and $s_z = A_z$ whenever x_z lies strictly inside the clip region on the side of its advantage ($x_z < p_z(1+\epsilon_c)$ if $A_z > 0$; $x_z > p_z(1-\epsilon_c)$ if $A_z \leq 0$). By (15),

$$\rho_{t+1} = \sum_j \frac{x_{\mathcal{F}_j}}{x_{\mathcal{V}}} = \sum_j \frac{\pi_0(\mathcal{F}_j)}{\pi_0(\mathcal{V})} \exp\left(\frac{s_{\mathcal{F}_j} - s_{\mathcal{V}}}{\beta}\right), \qquad \sum_j \frac{\pi_0(\mathcal{F}_j)}{\pi_0(\mathcal{V})} = \rho_0. \tag{16}$$

Part (2). In every round, $\min_i A_i \le 0 \le \max_i A_i$, so every slope lies in $[\min_i A_i, \max_i A_i]$, an interval of length at most D. Hence $|s_{\mathcal{F}_j} - s_V| \le D$ for all j, and (16) gives $\rho_0 e^{-D/\beta} \le \rho_{t+1} \le \rho_0 e^{D/\beta}$; since D does not depend on t, the ceiling is the same for all rounds. If $\rho_t = \rho_{\max}$, any round gives $\rho_{t+1} \le \rho_{\max}$, while a valid-favoring round gives, by Part (1), $\rho_{t+1} \le \max\{\frac{\rho_{\max}-\epsilon_c}{1+\epsilon_c},\ \rho_0 e^{-\delta/\beta}\} < \rho_{\max}$, because the first term is smaller than $\rho_{\max}$ and $\rho_0 e^{-\delta/\beta} < \rho_0 \le \rho_{\max}$. (For advantages standardized over G' one has $|A| \le \sqrt{K_{\text{act},t} - 1} \le \sqrt{K_{\max} - 1}$, and rejected candidates receive $\min_{G'} A - 1$ in Algorithm 1, so the range is at most $2\sqrt{K_{\max} - 1} + 1$ in every round.)

Part (1). In a valid-favoring round $\delta > 0$, i.e., $A(\mathcal{V}) > 0$ and $A(\mathcal{V}) > A(\mathcal{F}_j)$ for every j. (If the advantages are standardized over the plausible set, which contains $\mathcal{V}$, this holds exactly when $\mathcal{V}$ is the strict maximum there and the profile is non-constant: the strict maximum of a centered non-constant profile is positive, whereas a constant profile gives $A \equiv 0$ on G' and $\delta = 0$ even though rejected candidates sit at -1.) Two cases exhaust the possibilities.

(a) The clip binds: $x_V \ge p_V(1+\epsilon_c)$. Since $(1-x)/x$ is decreasing,

$$\rho_{t+1} = \frac{1 - x_V}{x_V} \le \frac{1 - p_V(1+\epsilon_c)}{p_V(1+\epsilon_c)} = \frac{p_F - \epsilon_c p_V}{p_V(1+\epsilon_c)} = \frac{\rho_t - \epsilon_c}{1+\epsilon_c},$$

where $p_F = 1 - p_V$. (Beyond the clip the surrogate is flat in x_V; nothing is claimed about the allocation of additionally drained mass.)

(b) The clip does not bind: $x_V < p_V(1+\epsilon_c)$. Then $\mathcal{V}$ is strictly inside its clip region, so $s_V = A(\mathcal{V})$, while $s_{\mathcal{F}_j} \le \max\{A(\mathcal{F}_j), 0\}$ for every j; hence $s_{\mathcal{F}_j} - s_V \le \max\{\max_j A(\mathcal{F}_j), 0\} - A(\mathcal{V}) = -\delta$, and (16) gives $\rho_{t+1} \le \rho_0 e^{-\delta/\beta}$.

Combining (a) and (b) proves the displayed bound. If $\rho_0 e^{-\delta/\beta} < \rho_t$, both terms of the maximum are below ρ_t, so $\rho_{t+1} < \rho_t$. Finally, if $\rho_t \ge 1$ and $\delta > \beta \log\left(\max\{\rho_0, 1\}\frac{1+\epsilon_c}{1-\epsilon_c}\right)$, then

$$\rho_0 e^{-\delta/\beta} \le \max\{\rho_0, 1\}\, e^{-\delta/\beta} < \frac{1-\epsilon_c}{1+\epsilon_c} \le \frac{\rho_t - \epsilon_c}{1+\epsilon_c},$$

so case (b) is impossible: it would give $\rho_{t+1} \le \rho_0 e^{-\delta/\beta} < \frac{\rho_t - \epsilon_c}{1+\epsilon_c}$, whereas $x_V < p_V(1+\epsilon_c)$ forces $\rho_{t+1} > \frac{\rho_t - \epsilon_c}{1+\epsilon_c}$. Hence case (a) holds and $\rho_{t+1} \le \alpha(\rho_t)\rho_t$. □

Remark 21 (From plurality to a positive margin)**.** *The hypothesis of Theorem 6 is the advantage margin, not a plurality of consensus mass. In the regime $|G'| \ge 2$, Algorithm 1 uses uniform weights over G', and the leave-one-out pseudo-count of a candidate on a test equals the number of other plausible candidates sharing its output. With strictly increasing coefficients and a non-constant reward profile over G', the valid mode then receives a strictly higher reward than every plausible flawed mode if the valid output is the strict count-plurality of G' on each evaluated test and each flawed mode disagrees with it on at least one evaluated test: the round is valid-favoring. The second condition is needed because a flawed program that produces the correct output on every evaluated test receives exactly the reward of the valid programs, so $\delta = 0$. In the soft-weight fallback ($|G'| \le 1$) the implication need not hold, because a candidate's own weight is excluded from its consensus mass: with $\tau_w = 0.45$, one valid candidate (seed score 1, weight 0.381) against fifteen flawed candidates (seed score 0, weight 0.041 each) in five behavior modes of three has the strict weighted plurality on a test (0.381 versus 0.124) but leave-one-out peer mass 0 and pseudo-count 0, whereas each flawed candidate has peer mass 0.086 and pseudo-count 1, so the valid candidate receives no higher, and with strictly increasing coefficients a lower, reward. The event is directly observable in the diagnostic groups of Appendix I.12; the clip-dominated threshold additionally depends on ρ_0, the reference policy's flawed-to-valid ratio, which those constructed groups do not determine. For $\beta = 0.02$ and $\epsilon_c = 0.2$ the threshold is 0.0081 when $\rho_0 \le 1$ and 0.054 for $\rho_0 = 10$, whereas under exact nondegenerate standardization $A(\mathcal{V}) \ge 1/\sqrt{K_{act} - 1} \approx 0.26$ whenever $\mathcal{V}$ is the top mode, so in practice the condition constrains the gap $A(\mathcal{V}) - \max_j A(\mathcal{F}_j)$.*

Remark 22 (Local drift threshold in the two-block model)**.** *In the two-block model of Theorem 20 (a single flawed mode $\mathcal{F}$), an ungrounded round ($A(\mathcal{F}) > 0 > A(\mathcal{V})$) with $1 \le \rho_t < 1/\epsilon_c$ satisfies*

$$\rho_{t+1} \le \max\{C(\rho_t)\rho_t,\ \rho_0\}, \qquad C(\rho_t) := \frac{1+\epsilon_c}{1 - \epsilon_c \rho_t}.$$

Indeed, if $\rho_{t+1} > C(\rho_t)\rho_t$ then $x_F > p_F(1+\epsilon_c)$ and $x_V < p_V - \epsilon_c p_F \le p_V(1-\epsilon_c)$, so both surrogates are flat, $s_F = s_V = 0$, and (15) gives $\rho_{t+1} = \rho_0$. Theorem 20 is the case $\rho_t \ge \rho_0$; the second term is the KL restoration toward the reference when $\rho_t < \rho_0$. If $\rho_0 \le \frac{1+\epsilon_c}{1-\epsilon_c} = C(1)$, then $\rho_{t+1} \le C(\rho_t)\rho_t$ for every $1 \le \rho_t < 1/\epsilon_c$, because $C(\rho)\rho$ is increasing on that interval. Suppose that whenever $\rho_t \in [1, 1+\eta]$ (with $1+\eta < 1/\epsilon_c$) a clip-dominated valid-favoring round occurs with conditional probability at least p. Then on that event, by Part (1) and the bound above (which also covers valid-favoring rounds that are not clip-dominated, since for them $\rho_{t+1} \le \max\{\rho_t, \rho_0\} \le C(1+\eta)\rho_t$, and neutral rounds, for which $\rho_{t+1} = \rho_0$),

$$\mathbb{E}[\rho_{t+1} \mid \mathcal{F}_t] \le \big[p\,\alpha(1+\eta) + (1-p)\,C(1+\eta)\big]\rho_t,$$

which is smaller than ρ_t if $p > p^*(\eta) := \frac{C(1+\eta)-1}{C(1+\eta)-\alpha(1+\eta)}$; as $\eta \to 0$, $p^*(\eta) \to p^* := \frac{1+\epsilon_c}{2}$ (0.55 for $\epsilon_c = 0.1$, 0.60 for $\epsilon_c = 0.2$). Thus p^* is a limiting sufficient threshold for negative expected drift under the stated conditions (two-block model, $\rho_0 \le C(1)$, a uniform conditional frequency of clip-dominated valid-favoring rounds, and a small neighborhood of the boundary); the bound is local, applying while the process remains in the neighborhood. With heterogeneous flawed modes the expansion in an ungrounded round can exceed $C(\rho_t)\rho_t$, because the KL term restores a flawed mode that is under-represented relative to π_0 beyond its clip, so the threshold is specific to the two-block model.

Remark 23 (Recurrence)**.** *If $\mathbb{P}[\text{valid-favoring round} \mid \mathcal{F}_t] \ge p_{\min} > 0$ for all t, then infinitely many valid-favoring rounds occur almost surely (Lévy's conditional Borel–Cantelli lemma), and the probability of none in t steps is at most $(1-p_{\min})^t$; by Part (1) each of them strictly decreases the ratio whenever $\rho_0 e^{-\delta_t/\beta} < \rho_t$. The assumption is not implied by $\pi_t(\mathcal{V})$ being bounded away from zero (a group with no flawed candidate carries no advantage signal); finite-sample frequencies of the event provide empirical evidence for it.*

Remark 24 (First entry into the valid-majority region)**.** *Extend the expansion factor by $C_{\text{ext}}(\rho) := \frac{1+\epsilon_c}{1-\epsilon_c\rho}$ for $1 \le \rho < 1/\epsilon_c$ and $C_{\text{ext}}(\rho) := +\infty$ for $\rho \ge 1/\epsilon_c$ (the rational expression is negative beyond its pole and must not be used there); in the two-block model with $\rho_0 \le C(1)$, an ungrounded round then satisfies $\rho_{t+1} \le \min\{C_{\text{ext}}(\rho_t)\rho_t,\ \rho_{\max}\}$ for every $\rho_t \ge 1$, by Remark 22 and Part (2). If $\rho_{\max} < 1$, every post-update policy already lies in the valid-majority region and the conclusion below is immediate; otherwise assume $\rho_{\max} \ge 1$ and let $\bar{C} := \sup_{1 \le \rho \le \rho_{\max}} \min\{C_{\text{ext}}(\rho),\ \rho_{\max}/\rho\} \le \rho_{\max} < \infty$. Fix a starting time $t_0 \ge 1$ and suppose that for every $t \ge t_0$ with $\rho_t \in [1, \rho_{\max}]$ a clip-dominated valid-favoring round occurs with conditional probability at least p. If $p > \bar{p} := \frac{\bar{C}-1}{\bar{C}-\alpha(\rho_{\max})}$, then $\gamma := p\,\alpha(\rho_{\max}) + (1-p)\,\bar{C} < 1$ and, by Part (1) and the bound above, $\mathbb{E}[\rho_{t+1} \mid \mathcal{F}_t] \le \gamma\rho_t$ whenever $t \ge t_0$ and $\rho_t \ge 1$. Hence, for the first entry time $\tau := \inf\{t \ge t_0 : \rho_t < 1\}$, the process $Y_t := \rho_t \mathbf{1}\{\tau > t\}$ satisfies $\mathbb{E}[Y_{t+1} \mid \mathcal{F}_t] \le \gamma Y_t$ for $t \ge t_0$, and Markov's inequality gives, conditionally on $\mathcal{F}_{t_0}$,*

$$\mathbb{P}[\tau > t_0 + n \mid \mathcal{F}_{t_0}] \le \gamma^n\,\rho_{t_0} \qquad (n \ge 0).$$

This gives eventual entry into the valid-majority region with a geometric tail under the assumption of a uniform frequency $p > \bar{p}$ (close to 1 when $\rho_{\max}$ is large).

Significance: This theorem is the partial converse to the Degenerate Consensus impossibility. Part (1) characterizes when a valid-favoring round reduces the flawed-to-valid ratio despite regularization toward the fixed reference: whenever the valid mode carries a positive advantage above every flawed mode, the ratio is bounded by the clip contraction $\alpha(\rho_t)\rho_t$ or by the KL anchor $\rho_0 e^{-\delta/\beta}$, whichever is larger; the ratio strictly decreases whenever it exceeds the anchor, and the anchor vanishes as $\beta \to 0$. Part (2) shows that the KL penalty confines the ratio to a fixed band around the reference, and that under grounding the upper edge of the band is not absorbing. Remark 22 identifies, in the two-block model of Theorem 20 with $\rho_0 \le (1+\epsilon_c)/(1-\epsilon_c)$ and in the limit of a small neighborhood of the boundary, the frequency $p^* = (1+\epsilon_c)/2$ of clip-dominated valid-favoring rounds that suffices for negative expected drift, a sufficient threshold specific to the two-block model. Whether rounds are valid-favoring is an empirical question: the controlled experiment of Appendix I.12 reports the sign of the valid candidate's advantage, a necessary condition for the event, and its duplicate-flaw diagnostic illustrates how identical flawed programs retain a concentrated consensus mode that tests can separate from the valid program but cannot split into different behavior modes. Both theorems are stated in the idealized model of Appendix A, item (viii).

G Test Input Generation as an Adversarial Game

Expanding upon the discussion in Section 4.3 of the main text, this section provides a deeper theoretical and practical analysis of framing test input generation as a min-max adversarial game.

Before discussing generation techniques, we formally establish the statistical limits of random testing via the necessary sample complexity of N.

Below, we formally state and prove the test input complexity bound referenced in the main text:

Theorem 25 (Sufficient Unit Test Complexity)**.** *Let G' be a fixed set of plausible programs. To estimate all pairwise functional disagreements among them to an additive accuracy ϵ with probability at least $1-\delta$, it is sufficient to draw N i.i.d. random test inputs $u_1, \dots, u_N$ from any distribution $\mathcal{D}_u$ over the input space, with N scaling as: $N = \mathcal{O}\Big(\frac{\log|G'| + \log(1/\delta)}{\epsilon^2}\Big)$.*

Proof. Condition on G' (the programs are fixed; all randomness is over the test inputs). For a fixed pair (y_i, y_j), define $Z_k = \mathbb{I}[y_i(u_k) \neq y_j(u_k)]$ for $k = 1, \dots, N$. Since the test inputs $u_1, \dots, u_N$ are drawn i.i.d. from $\mathcal{D}_u$, the variables $Z_1, \dots, Z_N$ are independent with values in $[0, 1]$. By Hoeffding's inequality, estimating a single pairwise distance to accuracy ϵ fails with probability at most $2\exp(-2N\epsilon^2)$. Applying a Union Bound over all $\binom{|G'|}{2} \leq |G'|^2/2$ pairs, the probability that *any* pair exceeds ϵ error is bounded by $|G'|^2 \exp(-2N\epsilon^2)$. Setting this equal to δ and solving for N yields the stated logarithmic bound. □

The Active Adversarial Objective. Theorem 25 reveals a surprising result: the sample complexity of random testing scales only *logarithmically* with the group size. However, random testing still remains insufficient in practice. The true bottleneck is not combinatorial, but *distributional sparsity*. If a subtle bug is only triggered by an edge-case manifold with a probability measure $\mu \ll \epsilon$, standard random sampling requires an intractable $N = \Omega(1/\mu)$ tests to trigger a disagreement.

To circumvent this sparsity limitation, we formalize $\mathcal{G}_{\text{test}}$ as an *active adversarial player*, conceptually bridging our framework with *mutation fuzzing* in software engineering. Rather than evaluating semantic coherence on a static distribution of random inputs, the objective of this Adversary is to dynamically synthesize edge-case inputs u that *maximize* the functional disagreement (Gini impurity) among the plausible programs in the current group G'.

Formally, for a given input u, the disagreement within the group can be quantified by the expected pairwise disagreement (or equivalently, the Gini impurity of the soft consensus distribution q^*_u). The Adversary's objective is to maximize this variance:

$$\max_{\mathcal{G}_{\text{test}}} \mathop{\mathbb{E}}_{u \sim \mathcal{G}_{\text{test}}} \left[\frac{1}{|G'|^2} \sum_{y_i, y_k \in G'} \mathbb{I}[y_i(u) \neq y_k(u)] \right] = \max_{\mathcal{G}_{\text{test}}} \mathop{\mathbb{E}}_{u \sim \mathcal{G}_{\text{test}}} \left[1 - \sum_{v \in \mathcal{Z}} (\mathsf{q}^*_u(v))^2 \right]. \tag{17}$$

By jointly optimizing the Policy (to maximize semantic coherence) and the Adversary (to maximize semantic incoherence), the framework transforms into a formal two-player adversarial game.

Since the environment execution $y(u)$ is fundamentally non-differentiable, $\mathcal{G}_{\text{test}}$ cannot be optimized via standard backpropagation. Instead, $\mathcal{G}_{\text{test}}$ can be instantiated as an auxiliary LLM parameterized by ϕ, updated via RL (e.g., PPO) to generate adversarial prompts, or driven by a black-box genetic algorithm (as in coverage-guided fuzzers) where the reward signal is strictly the Gini impurity of the generated execution traces.

Unsupervised Curriculum Discovery. This adversarial formulation naturally generates a self-adjusting curriculum. Early in training, the Adversary easily finds standard valid inputs that trigger disagreements among weak, syntactically diverse programs. As the Policy learns to generate consistently coherent code, the Adversary is forced to explore harder boundary conditions—such as null inputs, out-of-bounds indices, and unhandled environmental states—to secure its reward.

Crucially, because the Adversary operates solely on *disagreement* rather than ground-truth matching, it requires no human-annotated test oracles. The framework automatically discovers edge-case bugs and semantic errors and aligns the model to fix them entirely unsupervised. This active, adversarial

selection targets the exact decision boundaries where reasoning paths diverge, breaking the random-sampling bounds of Theorem 25 and allowing PC-GRPO to successfully resolve consensus using only a small, highly informative batch of adversarial tests per training step.

Computational Efficiency and Asynchronous Execution. A natural practical concern with execution-based alignment is the wall-clock latency of sandboxed evaluation. However, because the necessary test suite size scales only logarithmically with group size (Theorem 25), the execution burden remains low. Furthermore, candidate generation, adversarial test synthesis, and sandbox execution can be deployed using an asynchronous rollout architecture. CPU worker nodes function as an asynchronous data engine, executing the decoupled plausible sets in parallel and feeding evaluated coherence scores to the central GPU cluster for the optimal GRPO updates.

H Worked Example: Test Generation and Consensus Rewards

To make the dynamic test-generation mechanism concrete, this section walks through a complete example from our APPS training runs, showing exactly how the generator produces inputs, how candidates disagree on those inputs, and how the resulting consensus distribution assigns rewards without any ground-truth label.

How tests are generated and shared. In our APPS experiments, the test generator $\mathcal{G}_{\text{test}}$ produces *random inputs matching the problem's input specification*. For APPS problems, which provide free-text descriptions with implicit type signatures, we use a lightweight LLM prompt (the same base model, greedy decoding) to extract the input format from the problem description and generate a random input-generation script; this script is cached per-problem. All candidates in the group execute on the same generated inputs, and each test evaluates the entire program's input–output behavior, so disagreement is always well-defined across candidates: tests are input–output pairs on the shared problem specification rather than trace-specific assertions, and a test generated "for" one trajectory is therefore automatically meaningful for every other trajectory. Programs that crash or time out produce a distinct `error` output, naturally contributing to disagreement. For the adversarial active LLM generator (Appendix I.13), we use a separate LLM call that is conditioned on the problem description and the current group's disagreement pattern, with greedy decoding and the same base model; no RL or genetic-algorithm optimization of the generator was used in the reported experiments.

A concrete instance (APPS Interview split #1). The task: reach (n, m) from the origin in exactly k king-moves, maximizing the number of diagonal moves, or report -1 if infeasible. The input specification is "q queries, each a line of three integers $1 \le n, m, k \le 10^{18}$", and the generator samples random query tuples such as $(3, 3, 4)$.

How candidates disagree on this input. On the query $(3, 3, 4)$: a correct candidate outputs 2 (since n and m share parity but $k - n$ is odd, two diagonal moves must be sacrificed); two common flawed modes output 4 (omitting exactly this parity correction) or -1 (testing feasibility with $k < n + m$ instead of $k < \max(n, m)$); and a candidate that mis-parses the input contributes the `error` outcome.

The resulting reward. In one observed group of $|G'| = 8$ plausible candidates (4 correct, 2 with the parity flaw, 1 with the feasibility flaw, 1 crashing), the induced consensus distribution is

$$\{\, 2 : 0.5, \quad 4 : 0.25, \quad -1 : 0.125, \quad \texttt{error} : 0.125 \,\}.$$

The correct majority receives $\log 0.5 \approx -0.69$ while the singleton flaws receive $\log 0.125 \approx -2.08$, a 3× stronger penalty, with no ground-truth label needed. Note that no candidate needs to be verified correct for this signal to exist: the reward separates behavior modes purely through their agreement structure, exactly the mechanism formalized in Section 3. (This illustration uses raw group frequencies for exposition; the full algorithm uses leave-one-out counts and the Minimax polynomial estimator as described in Algorithms 1 and 3.)

Scope. This mechanism is straightforward for well-specified algorithmic tasks (specification-driven random input generation). For less structured domains, the generator design question is discussed in Appendix A, and end-to-end sensitivity to both the test count and the generator quality is quantified in Appendix I.7.

I Experimental Validation: Additional Details and Full Results

This appendix provides the detailed setup, full numerical tables, and additional interpretation supporting the main experimental claims in Section 5. The main body reports the compact benchmark comparison and mechanism checks in Table 1; here we expand each result, clarify the train/test protocol, and describe the stress-test settings used when strong models saturate the toy tasks. We first describe the shared 7B benchmark protocol and baselines in Appendix I.1, then provide the full 7B benchmark table with reward-signal descriptions in Appendix I.2 and the full mechanism-validation summary in Appendix I.3. We then provide the full results for the mechanism validations: cold-start gradient yield (Appendix I.5), the single-component ablation (Appendix I.6), end-to-end test-configuration sensitivity (Appendix I.7), reward hacking (Appendix I.8), finite-sample sensitivity (Appendix I.9), SWE-bench process coherence and first end-to-end resolve rates (Appendix I.10), degenerate consensus (Appendix I.11), the controlled grounding experiment for the valid-favoring condition of Theorem 6 (Appendix I.12), and adversarial input generation (Appendix I.13). Finally, we include the 1.5B benchmark sanity check in Appendix I.4.

I.1 General Experimental Setup

Main 7B benchmark setup. For the main benchmark evaluations in Table 1, we initialize from Qwen2.5-Coder-7B-Instruct (Hui et al., 2024). Model updates are performed using BF16 LoRA (Hu et al., 2022) with rank $r = 32$, $\alpha = 64$, and dropout 0.05. Unless otherwise stated, every RL-style method generates 16 traces per prompt per step: GRPO-style baselines (Vanilla, Partial-Credit, CodeRM-style, and R1-style GRPO) use group size $K = 16$ on the original prompt, and orbit-sampling methods (C2-GRPO and PC-GRPO) use $K = 8$ candidates per orbit element for a combined group of $|G| = 2K = 16$, so that generation compute is matched across all methods. All methods use maximum prompt length 2048, maximum completion length 768, KL penalty $\beta = 0.02$, soft plausibility temperature $\tau_w = 0.45$, plausibility threshold $\theta = 1.0$ (all seed tests passed), training temperature 0.7, and top-$p = 0.95$. We use AdamW (Loshchilov & Hutter, 2019) with peak learning rate 10^{-6} for RL-style methods and 1.5×10^{-5} for SFT-style methods. All trainable methods use a 512-example training cap unless the available training split is smaller. RL training runs for 3 epochs over the 512 training examples (prompt batch size 8, resulting in $3 \times 512/8 = 192$ gradient steps per seed); paraphrases are reused across epochs. Final evaluation uses greedy decoding and reports public-test execution accuracy.

Datasets and train/test protocol. We evaluate on APPS Intro and APPS Interview (Hendrycks et al., 2021), and LiveCodeBench release_v1 (Jain et al., 2025). For APPS Intro and APPS Interview, trainable methods are trained on the corresponding APPS train split and evaluated on the held-out APPS test split with a 500-example evaluation cap. For LiveCodeBench, trainable methods are trained on the same 512-example APPS Interview budget and evaluated zero-shot on LiveCodeBench release_v1 public tests; our release_v1 evaluation contains 400 examples. Thus, the LiveCodeBench column in Table 1 measures out-of-domain transfer rather than in-benchmark fine-tuning. HumanEval (Chen et al., 2021) sanity checks are included only in Appendix I.4, since these short-form function completion benchmarks are less diagnostic of the sparse-reward cold-start phenomena targeted by our method. The main benchmark table therefore focuses on APPS Interview, APPS Intro, and LiveCodeBench release_v1, where execution sparsity and temporal generalization are more central. We note that LiveCodeBench release_v1 problems predate the base model's release (late 2024); our LiveCodeBench evaluation measures out-of-domain transfer (APPS → LiveCodeBench) rather than temporal holdout generalization in the strictest sense.

Baselines. *Base* evaluates the original instruction model without training. *Self-play SFT* follows a CodeRL-style rejection-filtered self-training pipeline (Le et al., 2022): multiple candidates are sampled, public-test passing candidates are retained, and the model is fine-tuned on the accepted traces. *Vanilla GRPO* uses binary public-test rewards. *Partial-Credit GRPO* uses dense public-test progress, namely partial credit from the fraction of available tests passed, as a dense outcome-level reward (this is not a true step-level process reward model in the sense of Lightman et al. (2024); we use "Partial-Credit GRPO" throughout to reflect its outcome-based nature). *CodeRM-GRPO* uses synthetic tests generated by CodeRM-8B (Ma et al., 2025) as a learned code-reward-model baseline. *R1-style GRPO* adds a reasoning-format sparse-RL baseline inspired by DeepSeek-R1

(Guo et al., 2025); we note that the 768-token maximum completion length may constrain extended chain-of-thought reasoning, though longer budgets did not improve this baseline in preliminary experiments. *C2-GRPO* uses grounded hard consensus, while *PC-GRPO* uses soft plausibility and Minimax coherence. Best-of-16 Public is an inference-time candidate-selection ceiling using public tests and is not a deployable single-sample training method.

Reward and execution details. Reward computation uses at most two seed tests and three dynamic consensus tests during training, unless otherwise stated. For APPS-style standard-input problems, candidate programs are executed in an isolated Python subprocess with `stdin`; for function-style tasks, the required function is imported and called directly. Evaluation uses all available public tests for the selected benchmark instance. Candidate generation and evaluation are accelerated using vLLM (Kwon et al., 2023), while CPU worker pools execute sandboxed tests in parallel. Model training was conducted on 8×A100 80 GB GPUs (≈48 GPU-hours per training seed for APPS). The reported benchmark numbers use greedy pass@1 evaluation.

Controlled theorem-stress ablations. For failure-mode ablations where Qwen2.5-Coder-7B-Instruct saturates the toy task, we use Qwen2.5-Coder-0.5B-Instruct or Qwen2.5-Coder-1.5B-Instruct. These settings are designed to activate the assumptions of the corresponding theorem (e.g., flawed-majority initialization or reward-hacking vulnerability), rather than to measure benchmark capability. We nevertheless include 7B saturation checks in Tables 10 and 16 for completeness.

I.2 Full 7B Benchmark Table

Table 3 expands the compact benchmark panel in Table 1 by including the complete reward signal or supervision source for each method. This table is the main benchmark comparison used in the paper; the compact main-body version omits the reward-signal column only to save space.

Table 3: Full 7B code-alignment benchmark comparison with Qwen2.5-Coder-7B-Instruct. Values are mean pass@1 over five independent training seeds, with ± denoting sample standard deviation. Base is deterministic under greedy decoding and is reported from one evaluation. Best-of-16 Public is an inference-time public-test candidate-selection ceiling, not a single-sample trainable method.

Method	Reward Signal / Supervision	APPS Interview	LiveCodeBench release_v1	APPS Intro
Base	No training	10.60%	22.00%	37.20%
Self-play SFT	Rejection-filtered self-training	9.60 ± 0.64%	21.00 ± 0.88%	37.00 ± 0.74%
Vanilla GRPO	Binary public-test reward	10.80 ± 0.52%	22.75 ± 0.77%	36.80 ± 0.66%
Partial-Credit GRPO	Dense public-test progress	11.20 ± 0.58%	21.25 ± 0.93%	37.00 ± 0.71%
CodeRM-GRPO	Learned synthetic-test reward (Ma et al., 2025)	10.60 ± 0.49%	22.00 ± 0.69%	38.40 ± 0.82%
R1-style GRPO	Reasoning-format sparse RL (Guo et al., 2025)	11.00 ± 0.61%	22.00 ± 0.84%	37.40 ± 0.76%
C2-GRPO	Hard grounded consensus	11.60 ± 0.62%	24.00 ± 0.81%	39.20 ± 0.84%
PC-GRPO	**Soft plausibility + Minimax coherence**	**12.80 ± 0.64%**	**25.00 ± 0.86%**	**40.00 ± 0.81%**
Best-of-16 Public	Public-test candidate selection	18.00 ± 0.81%	39.50 ± 1.08%	48.80 ± 0.97%

Interpretation. The full table shows that PC-GRPO is the strongest trainable method on all three 7B evaluations. On APPS Interview, the sparsest of the three benchmark columns, PC-GRPO improves over Vanilla GRPO by 2.0 absolute points and over Partial-Credit GRPO by 1.6 absolute points. This supports the core claim that soft plausibility and coherence provide a useful signal beyond both binary outcome rewards and dense public-test progress rewards. On LiveCodeBench release_v1, all trainable methods are evaluated zero-shot after APPS Interview training; PC-GRPO still improves over the base model and Vanilla GRPO, indicating that the learned coherence signal transfers to a temporal out-of-domain benchmark. Best-of-16 Public remains substantially higher because it uses public tests at inference time to select among 16 candidates. We include it as a ceiling on available test-time headroom, not as a directly comparable single-sample trainable baseline.

I.3 Full Mechanism-Validation Summary

Table 4 expands panel (c) of Table 1. Each row links one theoretical claim to its diagnostic setting, the corresponding baseline failure mode, and the observed coherence-based behavior.

Table 4: Full mechanism-validation summary. This expands panel (c) of Table 1.

Claim tested	Setting	Baseline / failure mode	Coherence result
Cold-start gradient yield	APPS Interview, Qwen2.5-Coder-7B, orbit group $2K = 16$; see Table 6	33.3% of sampled groups are cold-start. Vanilla GRPO and hard-filtered C2-GRPO yield 0.0% usable variance in those groups.	Adaptive PC-GRPO yields **91.0%** usable variance in cold-start groups.
Process-vs-outcome separation	SWE-bench Lite, 300 issues, horizon $T = 10$; see Tables 12 and 13	Exact full-trajectory consensus drops to 0.7%, reflecting the exponential trajectory-matching limitation.	Step-wise process coherence gives **100.0%** any-step signal and remains 40.0% even at Step 10.
Finite-sample estimation	Qwen2.5-Coder-7B top-reward candidate selection; see Figure 2 and Table 11	Heuristic smoothing requires manually choosing ϵ, and the best ϵ shifts with group size K.	Minimax lookup is best at every K and removes the need for manual smoothing.
Reward hacking	Controlled 0.5B stress test; see Tables 9 and 10	Static proxy RM reaches reward 2.960 but only 60.0% correctness.	PC-GRPO reward remains bounded at $-0.004 \leq 0$ and reaches **97.5%** correctness.
Degenerate consensus	Controlled 1.5B stress test; see Tables 15 and 16	Ungrounded consensus reaches its proxy ceiling with 0.0% final correctness.	Grounded consensus reaches **95.0%** correctness.
Adversarial tests	Controlled fuzzing, $N = 3$; see Table 18	Random and mutation fuzzing trigger Gini impurity 0.325 and 0.070, respectively.	Active 7B adversarial generation triggers **0.480** Gini impurity.

I.4 Supplementary 1.5B Benchmark Sanity Check

For completeness, Table 5 reports the additional Qwen2.5-Coder-1.5B-Instruct comparison. These smaller-model results show the same qualitative trend as the 7B table in Table 1: PC-GRPO outperforms Vanilla GRPO and C2-GRPO on sparse APPS Interview and LiveCodeBench settings.

Table 5: Supplementary 1.5B benchmark sanity check. These results are included for completeness; the main text details the 7B model comparisons against stronger baselines. Values represent mean pass@1 over repeated training runs, with ± denoting the sample standard deviation over five seeds.

Method	Reward Signal	HumanEval	APPS Intro	APPS Interview	LiveCodeBench
SFT	Static Gold Traces	65.24 ± 1.14%	16.52 ± 0.79%	4.56 ± 0.46%	9.80 ± 0.68%
Vanilla GRPO	Binary Unit Tests	69.51 ± 1.03%	20.60 ± 0.86%	4.12 ± 0.42%	9.25 ± 0.71%
C2-GRPO	Hard Consensus	71.34 ± 1.04%	21.80 ± 0.84%	6.04 ± 0.43%	12.25 ± 0.70%
PC-GRPO	**Soft Consensus**	**72.56 ± 0.98%**	**23.20 ± 0.82%**	**7.64 ± 0.46%**	**14.25 ± 0.73%**

Interpretation. The 1.5B results are not the primary benchmark evidence, but they are useful for showing consistency with the 7B experiments in Table 1. At 1.5B scale, the sparse APPS Interview and LiveCodeBench settings are much more severe: Vanilla GRPO underperforms or barely improves over static supervision because many groups contain no fully correct trace. Both C2-GRPO and PC-GRPO improve over Vanilla GRPO, with PC-GRPO giving the strongest performance. This supports the same qualitative conclusion as the 7B table: coherence-based rewards provide useful signal where binary outcome rewards are too sparse.

I.5 Cold-Start Gradient-Yield Ablation

This subsection provides the full cold-start table summarized in panel (c) of Table 1 and in the first row of Table 4. A group is *cold-start* if no sampled candidate passes all evaluated public tests.

Table 6: Cold-start gradient-yield ablation on APPS Interview. "pass@K > 0" is the complement of the cold-start rate.

Model	Cold-start	pass@K > 0	Vanilla cold yield	C2 cold yield	PC cold yield
Qwen2.5-Coder-0.5B-Instruct	82.0%	18.0%	0.0%	0.0%	**97.6%**
Qwen2.5-Coder-7B-Instruct	33.3%	66.7%	0.0%	0.0%	**91.0%**

Interpretation. This ablation directly addresses the model-capacity concern summarized in panel (c) of Table 1. Scaling from 0.5B to 7B reduces the cold-start rate from 82.0% to 33.3%, showing that stronger models do solve a larger fraction of sampled groups. However, the cold-start regime remains substantial even at 7B. In those cold-start groups, binary Vanilla GRPO produces zero reward variance by definition: no candidate passes the evaluated public tests, so all binary rewards are zero and its 0.0% yield is a deterministic consequence of the cold-start definition rather than an empirical finding. Hard-filtered C2-GRPO, whose plausible set is defined by the seed tests, likewise produced no usable variance in these groups. The empirical content of the table lies in the cold-start rate, the C2-GRPO yield, and the PC-GRPO yield. Adaptive PC-GRPO remains active because soft plausibility and behavioral disagreement can assign fractional advantages to partially successful traces; it yields nonzero variance in 91.0% of 7B cold-start groups.

I.6 Single-Component Ablation

Table 7 reports a full single-component ablation on APPS Interview with Qwen2.5-Coder-7B-Instruct (5 seeds, protocol of Appendix I.1), removing one mechanism of PC-GRPO at a time. *w/o seed grounding* removes the seed-test anchor entirely (fully ungrounded consensus); *w/o soft plausibility* replaces the adaptive softmax weighting with the hard filter of Algorithm 3; *w/o Minimax* replaces the Minimax Polynomial lookup with the best manually tuned heuristic smoothing floor ($\epsilon = 0.10$).

Table 7: Single-component ablation on APPS Interview (Qwen2.5-Coder-7B-Instruct, 5 seeds).

Variant	pass@1
Vanilla GRPO (binary reward)	$10.80 \pm 0.52\%$
C2-GRPO (+ seed filter, hard consensus)	$11.60 \pm 0.62\%$
PC-GRPO w/o seed grounding (ungrounded)	$10.28 \pm 0.90\%$
PC-GRPO w/o soft plausibility	$12.20 \pm 0.66\%$
PC-GRPO w/o Minimax (tuned floor $\epsilon = 0.10$)	$12.08 \pm 0.71\%$
PC-GRPO (full)	$\mathbf{12.80 \pm 0.64\%}$

Interpretation. Roughly half of the total gain over binary rewards comes from the consensus reward with seed grounding, and half from soft plausibility plus Minimax estimation. Notably, removing grounding entirely is *worse* than Vanilla GRPO, as Theorem 4 predicts: ungrounded consensus can actively reward agreement on flawed behavior. We caution that the individual drops from removing soft plausibility (0.60 points) and the Minimax estimator (0.72 points) are within one seed standard deviation and should be read directionally; the statistical case for these two components rests on the targeted isolations below, in which their effect is large. This table complements the orthogonal single-mechanism isolations reported elsewhere in the paper: Table 6 isolates soft plausibility in the cold-start regime (91.0% vs. 0.0% gradient yield), Table 11 isolates the Minimax estimator against every tuned smoothing floor, and Table 15 isolates seed grounding under the degenerate-consensus failure mode (95.0% vs. 0.0% final correctness).

I.7 End-to-End Test-Configuration Sensitivity

The adversarial-efficiency analysis of Table 18 measures the reward signal (Gini impurity) as a function of the test suite size and generator strategy. Here we additionally measure *end-to-end* sensitivity by retraining PC-GRPO on APPS Interview under varied test configurations (5 seeds, protocol of Appendix I.1).

Table 8: End-to-end sensitivity of PC-GRPO to the dynamic test configuration on APPS Interview (Qwen2.5-Coder-7B-Instruct, 5 seeds). Top: varying the consensus test count N with the default random generator. Bottom: varying the generator strategy at $N = 3$.

Axis	Configuration	pass@1
Test count	$N = 1$	$11.92 \pm 0.71\%$
	$N = 3$ (default)	$12.80 \pm 0.64\%$
	$N = 5$	$12.96 \pm 0.61\%$
Generator (at $N = 3$)	Mutation fuzzing (weak)	$11.80 \pm 0.74\%$
	Random generation (default)	$12.80 \pm 0.64\%$
	Adversarial active LLM (Appendix I.13)	$13.08 \pm 0.62\%$

Interpretation. Accuracy is stable once $N \geq 3$, mirroring the Gini-impurity saturation in Table 18 and consistent with the logarithmic sufficiency bound of Theorem 25. Generator quality matters more than test count, but degrades gracefully: the deliberately weak mutation-fuzzing generator (near-zero Gini impurity at small N) degrades performance toward hard-consensus C2-GRPO rather than collapsing, while upgrading the default random generator to the adversarial generator yields a directional improvement, though the difference is within one standard deviation. PC-GRPO is thus robust to generator quality: random generation already suffices, weak generators fail gracefully, and better generators help but are not required.

I.8 Reward-Hacking Stress Test

This subsection provides the reward-hacking diagnostic summarized in panel (c) of Table 1 and in Table 4. We construct a controlled sorting prompt where a static proxy reward favors syntax and verbosity.

Table 9: Reward-hacking stress test on Qwen2.5-Coder-0.5B-Instruct.

Method	Final proxy / coherence reward	Final true correctness
Static Proxy RM	2.960	60.0%
PC-GRPO	-0.004	**97.5%**

Interpretation. The static proxy reward can be increased by superficial features such as syntax markers and verbosity, yet this does not imply robust functional correctness: the final proxy reward reaches 2.960 while true correctness is only 60.0%. By contrast, PC-GRPO's coherence reward remains bounded near zero, as predicted by Theorem 2, and the final true correctness reaches 97.5%. This experiment is a controlled stress test of the reward landscape, not a benchmark-capability comparison.

Qwen2.5-Coder-7B-Instruct solves the controlled sorting prompt at initialization or after minimal optimization, so the reward-hacking stress test is not diagnostic at that scale. We report it only to confirm that the boundedness claim remains numerically valid.

Table 10: Reward-hacking saturation check on Qwen2.5-Coder-7B-Instruct. The task is too easy for the 7B model, so both methods achieve 100% true correctness.

Method	Early		Final	
	Proxy / coherence reward	True correctness	Proxy / coherence reward	True correctness
Static Proxy RM	1.924	100.0%	1.925	100.0%
PC-GRPO	0.000	100.0%	0.000	100.0%

7B saturation interpretation. At 7B scale, the toy sorting prompt is saturated: both the static proxy and PC-GRPO variants achieve 100.0% true correctness. This does not contradict the 0.5B stress test; rather, it shows that the reward-hacking failure mode is diagnostic only when the task is difficult enough for the model to exploit the proxy without already solving the problem. We therefore use the 0.5B run as the main theorem-stress evidence and include the 7B run only as a completeness check.

I.9 Finite-Sample Sensitivity Grid

This subsection gives the full numerical values and full-size visualization for the 7B model-conditioned finite-sample ridge summarized in panel (b) of Table 1. We evaluate Qwen2.5-Coder-7B-Instruct candidate pools under group sizes $K \in \{4, 8, 16, 32, 64\}$ and heuristic probability floors $\epsilon \in \{0.01, 0.05, 0.10, 0.20\}$. For each (K, ϵ), we select the top-rewarded candidate and report public-test success. The Minimax lookup column uses the precomputed coefficient lookup used by PC-GRPO rather than a manually tuned floor.

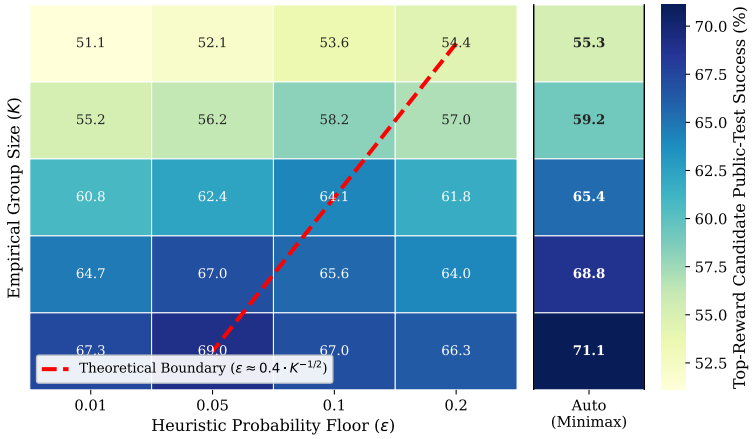


Figure 2: Finite-sample sensitivity grid with Qwen2.5-Coder-7B-Instruct. Values report top-reward candidate public-test success across empirical group sizes K and heuristic probability floors ϵ. The best manually tuned floor shifts from $\epsilon = 0.20$ at $K = 4$ to $\epsilon = 0.05$ at $K = 64$, tracking the predicted inverse-root boundary $\epsilon \approx 0.4K^{-1/2}$. The Minimax lookup used by PC-GRPO does not require choosing a heuristic smoothing floor.

Table 11: Finite-sample sensitivity grid with Qwen2.5-Coder-7B-Instruct. Values report top-reward candidate public-test success (%).

Group Size (K)	Heuristic Probability Floor (ϵ)				Minimax Lookup PC-GRPO
	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.20$	
$K = 4$	51.1%	52.1%	53.6%	**54.4%**	**55.3%**
$K = 8$	55.2%	56.2%	**58.2%**	57.0%	**59.2%**
$K = 16$	60.8%	62.4%	**64.1%**	61.8%	**65.4%**
$K = 32$	64.7%	**67.0%**	65.6%	64.0%	**68.8%**
$K = 64$	67.3%	**69.0%**	67.0%	66.3%	**71.1%**

Interpretation. The row-wise optima follow the inverse-root ridge of the bias–variance trade-off of a fixed smoothing floor. For $K = 4$, heavy smoothing ($\epsilon = 0.20$) performs best because the empirical consensus distribution has high variance. As K grows, the optimal smoothing floor decreases: $\epsilon = 0.10$ is best for $K = 8$ and $K = 16$, while $\epsilon = 0.05$ is best for $K = 32$ and $K = 64$. The plotted boundary $\epsilon \approx 0.4K^{-1/2}$ in Figure 2 is an empirical calibration of the theoretical $\Theta(K^{-1/2})$ scaling. The Minimax lookup column is not a tuned value of ϵ; it uses the precomputed coefficient lookup described in Appendix F.3, and it outperforms the best manually tuned heuristic floor at every group size.

I.10 SWE-bench Lite Process-Coherence Horizon Stress Test

This subsection expands the SWE-bench Lite consensus-yield row summarized in panel (c) of Table 1 and in Table 4. We evaluate consensus-yield proxies on the full SWE-bench Lite test split of 300 issues (Jimenez et al., 2024). We prompt Qwen2.5-Coder-7B-Instruct to generate $K = 16$ exploratory bash-command trajectories of length $T \in \{5, 8, 10\}$. We measure exact full-trajectory consensus and exact step-wise command consensus. This experiment intentionally does not execute repositories and does not report resolve rates; it only measures whether process-level consensus supplies a training signal when outcome-level trajectory matching becomes combinatorially rare. We acknowledge that exact syntactic matching is a conservative proxy that penalizes valid but syntactically diverse paths; a full operationalization of environmental invariance (Section 3) for SWE-bench would require executing each command prefix in a sandboxed repository clone, which we leave to future work.

Table 12: Horizon stress test on SWE-bench Lite.

Horizon	Outcome exact consensus	Process any-step consensus	Final-step process consensus
$T = 5$	34.7%	100.0%	89.7%
$T = 8$	3.0%	100.0%	59.0%
$T = 10$	0.7%	100.0%	40.0%

Horizon-level interpretation. The horizon summary isolates the exponential-separation phenomenon. Exact full-trajectory consensus drops from 34.7% at $T = 5$ to 3.0% at $T = 8$ and 0.7% at $T = 10$. In contrast, process coherence retains a 100.0% any-step signal across all horizons. This is the empirical counterpart of Theorem 1: matching full trajectories becomes exponentially harder with horizon length, while step-wise agreement remains locally observable.

Table 13: Full 10-step SWE-bench Lite process-coherence table on the 300-issue test split with Qwen2.5-Coder-7B-Instruct.

Coherence Horizon	Overall Yield	Step-wise Exact Command Consensus									
		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Step 10
Outcome Coherence (Full Trajectory)	0.7%	–	–	–	–	–	–	–	–	–	–
Process Coherence (Step-wise)	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**96.7%**	**92.3%**	**85.0%**	**72.7%**	**62.7%**	**46.0%**	**40.0%**

Step-level interpretation. The full 10-step table shows that process coherence remains dense at early and middle steps even when full-trajectory consensus has dropped to near zero. Exact command consensus is 100.0% through Step 3, remains 85.0% at Step 6, and still provides 40.0% signal at Step 10. This supports the use of step-wise coherence as an implicit PRM: rather than requiring two entire trajectories to match, the reward can assign local credit to shared exploratory actions.

First end-to-end resolve rates. The consensus-yield proxies above measure whether a training signal exists, not whether training on it resolves issues. To close this gap, we additionally report first end-to-end resolve rates on the same SWE-bench Lite test split of 300 issues, using the same command-generation scaffold as the stress test above. The key distinction is that *evaluation* of resolve rates executes the generated patches against the repository test suite via the SWE-bench harness, and *training* uses binary test-suite pass/fail as the reward signal for Vanilla GRPO, while PC-GRPO uses syntactic command-level process consensus (computed without execution) as the training reward. Because execution feedback is absent, plausibility weighting is disabled (uniform weights), isolating the raw ungrounded consensus signal; a full operationalization for SWE-bench requires executing

each command prefix in a sandboxed repository clone to derive a valid progress score. We note that training on ungrounded consensus does not contradict Theorem 4: the degenerate-collapse dynamic requires a flawed-but-consistent majority mode, whereas exploratory bash trajectories are highly fragmented (Table 13), and the KL ceiling of (7) bounds any degenerate drift over the short training budget; grounding via executed prefixes is nevertheless expected to strengthen this result. Because repository-level agentic environments are high-variance, each trained method is run with five independent training seeds and we report the mean and sample standard deviation of the number of resolved issues across seeds; the base model is decoded greedily and is therefore deterministic. We note that the training data is drawn from the SWE-bench *train* split, which is fully disjoint from the 300 test issues; no test-time adaptation or data leakage occurs.

Table 14: End-to-end SWE-bench Lite resolve rates (300 issues, same scaffold as Table 12). Trained methods report the mean ± sample standard deviation across five independent training seeds; Base is evaluated deterministically.

Method	Resolved	Resolve rate
Base	19.0 / 300	6.3%
Vanilla GRPO	20.2 ± 2.6 / 300	6.7 ± 0.9%
PC-GRPO	**25.0 ± 2.0** / 300	**8.3 ± 0.7%**

Resolve-rate interpretation. The improvement is modest in absolute terms but consistent across seeds: PC-GRPO resolves 4.8 more issues than Vanilla GRPO on average, which exceeds the seed-to-seed variability of either method (Welch's t-test on the per-seed resolved counts, $t = 3.27$, $p \approx 0.01$), whereas Vanilla GRPO is not separated from the untrained base model ($p \approx 0.36$). This is directionally consistent with our benchmark results and suggests that the process-coherence signal measured in Tables 12 and 13 translates into genuine end-to-end repository resolution rather than only consensus-yield proxies. Hidden-test SWE-bench evaluation at scale, and grounding the process reward via executed command prefixes, remain future work (Appendix A).

I.11 Degenerate Consensus Failure Mode

This subsection provides the degenerate-consensus stress test summarized in panel (c) of Table 1 and in Table 4. We evaluate a controlled prompt where ungrounded consensus can reward agreement on execution failures.

Table 15: Degenerate-consensus stress test on Qwen2.5-Coder-1.5B-Instruct.

Method	Step-10 reward	Step-10 acc.	Final reward	Final acc.
Ungrounded Consensus	-0.004	3.1%	0.000	0.0%
Grounded Consensus	-0.188	62.5%	-0.025	**95.0%**

Interpretation. This stress test activates the flawed-majority condition in Theorem 4. The ungrounded policy maximizes the proxy by collapsing to a unanimous error mode: its final proxy reward reaches the ceiling, but final correctness is 0.0%. Grounded consensus breaks this symmetry using a minimal plausibility seed and reaches 95.0% final correctness. This experiment demonstrates why pure agreement is insufficient: coherence must be anchored by at least weak grounding.

The Qwen2.5-Coder-7B-Instruct model solves the toy degenerate-consensus prompt at initialization, so the flawed-majority assumption is inactive. We therefore include the saturated 7B run only for completeness.

Table 16: Degenerate-consensus saturation check on Qwen2.5-Coder-7B-Instruct.

Method	Step-10 reward	Step-10 acc.	Final reward	Final acc.
Ungrounded Consensus	0.000	100.0%	0.000	100.0%
Grounded Consensus	0.000	100.0%	0.000	100.0%

7B saturation interpretation. At 7B scale, the toy prompt is already solved by both grounded and ungrounded variants, so the flawed-majority assumption is inactive. This is why the diagnostic result is the 1.5B stress test, while the 7B table is included only to show that no degenerate collapse occurs once the task is saturated.

I.12 Controlled Grounding Experiment: The Valid-Favoring Condition

Theorem 6 requires the valid mode to carry a positive advantage above every flawed mode (a *valid-favoring* round), which is strictly stronger than a valid candidate merely passing seed tests. To examine both the adversarial boundary case that motivates this condition and the generic regime encountered in practice, we constructed 40 groups from APPS training problems, each containing 1 correct solution and 9 flawed candidates that all pass the basic seed tests, and measured the sign of the valid candidate's advantage $A(\mathcal{V})$ under the log-consensus reward. A positive sign is a necessary condition for a valid-favoring round (the event further requires $A(\mathcal{V})$ to exceed every flawed candidate's advantage), so the frequencies below are upper bounds on the frequency of that event in these groups.

Table 17: Controlled grounding experiment: 40 groups, each with 1 valid + 9 seed-passing flawed candidates. We report the fraction of groups where the valid candidate receives strictly positive advantage.

Flawed-candidate regime	Largest flawed-mode mass	$A(\mathcal{V}) > 0$ at $N = 3$	$A(\mathcal{V}) > 0$ at $N = 5$
Identical copies (boundary case)	0.90	0.0%	0.0%
Distinct seed-passing implementations (generic)	0.24 (avg.)	**87.5%**	**95.0%**

Interpretation. The two regimes cleanly separate the boundary case from practice. *Identical copies* of one buggy program (the exact adversarial construction excluded by the strict-plurality condition): no behavioral test can separate exact duplicates, so the flawed mode retains mass 0.90 and $A(\mathcal{V}) < 0$ in every group, confirming this as the boundary case in which the valid mode cannot lead. *Distinct seed-passing implementations* (the generic case in actually sampled groups): $N = 3$ generated tests partition the flawed candidates, because on each input the correct-output mode includes the valid candidate plus all mutants whose bug does not trigger on that input, typically forming the plurality, while triggered mutants scatter into small wrong-output modes (largest flawed-mode mass 0.24 on average, 4.2 behavior modes per group). The valid candidate therefore sits in the plurality mode on nearly every input and receives $A(\mathcal{V}) > 0$ in 87.5% of groups at $N = 3$ and 95.0% at $N = 5$. Exact behavioral duplicates are rare in real sampled groups because generation is stochastic, which is why grounding succeeds empirically (Tables 6 and 15) despite the adversarial boundary case.

I.13 Adversarial Input Generation

This subsection provides the adversarial-fuzzing table summarized in panel (c) of Table 1 and in Table 4. We manually construct a diverse plausible group G' of semantically distinct implementations with hidden edge-case bugs.

Table 18: Maximum Gini impurity triggered by test suites of size N using a Qwen2.5-Coder-7B-Instruct adversarial generator.

Generator Strategy	**Max Gini Impurity at Suite Size N**			
	$N = 1$	$N = 3$	$N = 5$	$N = 10$
Uniform Random	0.208	0.325	0.326	0.351
Mutation Fuzzer	0.000	0.070	0.166	0.275
Adversarial Active LLM	**0.320**	**0.480**	**0.480**	**0.480**

Interpretation. The adversarial generator discovers high-disagreement edge cases much more quickly than unguided search. With a single test, the adversarial LLM reaches Gini impurity 0.320, while mutation fuzzing triggers no disagreement. By $N = 3$, the adversarial generator reaches the strongest observed disagreement 0.480, compared with 0.325 for random testing and 0.070 for mutation fuzzing. This validates the min-max formulation in Section 4.3: actively optimizing for disagreement finds informative boundary cases with far fewer tests than mutation-based exploration.