DESIGNING COUPLED SAMPLERS FOR LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling inference compute and reinforcement learning rely on generating batches of language model rollouts. These are typically sampled independently, which can lead to redundancy among the samples. Prior work on arithmetic sampling shows that rollouts can be *coupled* in a way that creates structured dependencies, while guaranteeing the resulting samples have the correct marginal distribution. However, this leaves open a consequential design choice—*how should the rollouts be coupled?* Infinitely many joint distributions preserve the marginals, and choosing the coupling is a crucial design choice in classical Monte Carlo. In this paper, we characterize this design space for test-time scaling and reinforcement learning. We first analyze how the choice of coupling shapes both coverage and policy-gradient variance, revealing a robustness–optimality tradeoff between different couplings. We then validate our theory by characterizing these tradeoffs empirically in four controlled reasoning benchmarks. We find that coupling boosts coverage, with the best sampler matching i.i.d. pass@k with 25–47% fewer rollouts. In reinforcement learning, all coupled samplers match i.i.d performance with 18-24% fewer steps and also improve final pass@1 by up to 2.3 pp. We also validate these results on coding benchmarks.

1 INTRODUCTION

Generating multiple rollouts from a language model (LM) is a fundamental primitive in both test-time scaling and reinforcement learning (RL). At inference time, methods such as pass@k, self-consistency, and verifier-based selection improve performance by considering multiple candidate solutions; during RL, methods such as GRPO (Guo et al., 2025) generate groups of responses to construct a learning signal. These procedures conventionally draw the rollouts *independently*. Independence is simple but wasteful: multiple samples can redundantly visit the same high-probability regions of the model distribution, spending additional compute without a proportional increase in coverage.

Instead, we could *couple* the rollouts in a way that better coordinates their exploration. Arithmetic sampling (Vilnis et al., 2023; Parashar et al., 2025) provides a mechanism for generating such correlated LM samples, but leaves open how the *coupling* itself should be designed. The randomly shifted lattice used by Vilnis et al. (2023); Parashar et al. (2025) is one of infinitely many joint distributions with uniform marginals. Choosing among these distributions is consequential: different sampling techniques can exhibit substantially different finite-sample behavior depending on the application (Owen, 2013). In this paper, we ask: *which couplings are most effective for inference, and which are most effective for learning?*

In particular, we initiate a systematic study of this design space for language models, asking how different couplings behave and which are best suited to different modern use cases. We build on the foundation laid out by arithmetic sampling (Vilnis et al., 2023; Parashar et al., 2025) and advance this line of research in two directions.

- We systematically explore the design space of couplings beyond the randomly shifted lattice used in prior work. We compare lattice, stratified, and Sobol sampling, developing a theoretical account of how different dependencies affect coverage and learning.
- To validate our theory, we use these samplers in test-time scaling and policy-gradient RL for LMs. In test-time scaling, we show these approaches match i.i.d. performance with

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

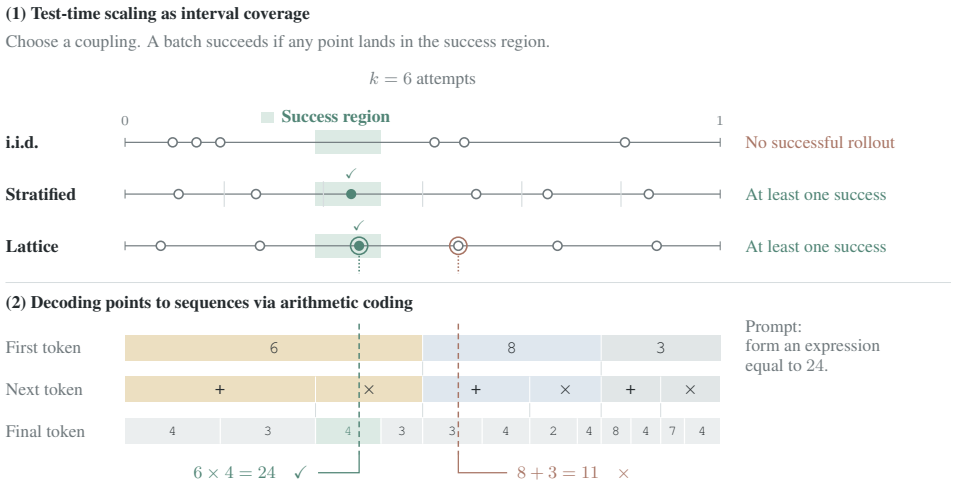

Figure 1: **Coupling as a design axis for scaling inference compute. (1)** Generating k candidate solutions can be viewed as placing k points in $[0, 1]$: the batch succeeds if any point falls in the shaded success region. Independent, stratified, and lattice sampling offer different ways to coordinate these attempts while preserving the language model's next-token distribution. This figure shows how the choice of coupling can affect the probability of finding a solution within a fixed sampling budget. **(2)** Arithmetic sampling (Vilnis et al., 2023; Parashar et al., 2025) maps uniforms to LM rollouts. The highlighted points produce one successful and one unsuccessful expression. We study which designs are most effective for test-time scaling and policy-gradient RL.

25–47% fewer inference samples. In RL, we show using coupled samplers outperforms i.i.d samplers.

2 Related Work

Arithmetic sampling Arithmetic sampling (Vilnis et al., 2023) introduced a method for generating correlated, marginally exact LM samples by mapping a randomly shifted lattice through an arithmetic decoder. Subsequent work (Parashar et al., 2025) applied this construction to multi-sample reasoning and minimum Bayes risk decoding. We build on this framework but study a different question: *how should the dependence between samples be designed?* We compare multiple coupling families, characterize their theoretical tradeoffs, and study their consequences for both test-time scaling and policy-gradient RL specifically for test-time scaling and RL. Randomized quasi-Monte Carlo (QMC) has been previously applied to policy-gradient and actor-critic estimation (Arnold et al., 2022), primarily using randomized Sobol constructions in continuous-control settings. In contrast, we study the choice of coupling itself—including lattice, stratified, and Sobol sampling—and develop an empirical and theoretical account of how this choice affects language-model inference and post-training.

Exploration in RL Recent approaches modify the RL or SFT training objective to induce broader coverage over rollouts (Tajwar et al., 2026; Hamid et al., 2026b; Orney et al., 2026; Chen et al., 2026; Hamid et al., 2026a). Our approach is complementary: rather than modifying the reward or training objective, we alter the joint distribution of rollout groups while preserving each rollout's marginal distribution under the policy.

3 Background

We briefly review the construction of Vilnis et al. (2023); Parashar et al. (2025). The construction is based on arithmetic coding (MacKay, 2002): starting from $[0, 1]$, we partition the interval into bins whose widths are the first-token probabilities $\pi_\theta(x_1)$, for $x_1 \in \mathcal{V}$. If u falls in the bin corresponding

to x_1, then x_1 is the first token. We then recursively partition this bin according to the conditional distribution $\pi_\theta(\cdot \mid x_1)$. Continuing in this way, every prefix $x_{1:t}$ is associated with an interval $I(x_{1:t}) \subseteq [0,1]$ whose length is

$$|I(x_{1:t})| = \pi_\theta(x_{1:t}).$$

Sampling a complete sequence amounts to drawing a single uniform point and following the unique chain of nested intervals containing it. Because the interval corresponding to each sequence has length equal to its probability under π_θ, the decoded sequence is distributed exactly according to the language model. See Figure 1 for details.

This representation is amenable to coupling the samples because it only requires uniform marginals. Any collection $(U_1, \ldots, U_k)$ with $U_i \sim \mathrm{Unif}[0,1]$ marginally can be decoded through the same arithmetic construction to produce a batch of correlated language model rollouts, while preserving the exact language model (marginal) distribution. Vilnis et al. (2023); Parashar et al. (2025) exploit this property using a randomly shifted lattice to generate diverse samples in parallel. Our work takes this arithmetic-sampling representation as a starting point; Section 4 studies the broader question of how the underlying uniform variables should be coupled for LM settings.

4 THE SPACE OF COUPLINGS

Arithmetic sampling reduces the problem of coupling rollouts to choosing a joint distribution over marginally uniform random variables (Section 3). However, this still leaves open a central design question: which couplings are most valuable in modern LM settings? To begin, we introduce several ways of coupling the randomness underlying autoregressive generation. All constructions below preserve the marginal distribution of every rollout exactly, but create different dependencies between the samples. We will study these implications on test-time scaling and RL in the next section.

Independent sampling. Standard i.i.d sampling does

$$U_i \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}[0,1], \qquad i = 1, \ldots, k.$$

The locations of different samples are completely unconstrained relative to one another, so points may cluster in the same region of the unit interval.

Stratified sampling. A simple way to coordinate samples while preserving randomness is to divide $[0,1]$ into k equal strata and draw exactly one point from each:

$$\widetilde{U}_i \sim \mathrm{Unif}\left[\frac{i-1}{k}, \frac{i}{k}\right], \qquad i = 1, \ldots, k.$$

We randomly permute the resulting points across rollout indices, so that every labeled U_i is marginally $\mathrm{Unif}[0,1]$. Stratification therefore prevents multiple points from occupying the same stratum, while leaving their locations within strata undetermined.

Randomly shifted lattice. A more rigid sequence-level coupling places the k points at exactly equal spacing,

$$U_i = \left(\Delta + \frac{i-1}{k}\right) \bmod 1, \qquad \Delta \sim \mathrm{Unif}[0,1].$$

This is the shifted-lattice construction used by arithmetic sampling (Vilnis et al., 2023). The shared random shift makes each U_i marginally uniform, but once one point is known the locations of all others are fixed. Relative to stratification, lattice sampling offers fewer additional degrees of freedom.

Token-level Sobol sampling. A T-dimensional Sobol sequence is an ordered infinite sequence of deterministic points

$$x_1, x_2, \ldots, \qquad x_i = (x_{i,1}, \ldots, x_{i,T}) \in [0,1)^T,$$

constructed so that finite prefixes are spread evenly throughout $[0,1)^T$ (i.e., have low discrepancy).

To generate k rollouts, we take the first k points of the Sobol sequence and stack them into the matrix

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,T} \\ \vdots & & \vdots \\ x_{k,1} & \cdots & x_{k,T} \end{pmatrix} \in [0,1)^{k \times T}.$$

Row i corresponds to rollout i, while column t corresponds to token position t.

To randomize the entire point set we use a shared Cranley–Patterson shift. Draw

$$\Delta = (\Delta_1, \ldots, \Delta_T) \sim \mathrm{Unif}([0,1)^T)$$

and define

$$U_{i,t} = (x_{i,t} + \Delta_t) \bmod 1.$$

The coordinate $U_{i,t}$ controls the inverse-CDF sampling decision for rollout i at token position t.

For every fixed i,

$$U_i = (U_{i,1}, \ldots, U_{i,T}) \sim \mathrm{Unif}([0,1)^T),$$

so each rollout has exactly the distribution required by ordinary autoregressive sampling. Across rollouts, however, the U_i remain coupled: they share the same random shift and therefore preserve the relative geometry of the underlying low-discrepancy Sobol points.

5 Theoretical analysis of couplings

In this section, we develop a theoretical understanding of when coupling can help and what distinguishes different couplings. Our main conclusion is that no single coupling is uniformly optimal: both the available headroom and the relative performance of different couplings depend on problem difficulty and the structure of the success region.

5.1 Setup

We focus our theoretical analysis on pass@k for binary-valued rewards. This quantity is practically important in test time scaling since it measures the benefit of drawing multiple candidate rollouts. For RL with binary rewards, pass@k is the probability that a group contains at least one successful rollout; this determines whether the model receives any gradient signal from a problem. Moreover, pass@k is amenable to theoretical analysis under arithmetic sampling because it corresponds to considering whether points land in a region on the unit interval.

Consider a fixed problem and a language model π_θ. Arithmetic sampling defines a deterministic map from a point $u \in [0,1]$ to a complete rollout. Let

$$A \subseteq [0,1]$$

denote the set of points that decode to a successful rollout; we call this set the *success region*. Define

$$p := \lambda(A),$$

where λ is the Lebesgue measure on $[0,1]$[1]. Under arithmetic sampling, p is exactly the probability that a single rollout succeeds (i.e., the problem's pass@1 probability).

5.2 How much can coupling help?

Intuitively, any sampler that respects the model's marginal distribution can only rearrange the model's capability across k rollouts. Before comparing couplings, we first ask how much room there is to improve beyond i.i.d sampling. Concretely, for a problem with pass@1 probability p, how much can pass@k improve, for *any marginal preserving sampler*, if we can only change the dependence among the k samples? We resolve this question below and visualize the upper bound in Figure 2.

[1]Note that, measurability of the success region follows immediately from the arithmetic coding construction: each rollout corresponds to an interval in $[0,1]$, so the success region is a countable union of intervals.

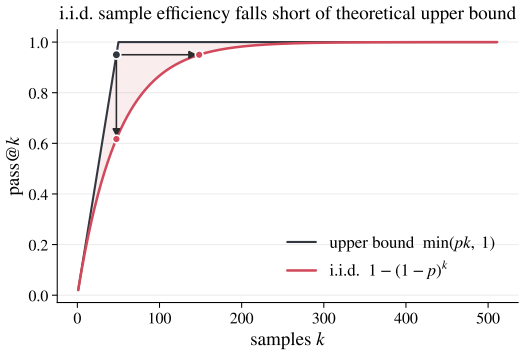

Figure 2: **i.i.d. sampling falls short of the per-problem pass@k upper bound.** For a problem with pass@1 probability p, the union bound gives a ceiling of $\min(pk, 1)$ (black) that places an upper bound on any sampler with target marginals. The i.i.d. curve $1 - (1 - p)^k$ (coral) lies strictly below this ceiling; the shaded gap represents the redundancy of independent sampling. Carefully constructed dependent samples can close the gap by being more spread out over the output space.

Theorem 1 (Per-problem pass@k lower and upper bound)**.** *Let p denote the probability that a single sample solves a fixed problem. Let $P(k)$ denote the probability that at least one of k samples solves the problem. Then*

$$p \leq P(k) \leq \min\{kp, 1\}.$$

We quantify the discrepancy between the upper bound and independent sampling for a problem with pass@1 probability p as:

$$\Delta_k(p) := P^*(k) - P_{\text{iid}}(k) = \min\{kp, 1\} - \left[1 - (1 - p)^k\right]. \tag{1}$$

Moreover, there is a Goldilocks zone of success as Proposition 1 makes precise. When $p \ll 1/k$, successes are so rare that i.i.d. batches contain little redundant success probability to eliminate. When $p \gg 1/k$, i.i.d. already achieves pass@k close to one. The largest gains therefore arise in the intermediate, hard-but-feasible regime $p \approx 1/k$.

Proposition 1 (Coupling helps most in the hard-but-feasible regime)**.** *For $k \geq 2$, the gap $\Delta_k(p)$ is maximized at*

$$p = \frac{1}{k}.$$

5.3 Couplings trade off robustness and optimality

The previous results characterize when there is room for dependence to improve pass@k. We now ask which dependence structure should be preferred. Our main conclusion is that *no single coupling is uniformly optimal*. Stratification provides a robust improvement over i.i.d. sampling regardless of the geometry of the success region, whereas lattice sampling can attain the optimal pass@k when the success region is geometrically favorable, but can perform substantially worse when that geometry is unfavorable. Thus, the relative advantage of different couplings depends not only on the marginal success probability p, but also on how successful rollouts are arranged under the decoding map. We now study the coverage of these different couplings under success regions with concrete geometries.

Stratification is uniformly robust, but not generally optimal. Stratification enjoys a strong worst-case guarantee: regardless of how the rollouts are arranged, it can never decrease pass@k relative to i.i.d. sampling. Note, these results are an adaptation of classic stratification results to the setting of arithmetic sampling. Importantly, as Proposition 3 shows, this robustness guarantee does not hold for all samplers.

Theorem 2 (Stratified sampling weakly dominates i.i.d. coverage)**.** *Let $A \subseteq [0, 1]$ be a measurable success region with $p = \lambda(A)$. Then*

$$P_{\text{strat}}(k; A) \geq P_{\text{iid}}(k; p)$$

Thus, stratified sampling is uniformly robust to the arrangement of the success region.

Lattice sampling is optimal for every contiguous success region. Lattice sampling enjoys a optimality property under additional structure on A that we make precise in Proposition 2.

Proposition 2 (Lattice attains the upper bound for contiguous success regions)**.** *Let $A \subseteq [0,1]$ be a single interval of measure $p = \lambda(A)$. Then a randomly shifted k-point lattice satisfies*

$$P_{\text{lattice}}(k; A) = \min\{kp, 1\}.$$

Consequently, lattice sampling saturates the pass@k ceiling from Theorem 1.

Importantly, stratified sampling *does not enjoy this guarantee* as we establish in Proposition 5 in the Appendix.

A worst case for both samplers. We now present a success region that attains the worst case for both stratified and lattice sampling, but exposes a sharp difference in their robustness: lattice performs *worse* than i.i.d. and stratified performs as well as i.i.d. sampling. We visualize this in Figure 6.

Proposition 3 (A periodic worst case for stratified and lattice sampling)**.** *Consider*

$$A = \bigcup_{j=0}^{k-1} \left[\frac{j}{k}, \frac{j}{k} + \frac{p}{k} \right), \qquad 0 < p < 1.$$

Then

$$P_{\text{strat}}(k; A) = P_{\text{iid}}(k; p),$$

while

$$P_{\text{lattice}}(k; A) = p < P_{\text{iid}}(k; p).$$

Both values attain the respective worst-case coverage for the method.

Finally, we present another result comparing the variance of the policy gradient estimator under stratified sampling against i.i.d sampling.

Theorem 3 (Stratified sampling weakly reduces policy-gradient variance)**.** *Fix a prompt x and policy π_θ, and let*

$$h(U) := r\big(x, D_\theta(U)\big) \nabla_\theta \log \pi_\theta\big(D_\theta(U) \mid x\big), \qquad U \sim \text{Unif}[0,1],$$

with $\mathbb{E}[\|h(U)\|^2] < \infty$.

Let $\widehat{g}_{\text{iid}}$ and $\widehat{g}_{\text{strat}}$ denote the policy-gradient estimators obtained by averaging k i.i.d. and stratified samples, respectively. Then both are unbiased for

$$g := \mathbb{E}[h(U)] = \nabla_\theta J_x(\theta),$$

and

$$\text{Cov}(\widehat{g}_{\text{strat}}) \preceq \text{Cov}(\widehat{g}_{\text{iid}}).$$

6 Empirical evaluation

Our theory identifies when coupling has room to help and characterizes how different couplings behave. We now empirically study what this implies in LM settings (test-time scaling and RL). We focus on four controlled reasoning tasks chosen to isolate the effect of the sampler. Each task is nontrivial at the evaluated rollout budgets, admits an exact verifier, and has a compact output space. This lets us cleanly study how the joint sampling schemes affects coverage.

Samplers. We draw k samples via the three approaches described in Section 4: lattice, stratified, and token-level Sobol. We benchmark against **i.i.d.** draws k independent samples from the model and a plug-in estimate of the **upper bound** that reflects the per-problem ceiling from Theorem 1. This estimates the best pass@k achievable by any sampler that preserves the LM's marginals.

Tasks. We evaluate on four verifier-based reasoning tasks spanning arithmetic search (**Countdown**; Gandhi et al., 2024; 2025; Pan et al., 2025), spatial planning (**Maze**; Tajwar et al., 2026), constraint satisfaction (**Sudoku**), and pattern induction (**1D-ARC**; Xu et al., 2024). See Appendix A for representative examples and solutions.

Models. For Countdown, Sudoku, and Maze, we fine-tune Qwen3.5-0.8B-Base on 50,000 correct question–answer pairs, since off-the-shelf models struggle with formatting and assign little mass to correct solutions. For 1D-ARC, we use Qwen3.5-2B-Base directly. ARC inputs are inductive by nature, so the base model already produces well-formatted outputs, and does not require additional supervised fine-tuning. We report additional results with 4B-9B models from the same Qwen3.5 family in Appendix A.2.

Evaluation. For each task we report pass@k (i.e. 1 if at least one of the k attempts is correct), and the plug-in estimate of the pass@k upper bound, averaged across all problems. For each problem, we estimate its pass@1 probability p from a separate set of 256 i.i.d. samples and compute the plug-in bound $\min\{k\hat{p}, 1\}$[2] To isolate the effect of the sampler, all methods share the same model, prompt, and decoding temperature; only the joint distribution of the underlying uniforms differs. We use a temperature of 1.0 and top-p 0.95 throughout, with a maximum completion length of 32 tokens for Countdown and Maze, and 280 tokens for Sudoku and 1D-ARC. We use a bootstrap approach that subsamples a batch of 64 responses to estimate pass@k for $k < 64$; we prove this procedure is unbiased in the appendix in Theorem 4.

6.1 Test-time scaling

The choice of coupling substantially affects pass@k. Figure 3 compares i.i.d., lattice, stratified, and token-level Sobol sampling while holding the model, prompt, decoding parameters, and one-rollout distribution fixed. All three dependent samplers improve over i.i.d. across the four tasks, but the magnitude of the gain depends strongly on the coupling. Among the samplers we study, the randomly shifted lattice consistently provides the strongest test-time coverage.

Lattice sampling approaches the upper bound. Our theory shows that lattice sampling makes a more aggressive tradeoff than stratification: its rigid spacing can attain the optimal coverage for "favorable" success geometries, but there are pathological geometries for which it performs provably worse than i.i.d. sampling. Empirically, we find the pass@k curves track the upper bound closely. This holds, to different degrees, across a diverse set of tasks with different solution geometries — arithmetic search (Countdown), spatial planning (Maze), constraint satisfaction (Sudoku), and pattern induction (1D-ARC). This behavior is consistent with the geometry of the success regions. For Sudoku and 1D-ARC, each problem has a single correct output sequence, so the success region is a single arithmetic-coding interval; Proposition 2 therefore applies directly and predicts that lattice sampling attains the optimal pass@k bound. Countdown, in contrast, generally admits multiple valid solution sequences, yielding more fragmented success regions for which this guarantee no longer holds. Correspondingly, lattice sampling is less dominant, although it still improves over i.i.d. sampling.

Better coverage translates into fewer rollouts. We can also ask how many fewer dependent samples are required to match a given i.i.d. pass rate; we present these results in Figure 4. The lattice-based sampler reaches i.i.d.'s performance with a 25-47% reduction in samples.

6.2 Policy-gradient based reinforcement learning

Table 1: Final-checkpoint pass@1 (%), mean ± SE over 3 training seeds. Best results bolded.

Sampling	Maze	Sudoku
i.i.d.	30.3 ± 0.3	20.0 ± 0.3
Stratified	32.4 ± 0.3	21.0 ± 0.8
Lattice	32.1 ± 0.2	$\mathbf{21.0 \pm 0.4}^*$
Sobol	$\mathbf{32.6 \pm 0.3}^*$	20.0 ± 0.4

[2]This estimator is consistent for the theoretical bound $\min\{kp, 1\}$, although it is downward biased at finite sample sizes due to the concavity of $\min\{kp, 1\}$.. Empirically, this finite-sample effect is negligible: increasing the estimation budget from 256 to 1024 samples produces essentially unchanged estimates.

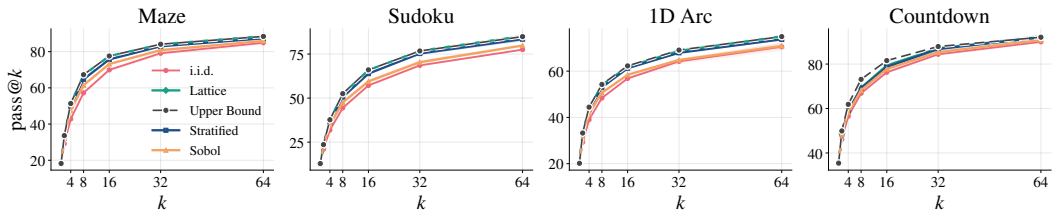

Figure 3: **pass@k analysis for all samplers.** Lattice closely tracks an empirical plug-in estimate of the pass@k upper bound. We plot error bars ± 1 SD but they are small enough that they are not visible in the plot above. Importantly, the upper bound curve aggregates over multiple problems, which is why it is not piecewise linear in k.

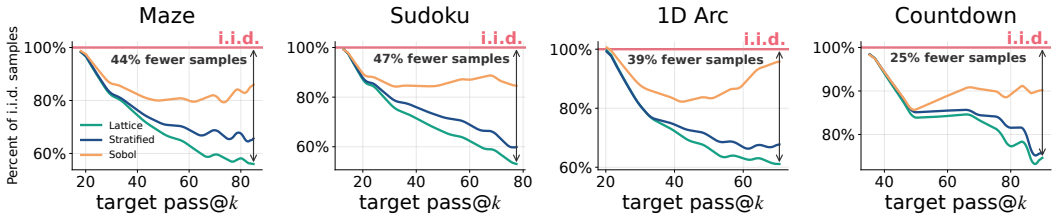

Figure 4: **Sample efficiency boosts from coupling samples.** Coupled samplers can use 25%-47% fewer samples than i.i.d. sampling while achieving the same accuracy. These performance gains hold across different methods although lattice performs best.

RL is a natural second setting for studying coupling samples because it also consumes groups of rollouts, and whether a group is useful is closely related to its coverage. For binary-reward group-relative methods such as GRPO, a rollout group provides a learning signal only when at least one member receives a reward. Thus, pass@k is closely related to whether a group is useful for learning.

Setup. We train Qwen3.5-0.8B-Base on Maze and Sudoku using GRPO (Guo et al., 2025), comparing i.i.d., lattice, stratified, and token-level Sobol sampling during training. We use a group size of $G = 8, 64$ unique prompts per gradient step, sampling temperature $\tau = 1.0$, and a maximum completion length of 256 tokens. We perform two gradient steps per rollout batch and use sequence-level off-policy importance sampling (Zheng et al., 2025). We report results averaged over three random training seeds. Experiments were run on 4 H100s/A100s.

Coupling boosts performance and reduces zero-variance groups. We plot the final checkpoint test set performance in Table 1. On Maze, all coupled samplers outperform i.i.d. by an amount that is well beyond the standard error (SE). We can understand these performance improvements by analyzing the training dynamics. In Figure 5, we see that the coupled samplers reduce the amount of zero-variance groups (averaged across training); zero variance group occur when all G rollouts receive the same reward—either all succeed or all fail. Lattice sampling produces the fewest zero-variance groups, consistent with our theory and the results in Section 6.1.

Dependence can bias group-relative updates. Unlike vanilla policy gradient, marginal correctness does not guarantee unbiasedness for GRPO because its group-mean baseline couples the rollouts. We quantify the magnitude of this effect empirically. Across dependent samplers, the RMS displacement of the mean gradient from the i.i.d. reference is only 0.17–0.20 times the RMS noise of an i.i.d. mean-gradient estimate (Table 4). We characterize this effect theoretically for stratified sampling in the Appendix (A.4), revealing a tradeoff between bias and coverage.[3]

[3]Our experiments use a mean-centered GRPO, but the theory relies on RLOO. In this case, the GRPO update is exactly a fixed rescaling by $(G-1)/G$ of the corresponding RLOO update. Thus, the dependence effects characterized for RLOO apply directly.

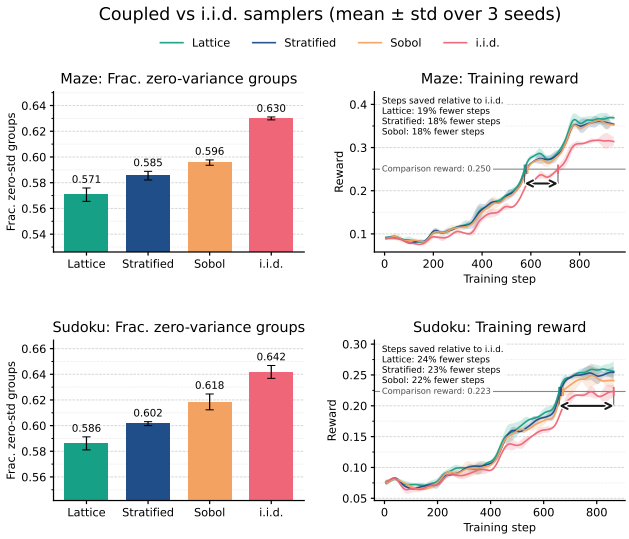

Figure 5: **RL training dynamics**. **(Left)** Dependent sampling consistently produces fewer zero-variance groups (i.e., groups that contribute no learning signal) throughout training because it samples responses with higher coverage. The ranking of samplers by percent of zero-variance is broadly consistent with our theory in Section 5.3. **(Right)** The training reward of coupled samplers also rises faster. We average results over 3 training seeds and report standard error.

6.3 Code generation

In Table 3 in the Appendix, we evaluate our coupled samplers on two code tasks, MBPP (Austin et al., 2021) (sanitized version, 7-shot) and HumanEval (Chen et al., 2021) (zero-shot), using the Llama-3.2-3B base model, temperature 0.7, top-p 0.95, and $K = 128$ samples per problem. Among the 257 MBPP (164 HumanEval) test problems, many are either almost never solved or almost always solved; these are regimes where our theory predicts little room for any marginal-preserving sampler to improve over i.i.d. sampling. Thus, we report both aggregate results and results on the non-saturated subset of problems, defined as problems where an independent run with $k = 1024$ achieves a pass rate strictly between 0 and 0.5. We also report the unique solutions as measured by exact match. The lattice sampler improves aggregate pass@128 by 2.08 points and average unique solutions by 4.97 on MBPP (pass@8 by 2.12 points and unique solutions by 5.17 on HumanEval). On the non-saturated problems, where there is meaningful headroom for the sampler to matter (see Theorem 1), the pass@128 gain increases to 3.70 points; on HumanEval the pass@8 gain increases to 2.79 points.

7 Conclusion

We study coupling as a design choice for generating batches of language-model rollouts. Building on arithmetic sampling, we show that marginal-preserving couplings trade off robustness and optimality. Across four reasoning tasks, coupled samplers consistently improve coverage over i.i.d. sampling, with lattice sampling matching i.i.d. pass@k using 25–47% fewer rollouts. In reinforcement learning, coupling similarly reduces zero-variance groups and improves sample efficiency. More broadly, our results identify the joint distribution of rollouts as an underexplored axis for scaling inference and learning. The benefits depend on the geometry induced by decoding: compact success regions favor stronger couplings, while fragmented regions can weaken their gains. This motivates designing samplers that adapt their dependence structure to the underlying generation problem.

AI USE STATEMENT

In this work, we used generative AI tools for help developing theoretical models or conceptual frameworks, coding assistance, implement methods, writing assistance, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs. We have not used generative AI tools for generating synthetic data sets, cleaning and reformating dataset, proposing or refine hypotheses, assisting with translation, designing or providing feedback on research methodology or experiments. The rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism and LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Sebastien M. R. Arnold, Pierre L'Ecuyer, Liyu Chen, Yi fan Chen, and Fei Sha. Policy learning and evaluation with randomized quasi-monte carlo, 2022. URL https://arxiv.org/abs/2202.07808.

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

Feng Chen, Allan Raventos, Nan Cheng, Surya Ganguli, and Shaul Druckmann. Rethinking fine-tuning when scaling test-time compute: Limiting confidence improves mathematical reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

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

Alek Dimitriev and Mingyuan Zhou. CARMS: Categorical-antithetic-REINFORCE multi-sample gradient estimator. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021.

Kanishk Gandhi, Denise H J Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah Goodman. Stream of search (sos): Learning to search in language. In *First Conference on Language Modeling*, 2024.

Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*, 2025.

Daya Guo et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, pp. 633–638, 2025.

Jubayer Ibn Hamid, Ifdita Hasan Orney, Michael Y. Li, Omar Shaikh, Yoonho Lee, Dorsa Sadigh, Chelsea Finn, and Noah Goodman. Spiral: Learning to search and aggregate, 2026a. URL https://arxiv.org/abs/2606.23595.

Jubayer Ibn Hamid, Ifdita Hasan Orney, Ellen Xu, Chelsea Finn, and Dorsa Sadigh. Polychromic objectives for reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026b.

David J. C. MacKay. Information theory, inference & learning algorithms, 2002.

Ifdita Hasan Orney, Jubayer Ibn Hamid, Shreya S Ramanujam, Shirley Wu, Hengyuan Hu, Noah Goodman, Dorsa Sadigh, and Chelsea Finn. Poly-epo: Training exploratory reasoning models, 2026.

Art B. Owen. *Monte Carlo Theory, Methods and Examples*. Art Owen, 2013.

Jiayi Pan, Xiuyu Li, Long Lian, Charlie Victor Snell, Yifei Zhou, Adam Yala, Trevor Darrell, Kurt Keutzer, and Alane Suhr. Learning adaptive parallel reasoning with language models. In *Second Conference on Language Modeling*, 2025.

Aditya Parashar, Aditya Vikram Singh, Avinash Amballa, Jinlin Lai, and Benjamin Rozonoyer. Quasi-random multi-sample inference for large language models, 2025.

Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning, 2026.

Luke Vilnis, Yury Zemlyanskiy, Patrick Murray, Alexandre Passos, and Sumit Sanghai. Arithmetic sampling: parallel diverse decoding for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Yudong Xu, Wenhao Li, Pashootan Vaezipoor, Scott Sanner, and Elias B. Khalil. Llms and the abstraction and reasoning corpus: Successes, failures, and the importance of object-based representations, 2024.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025.

A Appendix

A.1 Supporting figures for methods

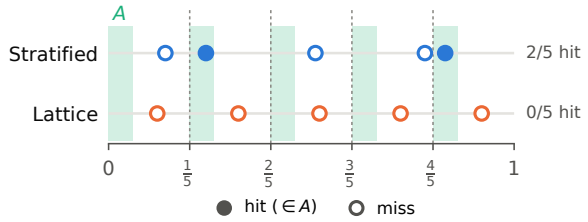

Figure 6: **Visualizing worst case success region**

Maze

```
*********
*E..*...*
*.***.***
*.......*
***.*.***
*...*...*
*S*.*****
*.*.....*
*********

URRUULLUU
```

Countdown

```
Target: 89
Numbers: [57, 43, 11]

43+57-11
```

Sudoku

```
8 4 5 2 _ _ 1 3 7
2 _ 9 _ 5 3 _ _ 6
6 _ _ _ _ _ _ _ 5
7 5 _ 9 8 6 _ 1 3
1 _ _ _ 4 _ _ _ 2
4 _ 6 _ _ _ _ _ 8
_ 2 _ _ 3 _ _ 8 9
_ 8 _ _ _ 1 6 2 _
9 6 _ 8 7 _ 3 5 1

8 4 5 2 6 9 1 3 7
2 1 9 7 5 3 8 4 6
6 7 3 4 1 8 2 9 5
7 5 2 9 8 6 4 1 3
1 9 8 3 4 7 5 6 2
4 3 6 1 2 5 9 7 8
5 2 1 6 3 4 7 8 9
3 8 7 5 9 1 6 2 4
9 6 4 8 7 2 3 5 1
```

1D-ARC

```
Input: [0,0,5,0,0,0,0,0,0,0,0,5,0,0]
Output: [0,0,5,5,5,5,5,5,5,5,5,5,0,0]

Input: [0,0,7,0,0,0,0,0,0,0,0,7,0,0]
Output: [0,0,7,7,7,7,7,7,7,7,7,7,0,0]

Input: [2,0,0,0,0,0,0,0,0,2,0,0,0,0]
Output: [2,2,2,2,2,2,2,2,2,2,0,0,0,0]

Input: [7,0,0,0,7,0,0,0,0,0,0,0,0,0]
Output:

[7,7,7,7,7,0,0,0,0,0,0,0,0,0]
```

Figure 7: Representative examples from each of the four tasks.

Table 2: Hyperparameters for Maze and Sudoku experiments.

Hyperparameter	**Value**
Optimizer	AdamW
Learning rate	1×10^{-6}
(β_1, β_2)	$(0.9,\ 0.95)$
ϵ_{Adam}	1×10^{-8}
Weight decay	0.01
Max gradient norm	1.0
KL coefficient (β)	0.01
PPO clip range (ϵ)	0.2
Off-policy epochs	2

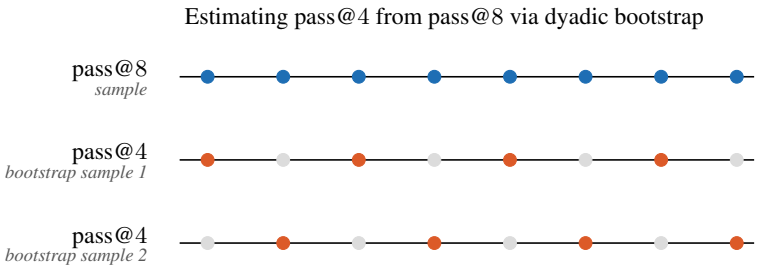

Figure 8: **Bootstrapped QMC pass@k estimator.** Because the k samples are dependent rather than i.i.d. draws, the standard pass@k estimator is invalid; instead, we exploit that any stride-2 subsample is itself a valid pass@$k/2$ lattice. Stride 2 admits two starting offsets, each yielding an unbiased pass@4 estimate from the same pass@8 rollout. This generalizes to larger strides.

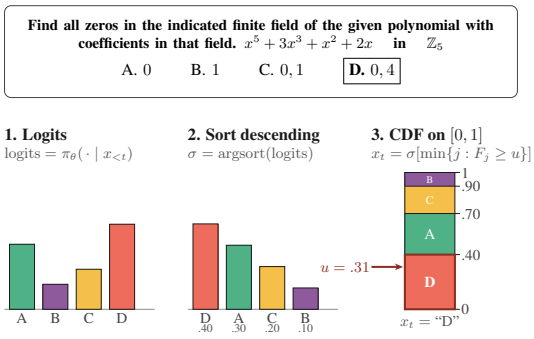

Figure 9: **Exact sampling from the LM using inverse-CDF sampling.** The LM produces logits over the answer choices. We sort them in descending order to yield a permutation σ and a sorted probability vector (panel 2). Stacking these probabilities partitions the unit interval into bins whose widths equal the token probabilities; a uniformly random point lands in each bin with probability equal to that token's mass. To sample a token, we first sample $u \in [0, 1)$ and find the token whose interval contains it. $u = .31$ falls in $[0, .40)$, returning $x_t =$ "D" (panel 3). For illustrative purposes, we show decoding a single token, but repeating this process yields an entire sequence. This recovers the same distribution as standard autoregressive sampling, which usually uses Gumbel-max.

A.2 ADDITIONAL EXPERIMENTAL RESULTS

(a) Coverage

	MBPP (7-shot, pass@128)		**HumanEval** (pass@8)	
Sampler	**Non-sat. (135/257)**	**All (257/257)**	**Non-sat. (125/164)**	**All (164/164)**
i.i.d.	84.69 ± 0.89	74.19 ± 0.57	35.68 ± 0.06	40.05 ± 0.08
Sobol	86.67 ± 0.43	75.10 ± 0.23	38.13 ± 0.37	41.90 ± 0.29
Stratified	87.41 ± 1.13	75.88 ± 0.59	36.96 ± 0.09	40.99 ± 0.06
Lattice	**88.40** ± 1.08	**76.27** ± 0.67	**38.47** ± 0.13	**42.16** ± 0.06
Improvement (lattice)	**+3.70 pp** ± 1.40	**+2.08 pp** ± 0.88	**+2.79 pp** ± 0.15	**+2.12 pp** ± 0.10

(b) Diversity (unique completions per problem, out of $K = 128$)

Sampler	**MBPP** (7-shot, 257/257)	**HumanEval** (164/164)
i.i.d.	111.07 ± 0.06	100.21 ± 0.12
Sobol	111.97 ± 0.08	101.96 ± 0.10
Stratified	115.11 ± 0.02	104.59 ± 0.03
Lattice	**116.05** ± 0.03	**105.38** ± 0.07
Improvement (lattice)	**+4.97** ± 0.07	**+5.17** ± 0.14

Table 3: **Coupling improves code-generation coverage and diversity on MBPP and HumanEval.** Llama-3.2-3B, $K = 128$ samples per problem, ± standard error over three seeds. Coverage is reported on all problems and on the non-saturated subset, which contains problems whose upper-bound per-problem pass@1 lies strictly between 0 and 0.5, estimated from an independent run with 1024 samples per problem. Diversity is reported on all problems.

Sampler	$\|\bar{g}_m - \bar{g}_{\text{iid}}\|^2$	$\operatorname{tr} \widehat{\operatorname{Cov}}(\bar{g}_m)$
i.i.d.	0.0000	0.082
Lattice	0.0011	0.031
Stratified	0.0007	0.041
Sobol	0.0011	0.033

Table 4: Empirical mean-gradient displacement and covariance trace under dependent sampling, estimated using a batch of 32 prompts × 8 rollouts per prompt, and averaged over 256 trials. Let $\bar{g}_m$ denote the empirical mean gradient under sampler m. The non-i.i.d. samplers' squared mean shifts are only 0.9–1.4% of the i.i.d. estimator's inherent stochastic variation, $\operatorname{tr} \widehat{\operatorname{Cov}}(\bar{g}_{\text{iid}}) = 0.082$. All dependent samplers also reduce gradient variance relative to i.i.d. sampling, with covariance traces of 0.031–0.041 versus 0.082.

Sampling Model	**Countdown (16-shot)** Qwen 4B base	**Maze** Qwen 4B SFT	**Sudoku** Qwen 4B SFT	**1D-ARC** Qwen 9B base
i.i.d.	68.7 ± 0.6	51.4 ± 0.8	84.9 ± 0.2	89.6 ± 0.2
Stratified	70.5 ± 0.6	58.1 ± 0.6	89.5 ± 0.6	90.9 ± 0.1
Sobol	69.7 ± 0.4	53.7 ± 0.7	85.1 ± 0.4	89.5 ± 0.2
Lattice	$\mathbf{71.0 \pm 0.3}$	$\mathbf{59.7 \pm 0.7}$	$\mathbf{90.3 \pm 0.1}$	$\mathbf{91.4 \pm 0.0}$
UB	71.7	60.5	90.4	91.9

Table 5: pass@64 (%) by sampling method on larger 4B-9B scale models. Mean ± SEM over 3 evaluation seeds; best mean per column in bold. UB is the upper bound on pass@64.

A.3 THEORETICAL ANALYSES

Theorem 1 (Per-problem pass@k lower and upper bound)**.** *Let p denote the probability that a single sample solves a fixed problem. Let $P(k)$ denote the probability that at least one of k samples solves the problem. Then*

$$p \leq P(k) \leq \min\{kp, 1\}.$$

Proof. Let S_j be the event that the j-th sample solves the problem. By supposition, $\Pr(S_j) = p$ for each j. The event that the problem is solved by at least one of the k samples is given by $\bigcup_{j=1}^{k} S_j$ and by the union bound,

$$P(k) = \Pr\left(\bigcup_{j=1}^{k} S_j\right) \leq \sum_{j=1}^{k} \Pr(S_j) = kp.$$

We also must have $P(k) \leq 1$. Combining the two bounds gives the desired upper bound. The lower bound follows from the union bound as well, from the trivial observation that, for any S_j, $S_j \subset \cup_{j=1}^{k} S_j$. □

Proposition 1 (Coupling helps most in the hard-but-feasible regime)**.** *For $k \geq 2$, the gap $\Delta_k(p)$ is maximized at*

$$p = \frac{1}{k}.$$

Proof. For $p \leq 1/k$, the optimum is $P^*(k) = kp$, so

$$\Delta_k(p) = kp - 1 + (1-p)^k.$$

Differentiating gives

$$\Delta_k'(p) = k\left[1 - (1-p)^{k-1}\right] \geq 0,$$

with strict inequality for $p > 0$. Thus the gap increases on $[0, 1/k]$.

For $p \geq 1/k$, the bound saturates at 1 and therefore we have

$$\Delta_k(p) = (1-p)^k,$$

so

$$\Delta_k'(p) = -k(1-p)^{k-1} \leq 0.$$

The gap decreases on $[1/k, 1]$ and is maximized at $p = 1/k$. □

Proposition 4 (Pairwise mutual information of sequence-level samplers)**.** *For any two distinct samples U_i, U_j,*

$$I(U_i; U_j) = \begin{cases} 0, & \textit{i.i.d.}, \\ \log \dfrac{k}{k-1}, & \textit{stratified}, \\ \infty, & \textit{lattice}. \end{cases}$$

Proof. For i.i.d. sampling, U_i and U_j are independent by construction, so $I(U_i; U_j) = 0$.

For stratified sampling, observing U_j means that no other point occupy the same stratum. Therefore, $U_i \mid U_j$ is uniform over the remaining $k-1$ strata, whose total measure is $\frac{k-1}{k}$. This random variable can be interpreted as a uniform variable with support $[0, \frac{k-1}{k}]$ Therefore,

$$H(U_i \mid U_j) = \log \frac{k-1}{k},$$

and

$$I(U_i; U_j) = -\log \frac{k-1}{k} = \log \frac{k}{k-1}.$$

For lattice sampling, observing one point determines every other point.

Let

$$P := P_{U_i,U_j}, \qquad Q := P_{U_i} \otimes P_{U_j}.$$

The joint distribution P is supported on a one-dimensional subset of $[0,1]^2$.

$$G = \left\{(u,v) : v = \left(u + \frac{j-i}{k}\right) \bmod 1\right\}.$$

Therefore, P takes full measure on this set

$$P(G) = 1.$$

On the other hand, since both marginals are continuous uniform variables, the product measure assigns zero mass to this one-dimensional set:

$$Q(G) = 0.$$

Therefore $P \not\ll Q$, and by the definition of KL divergence,

$$I(U_i; U_j) = \infty.$$

□

Theorem 2 (Stratified sampling weakly dominates i.i.d. coverage)**.** *Let $A \subseteq [0,1]$ be a measurable success region with $p = \lambda(A)$. Then*

$$P_{\text{strat}}(k; A) \geq P_{\text{iid}}(k; p)$$

Proof. Let the j-th of k total equally-sized stratum be denoted as

$$I_j = \left[\frac{j-1}{k}, \frac{j}{k}\right), \qquad j = 1, \ldots, k,$$

and define the probability of success when sampling from stratum j as

$$p_j := \Pr(U \in A \mid U \in I_j)$$

For stratified sampling, we have that

$$P_{\text{strat}}(k) = 1 - \prod_{j=1}^{k}(1 - p_j). \tag{2}$$

To finish, we make a trivial observation: a marginally uniform draw can be generated by first choosing a stratum uniformly at random and then sampling within that stratum. By the law of total probability, if p is the marginal probability of success we have

$$\frac{1}{k}\sum_{j=1}^{k} p_j = p. \tag{3}$$

Applying the AM-GM inequality and the characterization above gives

$$\left(\prod_{j=1}^{k}(1 - p_j)\right)^{1/k} \leq \frac{1}{k}\sum_{j=1}^{k}(1 - p_j) = 1 - p.$$

Therefore

$$\prod_{j=1}^{k}(1 - p_j) \leq (1-p)^k,$$

giving

$$P_{\text{strat}}(k) \geq 1 - (1-p)^k = P_{\text{iid}}(k).$$

□

Proposition 2 (Lattice attains the upper bound for contiguous success regions)**.** *Let $A \subseteq [0,1]$ be a single interval of measure $p = \lambda(A)$. Then a randomly shifted k-point lattice satisfies*

$$P_{\text{lattice}}(k; A) = \min\{kp, 1\}.$$

Consequently, lattice sampling saturates the pass@k ceiling from Theorem 1.

Proof. The result follows directly from the fixed spacing of the lattice. When $p \leq 1/k$, the interval A cannot contain two points simultaneously because any two lattice points are separated by at least $1/k$. The k success events are therefore disjoint, and

$$P_{\text{lattice}}(k; A) = \sum_{i=1}^{k} \Pr(U_i \in A) = kp.$$

When $p \geq 1/k$, a complementary picture holds. The gaps between adjacent lattice points have length exactly $1/k$, so an interval of length at least $1/k$ cannot avoid all lattice points. Therefore,

$$P_{\text{lattice}}(k; A) = 1.$$

Combining the two regimes gives

$$P_{\text{lattice}}(k; A) = \min\{kp, 1\}.$$

□

Proposition 5 (Stratified sampling need not attain optimal coverage)**.** *Let $k \geq 2$. For any $0 < p < 1/k$, there exists a contiguous interval $A \subset [0,1)$ of Lebesgue measure p such that*

$$P_{\text{strat}}(k; A) < P^*(k) = kp.$$

In particular, fixed stratification does not uniformly attain the optimal coverage bound, even over contiguous success regions.

Proof. Let A straddle the boundary between two adjacent strata, and let $a, b > 0$ denote the lengths of the portions of A lying in the left and right strata, respectively, so that $a + b = p$.

The samples from these two strata fall in A with probabilities ka and kb, respectively. Since the draws from distinct strata are independent,

$$\begin{aligned} P_{\text{strat}}(k; A) &= 1 - (1 - ka)(1 - kb) \\ &= k(a + b) - k^2 ab \\ &= kp - k^2 ab \\ &< kp = P^*(k), \end{aligned}$$

where the strict inequality follows from $a, b > 0$. □

Proposition 3 (A periodic worst case for stratified and lattice sampling)**.** *Consider*

$$A = \bigcup_{j=0}^{k-1} \left[\frac{j}{k}, \frac{j}{k} + \frac{p}{k} \right), \qquad 0 < p < 1.$$

Then

$$P_{\text{strat}}(k; A) = P_{\text{iid}}(k; p),$$

while

$$P_{\text{lattice}}(k; A) = p < P_{\text{iid}}(k; p).$$

Both values attain the respective worst-case coverage for the method.

Proof. We first consider the lattice sampler. Recall that the lattice points themselves are separated by exactly $1/k$. Therefore, if a given point lies in one of the success region, every other point occupies the same relative position within one of the size $\frac{p}{k}$ success regions. This implies that

$$U_1 \in A \quad \Longleftrightarrow \quad U_2 \in A \quad \Longleftrightarrow \quad \cdots \quad \Longleftrightarrow \quad U_k \in A.$$

Thus, either every point succeeds or every point fails. Concretely, a point lies in A with probability p, so the probability that the batch contains any success is given exactly by

$$P_{\text{lattice}}(k; A) = p.$$

For $0 < p < 1$ and $k > 1$,

$$1 - (1 - p)^k > p,$$

so in this geometry, lattice sampling performs strictly worse than independent sampling.

The same construction also attains the worst case for stratified sampling. Each stratum contains the same fraction p of successful mass, so the per-stratum success probabilities satisfy $p_j = p$ for all j. The proof of Theorem 2 applies AM–GM to the quantities $1 - p_j$, with equality exactly when the p_j are equal. Therefore, for this example, the stratified sampler has the same performance as i.i.d sampling. □

Theorem 3 (Stratified sampling weakly reduces policy-gradient variance)**.** *Fix a prompt x and policy π_θ, and let*

$$h(U) := r\big(x, D_\theta(U)\big)\nabla_\theta \log \pi_\theta\big(D_\theta(U) \mid x\big), \qquad U \sim \text{Unif}[0, 1],$$

with $\mathbb{E}[\|h(U)\|^2] < \infty$.

Let $\widehat{g}_{\text{iid}}$ and $\widehat{g}_{\text{strat}}$ denote the policy-gradient estimators obtained by averaging k i.i.d. and stratified samples, respectively. Then both are unbiased for

$$g := \mathbb{E}[h(U)] = \nabla_\theta J_x(\theta),$$

and

$$\text{Cov}(\widehat{g}_{\text{strat}}) \preceq \text{Cov}(\widehat{g}_{\text{iid}}).$$

Proof. The unbiasedness of $\widehat{g}_{\text{iid}}$ follows immediately from linearity of expectation. For $\widehat{g}_{\text{strat}}$, randomly permuting the stratified points across the batch makes each labeled sample marginally uniform on $[0, 1]$, so the same argument applies. In fact, the permutation is not required for unbiasedness of the average. By the law of total expectation,

$$\frac{1}{k}\sum_{j=1}^{k} \mathbb{E}[h(U) \mid Z = j] = \mathbb{E}[h(U)] = g.$$

Applying the law of total covariance while conditioning on the stratum index Z gives

$$\text{Cov}(h(U)) = \mathbb{E}[\text{Cov}(h(U) \mid Z)] + \text{Cov}(\mathbb{E}[h(U) \mid Z]).$$

Since $\widehat{g}_{\text{iid}}$ averages k independent copies of $h(U)$,

$$\begin{aligned}\text{Cov}(\widehat{g}_{\text{iid}}) &= \frac{1}{k}\,\text{Cov}(h(U)) \\ &= \frac{1}{k}\mathbb{E}[\text{Cov}(h(U) \mid Z)] + \frac{1}{k}\,\text{Cov}(\mathbb{E}[h(U) \mid Z]).\end{aligned}$$

For the stratified estimator, we have

$$\begin{aligned}\text{Cov}(\widehat{g}_{\text{strat}}) &= \text{Cov}\left(\frac{1}{k}\sum_{j=1}^{k} h(\widetilde{U}_j)\right) \\ &= \frac{1}{k^2}\sum_{j=1}^{k} \text{Cov}(h(U) \mid Z = j) \\ &= \frac{1}{k}\mathbb{E}[\text{Cov}(h(U) \mid Z)].\end{aligned}$$

The second equality follows from independence across strata, and the last is simply the definition of the expectation over the stratum index Z. Subtracting, we have

$$\text{Cov}(\widehat{g}_{\text{iid}}) - \text{Cov}(\widehat{g}_{\text{strat}}) = \frac{1}{k}\,\text{Cov}(\mathbb{E}[h(U) \mid Z]).$$

The RHS is a positive-semidefinite matrix so the result immediately follows.

□

Theorem 4. *Correctness of dyadic bootstrap sampling. Let $k = 2^L$, and let*

$$u_i = \left(\Delta + \frac{i}{k}\right) \bmod 1, \qquad i = 0, \dots, k-1,$$

where $\Delta \sim \mathrm{Unif}[0,1]$. For any $x \le L$, set $m = k/2^x$. Then each stride-2^x subsequence

$$u_j^{(r)} := u_{r+2^x j}, \qquad j = 0, \dots, m-1,$$

with $r \in \{0, \dots, 2^x - 1\}$, is distributed exactly as a fresh randomly shifted lattice of size m. Consequently, each such subsequence gives an unbiased pass@m estimate under the randomized-lattice construction.

Proof. Let $s = 2^x$, so $m = k/s$. For any offset r,

$$u_{r+sj} = \left(\Delta + \frac{r+sj}{k}\right) \bmod 1 = \left(\Delta + \frac{r}{k} + \frac{j}{m}\right) \bmod 1.$$

The last equality substitutes $\frac{1}{m}$ for $\frac{s}{k}$. Define

$$\Delta^{(r)} := \left(\Delta + \frac{r}{k}\right) \bmod 1.$$

For a fixed lattice subsequence, r and k, are constants, so $\frac{r}{k}$ is a deterministic shift. Since adding a deterministic constant modulo 1 preserves the uniform distribution $\Delta^{(r)}$ is also uniform on $[0,1]$. Finally substituting $\Delta^{(r)}$ and noting that adding a constant before reducing modulo 1 is the same as first reducing the shifted quantity modulo 1, we obtain

$$u_{r+sj} = \left(\Delta^{(r)} + \frac{j}{m}\right) \bmod 1,$$

which is exactly an m-point lattice with a uniform random shift. Therefore any pass@m estimator computed on this subsequence has the same expectation as the estimator computed on a fresh m-point randomized lattice. □

Theorem 5 (Bootstrap estimator for stratified sampling)**.** *Following the notation above, let $\{u_i\}_{i=0}^{k-1}$ be a set of stratified samples*

$$u_i \sim \mathrm{Unif}[\tfrac{i}{k}, \tfrac{i+1}{k}], \qquad i = 0, \dots, k-1.$$

Then, to obtain a bootstrapped stratified sample for $m = k/2^x$, we can partition u_i's into contiguous blocks of size $k/m = 2^x$

$$\{u_0, \dots, u_{k/m-1}\} \cup \{u_{k/m}, \dots, u_{2k/m-1}\} \cup \cdots \cup \{u_{(m-1)k/m}, \dots, u_{k-1}\}$$

and sample once from each partition

$$u'_i \sim \{u_{ik/m}, \dots, u_{(i+1)k/m-1}\}$$

so that $\{u'_i\}_{i=0}^{m-1}$ is distributed according to the stratified rule for m samples, i.e., $u_i \sim \mathrm{Unif}[\frac{i}{m}, \frac{i+1}{m}]$.

Proof. Let $q = k/m = 2^x$. For each $i = 0, \dots, m-1$, the i-th block contains the q adjacent strata

$$\left[\frac{iq}{k}, \frac{iq+1}{k}\right], \dots, \left[\frac{(i+1)q-1}{k}, \frac{(i+1)q}{k}\right],$$

whose union is exactly

$$\left[\frac{i}{m}, \frac{i+1}{m}\right].$$

Since all q strata have equal width, choosing one of their samples uniformly at random is equivalent to sampling uniformly from their union. Therefore,

$$u'_i \sim \mathrm{Unif}\left[\frac{i}{m}, \frac{i+1}{m}\right].$$

Applying this independently to each block gives exactly an m-sample stratified draw. □

Theorem 6 (Unbiased prefix estimator for randomized Sobol sampling)**.** *Fix a prompt, policy, and decoding rule with maximum rollout length T. Let $x_1, x_2, \ldots \in [0,1)^T$ be a fixed Sobol sequence. A batch of size n is generated using*

$$U_i = (x_i + \Delta) \bmod 1, \qquad i = 1, \ldots, n, \qquad \Delta \sim \mathrm{Unif}([0,1)^T).$$

Let $P_{\mathrm{Sobol}}(m)$ denote the probability that at least one rollout succeeds in a batch of size m.

For $m \le K$, generate a batch of size K and define

$$\widehat{P}_m := \mathbf{1}\{\textit{at least one of its first } m \textit{ rollouts succeeds}\}.$$

Then

$$\mathbb{E}[\widehat{P}_m] = P_{\mathrm{Sobol}}(m).$$

Proof. Compare a size-K batch and a size-m batch generated using the same random shift Δ. Both use the same Sobol points $x_1, \ldots, x_m$ for their first m rows, so these rows are identical:

$$U_i = (x_i + \Delta) \bmod 1, \qquad i = 1, \ldots, m.$$

Their first m decoded rollouts are therefore also identical. Consequently, the first m rollouts of the size-K batch contain a success exactly when the size-m batch contains a success. This holds for every Δ, so taking probabilities over the random shift gives

$$\mathbb{E}[\widehat{P}_m] = \Pr(\text{a size-}m \text{ batch contains a success}) = P_{\mathrm{Sobol}}(m).$$

□

A.4 DEPENDENT SAMPLING AND GROUP-RELATIVE POLICY GRADIENTS

For a separable estimator of the form

$$\widehat{g} = \frac{1}{n}\sum_{i=1}^{n} h(Y_i),$$

preserving the marginal distribution of every rollout is sufficient for unbiasedness. This is a simple consequence of linearity of expectation. We now characterize exactly how an arbitrary dependent sampler changes its expectation. CARMS (Dimitriev & Zhou, 2021) constructs unbiased REINFORCE estimators from negatively correlated categorical samples by correcting for their pairwise joint distribution but their approach requires marginalizing over all possible orderings which is intractable for language modeling.

Note that, although our experiments use a mean-centered GRPO, the theoretical results rely on RLOO. In this case, the GRPO update is exactly $(G-1)/G$ times the corresponding RLOO update. Thus, the dependence effects characterized for RLOO apply directly, up to a fixed rescaling.

Proposition 6 (Pairwise U-statistic representation of RLOO)**.** *Let $Y_1, \dots, Y_n$ be sampled trajectories from a policy π_θ, let $r(Y_i) \in \mathbb{R}$ denote their rewards, and define*

$$s_\theta(Y_i) := \nabla_\theta \log \pi_\theta(Y_i).$$

Define the symmetric kernel

$$\psi_\theta(y, y') := \frac{1}{2}\big(r(y) - r(y')\big)\big(s_\theta(y) - s_\theta(y')\big).$$

Then the RLOO estimator

$$\widehat{g}_{\mathrm{RLOO}} = \frac{1}{n}\sum_{i=1}^{n}\left(r(Y_i) - \frac{1}{n-1}\sum_{j\neq i} r(Y_j)\right) s_\theta(Y_i)$$

can be written as

$$\widehat{g}_{\mathrm{RLOO}} = \binom{n}{2}^{-1}\sum_{i<j}\psi_\theta(Y_i, Y_j).$$

Thus, under i.i.d. sampling, RLOO is a second-order U-statistic.

Proof. Expanding the leave-one-out estimator,

$$\widehat{g}_{\mathrm{RLOO}} = \frac{1}{n}\sum_{i=1}^{n} r(Y_i)s_\theta(Y_i) - \frac{1}{n(n-1)}\sum_{i\neq j} r(Y_j)s_\theta(Y_i)$$

We first make the left and right term symmetric by introducing dummy indices that allow us to re-write the first term as a pairwise sum:

$$\frac{1}{n}\sum_{i=1}^{n} r(y_i)s_\theta(y_i) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n-1}\sum_{j\neq i} r(y_i)s_\theta(y_i) = \frac{1}{n(n-1)}\sum_{i\neq j} r(y_i)s_\theta(y_i).$$

Therefore,

$$\hat{g}_{\mathrm{RLOO}}(\theta) = \frac{1}{n(n-1)}\sum_{i\neq j}\left(r(y_i)s_\theta(y_i) - r(y_j)s_\theta(y_i)\right).$$

We then have that

$$\widehat{g}_{\mathrm{RLOO}} = \binom{n}{2}^{-1}\sum_{i<j}\psi_\theta(Y_i, Y_j).$$

□

This U-statistic characterization makes it clear that marginal correctness is not sufficient to maintain unbiasedness.

Proposition 7 (Exact RLOO bias under dependent sampling)**.** *Let $(Y_1, \ldots, Y_n)$ have an arbitrary joint distribution such that each Y_i is marginally distributed as π_θ, and let*

$$g(\theta) := \mathbb{E}_{Y\sim\pi_\theta}\left[r(Y)s_\theta(Y)\right]$$

denote the vanilla policy gradient. Then

$$\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] - g(\theta) = -\frac{1}{n(n-1)}\sum_{i\neq j}\mathbb{E}[r(Y_j)s_\theta(Y_i)]\,.$$

Equivalently, since $\mathbb{E}[s_\theta(Y_i)] = 0$,

$$\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] - g(\theta) = -\frac{1}{n(n-1)}\sum_{i\neq j}\mathrm{Cov}(s_\theta(Y_i), r(Y_j))\,.$$

Hence the bias depends only on the pairwise marginals induced by the dependent sampler.

Proof. Taking expectations in the definition of RLOO gives

$$\begin{aligned}\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[r(Y_i)s_\theta(Y_i)] \\ - \frac{1}{n(n-1)}\sum_{i\neq j}\mathbb{E}[r(Y_j)s_\theta(Y_i)].\end{aligned}$$

Because every Y_i has marginal distribution π_θ, by the policy-gradient theorem we have

$$\mathbb{E}[r(Y_i)s_\theta(Y_i)] = g(\theta),$$

and therefore

$$\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] - g(\theta) = -\frac{1}{n(n-1)}\sum_{i\neq j}\mathbb{E}[r(Y_j)s_\theta(Y_i)] = -\frac{1}{n(n-1)}\sum_{i\neq j}\mathrm{Cov}(s_\theta(Y_i), r(Y_j))$$

Here the last line follows by recognize each term as a covariance, since the score-function identity gives $\mathbb{E}[s_\theta(Y_i)] = 0$. □

Proposition 7 shows that the source of bias under dependent sampling is precisely the covariance between scores and rewards.

Stratified sampling. For stratified sampling, this bias admits a particularly simple form. Let $Z \in \{1, \ldots, n\}$ denote the stratum index, and define

$$\bar{r}_j := \mathbb{E}[r(Y) \mid Z = j], \qquad \bar{s}_j := \mathbb{E}[s_\theta(Y) \mid Z = j].$$

Let Z_i and Z_j denote the stratum indices of two distinct labeled rollouts $i \neq j$. Under stratified sampling with a random permutation of the strata,

$$\Pr(Z_i = a, Z_j = b) = \frac{1}{n(n-1)}, \qquad a \neq b.$$

Conditioning on the pair (Z_i, Z_j) and applying the law of total expectation,

$$\mathbb{E}[r(Y_j)s_\theta(Y_i)] = \sum_{a\neq b}\Pr(Z_i = a, Z_j = b)\,\mathbb{E}[r(Y_j)s_\theta(Y_i) \mid Z_i = a, Z_j = b].$$

Conditioned on $Z_i = a$ and $Z_j = b$, the samples drawn within the two distinct strata are independent. Therefore,

$$\begin{aligned}\mathbb{E}[r(Y_j)s_\theta(Y_i) \mid Z_i = a, Z_j = b] &= \mathbb{E}[r(Y_j) \mid Z_j = b]\,\mathbb{E}[s_\theta(Y_i) \mid Z_i = a] \\ &= \bar{r}_b\,\bar{s}_a,\end{aligned}$$

Marginalizing over the strata and applying conditional independence, we have

$$\mathbb{E}[r(Y_j)s_\theta(Y_i)] = \frac{1}{n(n-1)} \sum_{a \neq b} \bar{s}_a \bar{r}_b.$$

Next, we observe

$$\begin{aligned}\sum_{a \neq b} \bar{s}_a \bar{r}_b &= \sum_{a=1}^{n} \sum_{b=1}^{n} \bar{s}_a \bar{r}_b - \sum_{a=1}^{n} \bar{s}_a \bar{r}_a \\ &= \left(\sum_{a=1}^{n} \bar{s}_a \right) \left(\sum_{b=1}^{n} \bar{r}_b \right) - \sum_{a=1}^{n} \bar{s}_a \bar{r}_a.\end{aligned}$$

Because the strata have equal mass, by construction, we have

$$\frac{1}{n} \sum_{a=1}^{n} \bar{s}_a = \mathbb{E}[s_\theta(Y)] = 0,$$

Therefore,

$$\sum_{a \neq b} \bar{s}_a \bar{r}_b = -\sum_{a=1}^{n} \bar{s}_a \bar{r}_a,$$

and thus

$$\mathbb{E}[r(Y_j)s_\theta(Y_i)] = -\frac{1}{n(n-1)} \sum_{a=1}^{n} \bar{r}_a \bar{s}_a.$$

We recognize the above as

$$\mathrm{Cov}_Z(\bar{r}_Z, \bar{s}_Z) = \frac{1}{n} \sum_{a=1}^{n} \bar{r}_a \bar{s}_a,$$

This gives

$$\mathbb{E}[r(Y_j)s_\theta(Y_i)] = -\frac{1}{n-1} \mathrm{Cov}_Z\big(\bar{r}_Z, \bar{s}_Z\big).$$

Substituting into Proposition 7 gives

$$\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] - g = \frac{1}{n-1} \mathrm{Cov}_Z\big(\bar{r}_Z, \bar{s}_Z\big).$$

Thus, under stratified sampling, the RLOO bias is determined entirely by between-stratum covariance.

Binary rewards and the geometry of the success region. We now specialize to the binary-reward setting used in our experiments, $r(Y) \in \{0, 1\}$. In this case, the conditional mean reward in stratum a is exactly its conditional success probability. Define

$$p_a := \Pr(r(Y) = 1 \mid Z = a).$$

Then

$$\bar{r}_a = \mathbb{E}[r(Y) \mid Z = a] = p_a.$$

Similarly, the marginal success probability is

$$p := \Pr(r(Y) = 1) = \mathbb{E}[r(Y)] = \frac{1}{n} \sum_{a=1}^{n} p_a.$$

Under arithmetic sampling, let

$$A := \{u \in [0, 1] : r(D_\theta(u)) = 1\}$$

denote the success region, and let

$$I_a := \left[\frac{a-1}{n}, \frac{a}{n}\right), \qquad a = 1, \dots, n,$$

denote the strata. Writing

$$\ell_a := \lambda(A \cap I_a)$$

for the successful mass contained in stratum a, we have

$$p_a = \Pr(U \in A \mid U \in I_a) = \frac{\lambda(A \cap I_a)}{\lambda(I_a)} = n\ell_a.$$

Since $\bar{r}_a = p_a$, the between-stratum covariance becomes

$$\begin{aligned}
\mathrm{Cov}_Z(\bar{r}_Z, \bar{s}_Z) &= \mathrm{Cov}_Z(p_Z, \bar{s}_Z) \\
&= \frac{1}{n}\sum_{a=1}^{n}(p_a - p)\bar{s}_a \\
&= \frac{1}{n}\sum_{a=1}^{n} p_a \bar{s}_a \\
&= \sum_{a=1}^{n} \lambda(A \cap I_a)\bar{s}_a,
\end{aligned}$$

where the third equality uses $\sum_a \bar{s}_a = 0$.

Therefore, the RLOO bias under stratified sampling can be written as

$$\boxed{\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] - g = \frac{1}{n-1}\sum_{a=1}^{n}\lambda(A \cap I_a)\bar{s}_a}$$

or equivalently,

$$\boxed{\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] - g = \frac{1}{n(n-1)}\sum_{a=1}^{n}(p_a - p)\bar{s}_a.}$$

Thus, for binary rewards, the geometry of the success region enters the bias through the amount of successful mass contained in each stratum and its alignment with the corresponding mean policy score.

The bias vanishes and the direction is not arbitrary. The bias above has a particularly structured form. By conditioning on the stratum index and applying law of total covariance

$$g := \mathbb{E}[r(Y)s_\theta(Y)] = \underbrace{\mathbb{E}_Z[\mathrm{Cov}(r(Y), s_\theta(Y) \mid Z)]}_{g_{\mathrm{within}}} + \underbrace{\mathbb{E}_Z[\bar{r}_Z\bar{s}_Z]}_{g_{\mathrm{between}}}.$$

Thus the true policy gradient decomposes as

$$g = g_{\mathrm{within}} + g_{\mathrm{between}}.$$

The first term captures reward–score covariation within strata, while the second captures systematic alignment between mean rewards and mean policy scores across strata.

From the stratified RLOO bias characterization above,

$$\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] - g = \frac{1}{n-1}\mathrm{Cov}_Z(\bar{r}_Z, \bar{s}_Z).$$

By the score identity, again

$$\mathbb{E}_Z[\bar{s}_Z] = \mathbb{E}[s_\theta(Y)] = 0,$$

so

$$\mathrm{Cov}_Z(\bar{r}_Z, \bar{s}_Z) = \mathbb{E}_Z[\bar{r}_Z\bar{s}_Z] = g_{\mathrm{between}}.$$

Therefore,

$$\boxed{\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] = g_{\mathrm{within}} + \frac{n}{n-1}g_{\mathrm{between}}.}$$

Importantly, stratified sampling preserves the within-stratum component and rescales the between-stratum component by $n/(n-1)$. The weight goes to 1 as the group size grows.