MIPS: MITIGATING MODE COLLAPSE IN REINFORCEMENT LEARNING FOR LARGE OUTCOME SPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Expected-reward maximization in reinforcement learning concentrates probability mass on a few outcomes, even in tasks that require many distinct, high-quality solutions, a failure known as mode collapse. Inverse Probability Scaling (IPS) counters this in group-based policy gradients such as GRPO by dividing each reward by the outcome's probability, estimated from its frequency within the sampled group. When the outcome space is large relative to the group, outcomes rarely repeat and these estimates become uninformative. Since an outcome can be constructed along many trajectories, its probability is a sum over all of them, and computing it directly is intractable. We propose **Marginal IPS (MIPS)**, which performs marginal evaluation by combining the sampled trajectory with a learned model of the trajectories that lead to each outcome. On Hypergrids, where the exact sampling distribution is computable, MIPS stays near the reward-distribution target as the outcome space grows relative to the rollout group, while frequency-based IPS degrades. In molecular synthesis and phylogenetic inference, MIPS sustains diverse, high-quality sampling where GRPO and frequency-based IPS collapse, showing that MIPS serves as a reliable solution to mode collapse in RL.

1 INTRODUCTION

Many reinforcement learning (RL) problems require a policy to preserve multiple distinct, high-quality solutions rather than identify a single optimum. This setting arises in large-language-model post-training (Ouyang et al., 2022), molecular discovery (Zhu et al., 2023; Koziarski et al., 2024), and combinatorial generation (Bengio et al., 2021), where several terminal outcomes may be simultaneously useful (Kirk et al., 2024; Pan et al., 2022). Standard expected-reward optimization in RL, however, often concentrates probability mass on a small subset of rewarding outcomes, yielding repetitive or homogeneous samples despite high expected reward (Sinha et al., 2026; Kirk et al., 2024; Olivecrona et al., 2017; Li et al., 2026). We refer to this concentration of the terminal sampling distribution as *mode collapse*.

To mitigate mode collapse in RL, we seek a policy that maintains broad coverage of useful outcomes while assigning greater probability to higher-quality ones. This requires the learned outcome probabilities to reflect the reward landscape, so that distinct rewarding regions remain represented according to their relative quality. Under this view, diversity depends not only on how broadly probability mass is spread, but also on where it is placed in the outcome space. This turns mode-collapse mitigation into learning reward-aligned outcome probabilities, i.e., $p_\theta(o_i) \propto R(o_i)$, rather than optimizing generic diversity proxies such as policy entropy, novelty, or pairwise disagreement. Generic entropy regularization can increase stochasticity without specifying how probability should be allocated across outcomes (Haarnoja et al., 2018), although specialized formulations can induce reward-aligned behavior (Tiapkin et al., 2024; Deleu et al., 2024). GFlowNets provide an established framework for learning such outcome probabilities (Bengio et al., 2021; 2023); our focus is on achieving this behavior within a conventional RL optimization pipeline.

Inverse Probability Scaling (IPS) provides such a route by scaling each sampled reward by the inverse probability of its terminal outcome (Sinha et al., 2026). Intuitively, IPS increases the relative advantage of rare rewarding outcomes and decreases that of outcomes sampled more frequently.

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.

The difficulty with IPS is estimating the marginal probability of a terminal outcome. IPS-GRPO uses outcome frequencies within a rollout group, which become uninformative for outcomes whose probability is too small to yield repeated observations within the group (Section 2).

A readily available trajectory likelihood is not a substitute for the needed terminal probability. In constructive environments, many distinct action sequences can lead to the same outcome, so one path's probability does not include the probability of the alternative paths producing that outcome.

We introduce *Marginal Inverse Probability Scaling* (MIPS), which evaluates the marginal terminal probability using one sampled trajectory together with a learned trajectory-conditional model. The conditional model accounts for the many-to-one mapping from trajectories to outcomes, allowing MIPS to estimate the outcome marginal without sampling a large number of trajectories or explicit path enumeration. The resulting scaled rewards enter the standard GRPO surrogate unchanged (Shao et al., 2024; Schulman et al., 2017). Figure 1 illustrates the effect on a Hypergrid with four rewarding modes: entropy-regularized RL and frequency-based IPS each collapse onto a single mode, whereas MIPS places probability on all four in line with the reward.

We evaluate MIPS across three increasingly large and structured domains. On enumerable Hypergrids, exact policy probabilities allow direct measurement of agreement between the reward landscape and policy; sweeps over terminal-space size and rollout-group size additionally test how the effectiveness of IPS depends on group size G as the outcome space grows. In reaction-based molecular synthesis (Koziarski et al., 2024), we measure retention of diverse high-reward modes and molecular scaffolds. In phylogenetic tree construction (Zhou et al., 2024), we measure terminal coverage and alignment between sampling mass and reward. The latter two evaluations assess diverse coverage and the retention of high-quality solutions.

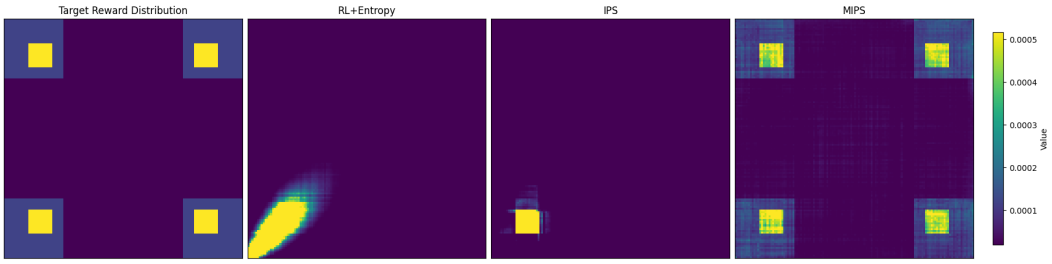


Figure 1: Terminal distributions on a Hypergrid. Left to right: the reward-distribution target, entropy-regularized RL, frequency-based IPS-GRPO, and MIPS-GRPO.

Our contributions are:

- We introduce MIPS, which estimates an outcome's marginal probability using a learned trajectory-conditional model, while retaining the standard GRPO update.
- We evaluate MIPS on Hypergrids, where the learned outcome distribution can be compared exactly with the reward distribution. We then extend the evaluation to molecular synthesis and phylogenetic tree construction, where we measure coverage of high-quality outcomes and alignment with the reward.

2 Why IPS has limitations

Inverse Probability Scaling (IPS) modifies the reward for an outcome o as

$$\widetilde{R}(o) = \frac{R(o)}{P_\theta(o)},$$

where $P_\theta(o)$ is the probability that the current policy generates o. With the true $P_\theta(o)$, IPS admits a reward-proportional stationary solution, $P_\theta(o_i) \propto R(o_i)$ (Sinha et al., 2026).

In practice, however, $P_\theta(o)$ is generally unknown. IPS-GRPO estimates it from outcome frequencies within a group of G rollouts (Sinha et al., 2026), so the estimate has resolution $1/G$. For an outcome

o, the expected number of occurrences in a group is $GP_\theta(o)$, and after one occurrence, the probability of observing it again is $1-(1-P_\theta(o))^{G-1}$. When $GP_\theta(o) \ll 1$, repeated observations are unlikely, and a sampled outcome is therefore typically observed only once and assigned probability $1/G$. In this regime, the within-group frequency provides little information about its underlying probability. This makes the choice of G important in practice.

In large constructive problems, the size of the outcome space is typically unknown in advance, and the effective outcome space accessible to the policy evolves throughout training. Because neither the scale nor the effective support of this space is fixed, no single choice of G can reliably provide enough repeated outcomes throughout training. Choosing G to be conservatively large is also impractical because the sampling cost grows directly with the group size, motivating an estimator that does not depend on repeated occurrences of the same outcome.

MIPS addresses this limitation by estimating the marginal probability of each sampled outcome without relying on its frequency within the rollout group.

3 Marginal Inverse Probability Scaling (MIPS)

IPS requires the probability of the outcome o, rather than the probability of any particular trajectory that generates it. For an outcome o, this *marginal outcome probability* is

$$P_\theta(o) = \sum_{\tau \in \mathcal{T}(o)} P_\theta(\tau),$$

where $\mathcal{T}(o)$ denotes the set of construction trajectories generating output o. This marginalization motivates the name *Marginal Inverse Probability Scaling* (MIPS). A single trajectory likelihood cannot generally replace this quantity because multiple distinct trajectories may generate the same outcome. In large constructive spaces, explicitly summing over all trajectories in $\mathcal{T}(o)$ is generally intractable.

For a finite, acyclic, deterministic constructive process with positive rewards, $R(o) > 0$, MIPS avoids this sum using a learned trajectory-conditional model.

Let $\tau = (s_0, a_0, \ldots, s_T)$ be an on-policy trajectory with likelihood $P_\theta(\tau) = \prod_{t=0}^{T-1} \pi_\theta(a_t \mid s_t)$. Since o is determined by τ, $P_\theta(\tau, o) = P_\theta(\tau)$. By the chain rule, $P_\theta(\tau, o) = P_\theta(o)P_\theta(\tau \mid o)$, and therefore $P_\theta(\tau) = P_\theta(o)P_\theta(\tau \mid o)$. For any supported trajectory,

$$P_\theta(o) = \frac{P_\theta(\tau)}{P_\theta(\tau \mid o)}.$$

Thus, given the trajectory-conditional probability, the marginal outcome probability can be recovered using one trajectory for the sampled outcome.

MIPS therefore introduces a learnable **trajectory-conditional model** $c_\phi(\tau \mid o)$, such that:

$$c_\phi(\tau \mid o) \approx P_\theta(\tau \mid o), \qquad \textit{and} \qquad \sum_{\tau \in \mathcal{T}(o)} c_\phi(\tau \mid o) = 1,$$

and substitutes it into the marginal identity. For each sampled trajectory, MIPS defines

$$P_{\theta,\phi}(o) = \frac{P_\theta(\tau)}{c_\phi(\tau \mid o)}, \qquad \widetilde{R} = \frac{R(o)}{P_{\theta,\phi}(o)} = \frac{R(o)c_\phi(\tau \mid o)}{P_\theta(\tau)}. \tag{1}$$

If $c_\phi(\tau \mid o) = P_\theta(\tau \mid o)$, then $P_{\theta,\phi}(o) = P_\theta(o)$ for every trajectory generating o.

To make the conditional model tractable, it can be factorized over predecessor transitions. For a finite acyclic construction process with a unique initial state and a Markov policy,

$$c_\phi(\tau \mid o) = \prod_{t=1}^{T} c_\phi(s_{t-1}, a_{t-1} \mid s_t),$$

with each factor normalized over valid incoming transitions.

The conditional model is trained by maximum likelihood using on-policy trajectories:

$$\mathcal{L}_{\text{cond}}(\phi) = -\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T}\log c_\phi(s_{t-1}, a_{t-1} \mid s_t)\right]. \tag{2}$$

At each training step, MIPS uses the current conditional model $c_\phi(\tau \mid o)$ to compute scaled rewards for the new rollouts. It then keeps those scaled rewards fixed, normalizes them within the rollout group, and updates the policy with the usual clipped GRPO objective. The complete update is given in Appendix A.

With an approximate conditional model, $P_{\theta,\phi}(o)$ may vary across trajectories to the same outcome and is not generally an unbiased estimate of $P_\theta(o)$; nor do the samplewise estimates necessarily form a normalized outcome distribution. Its accuracy depends on conditional-model fit and adequate policy support.

3.1 Relation to GFlowNets

GFlowNets also target reward-proportional terminal sampling (Bengio et al., 2021; 2023). The trajectory-conditional factorization used by MIPS is similar to the backward policy used in GFlowNets (Malkin et al., 2022). Related work connects GFlowNet objectives to entropy-regularized RL (Tiapkin et al., 2024), analyzes multi-path marginal corrections (Deleu et al., 2024), and develops policy-gradient and PPO-style GFlowNet training (Niu et al., 2024; Zykova-Myzina et al., 2026).

MIPS differs in how this conditional distribution is used: it estimates the *current policy's marginal terminal probability* for IPS, thereby avoiding repeated terminal observations while retaining the standard GRPO surrogate. It does not introduce a new GFlowNet objective. Detailed comparisons appear in Appendix B.

4 Experiments

We first study MIPS in a controlled setting, the Hypergrid environment. Because the terminal distribution of each trained policy can be computed exactly in this setup, we can compare it directly with the reward-distribution target and vary the outcome-space size and group size independently. We then test whether these findings carry over to two large structured domains, molecular synthesis and phylogenetic tree construction, where the outcome spaces are too large to enumerate. In these domains, we assess coverage of high-quality outcomes and the alignment between sampling density and reward, and include task-specific GFlowNet samplers as reference points.

4.1 Hypergrid scalability with outcome-space size and group size

We use a two-dimensional Hypergrid with side length H, in which each state is represented by a pair of coordinates in $\{0, \ldots, H-1\}^2$. An agent starts at $(0, 0)$ and, at each step, the agent can either increment one coordinate or stop. Any state at which the agent stops is a terminal state, so the agent can choose to terminate at any point in the grid. Different orders of coordinate increments can reach the same terminal state, creating multiple construction trajectories for each outcome. A grid with side length H has H^2 possible terminal states. Here the outcomes are the same as the terminal states. Since the Hypergrid is enumerable, we can compute the learned outcome distribution P_θ exactly and directly assess whether it matches the desired reward-aligned distribution. Each episode terminates at a grid coordinate $s \in \{0, \ldots, H-1\}^2$, which is mapped uniformly to $x \in [-1, 1]^2$. The reward is defined as

$$R(x) = R_0 + R_1 \prod_{i=1}^{2} 1(|x_i| > 0.5) + R_2 \prod_{i=1}^{2} 1(0.6 < |x_i| < 0.8)\,, \tag{3}$$

where $0 < R_0 \ll R_1 < R_2$. This construction produces 4 symmetric high-reward modes surrounded by a ring, resulting in a sharply multimodal reward landscape (Figure 1, left). We set $R_0 = 0.1$, $R_1 = 0.5$, and $R_2 = 2.0$, ensuring that every terminal state receives a positive reward.

Experiment. We compare three training methods: GRPO, IPS-GRPO, and MIPS-GRPO. For each method, we consider grid side lengths $H \in \{4, 8, 16, 32, 64, 128, 256, 512\}$ and rollout-group sizes

$G \in \{16, 32, 64, 128, 256, 512, 1024, 2048\}$, yielding 192 experimental configurations in total. For each configuration, we trained the policy until convergence. To quantify how closely the trained policy's outcome distribution matches the reward distribution, we use Jensen–Shannon divergence (JSD) and total variation distance (TV), with lower values indicating better alignment. Because the environment is enumerable, we use dynamic programming to compute the exact outcome distribution of each trained policy.

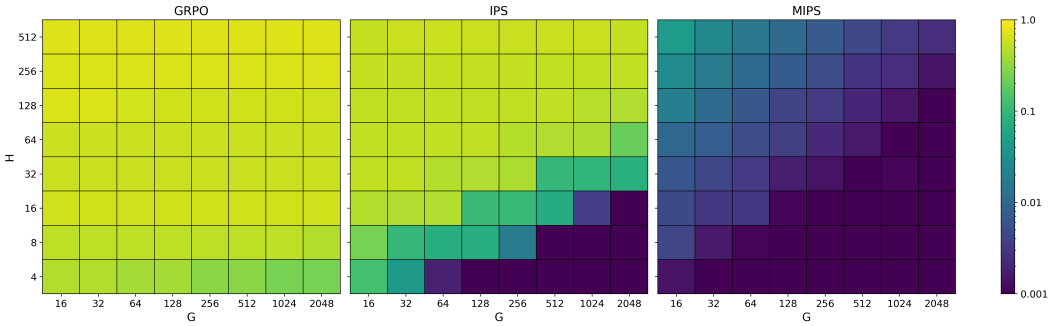


Figure 2: **Hypergrid scalability.** JSD between the exact outcome distribution and the reward-distribution target for GRPO (left), IPS-GRPO (center), and MIPS-GRPO (right). Grid side length H is shown on the vertical axis and group size G on the horizontal axis, with H^2 outcomes and a shared logarithmic color scale across panels.

Results. Figure 2 shows how the three methods behave as the outcome space becomes larger and the rollout group size changes. GRPO remains far from the reward-distribution target across most settings.

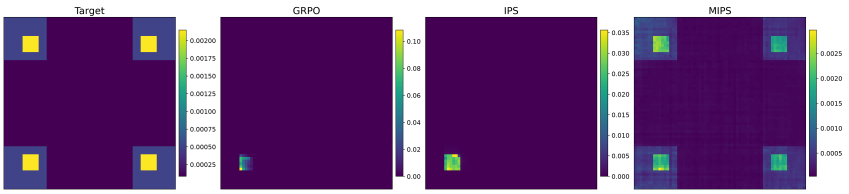
IPS behaves differently. On smaller outcome spaces, it can closely match the target reward distribution, especially when the rollout group is sufficiently large. However, its performance deteriorates as the outcome space grows. Increasing the group size helps, but mainly shifts the point at which this deterioration occurs rather than removing it. MIPS shows a different scaling behavior. Its outcome distribution remains close to the reward-distribution target over a much wider range of outcome spaces and group sizes. In particular, its behavior changes much less when the group is small relative to the number of possible outcomes.

Figure 3 makes this behavior more concrete by showing the learned outcome distributions for a fixed Hypergrid while the group size is increased. GRPO concentrates almost all of its probability around a single rewarding region, exhibiting mode collapse. With a small group, IPS-GRPO shows a similar form of collapse but covers more outcomes. As the group size G becomes larger, IPS becomes more effective. MIPS behaves differently. Even with the smaller group size G, it places probability around all rewarding regions. Increasing the group size G mainly refines this distribution rather than changing which modes are represented.

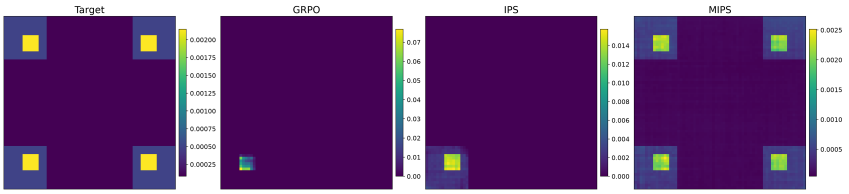
The important observation is therefore not simply that MIPS produces a more diverse distribution, but that its diversity follows the structure of the reward distribution itself. The complete JSD and TV results (Appendix C.1, Tables 4 and 5) show that, although sufficiently large groups allow IPS-GRPO to match MIPS on the smallest grids, MIPS achieves consistently better alignment as the outcome space grows, across all tested group sizes G.

Taken together, the Hypergrid experiments verify two key properties of MIPS. First, MIPS preserves diversity as the outcome space grows without requiring the rollout group size G to grow with it. Second, this diversity is not produced by indiscriminate randomness: the learned policy distribution remains structured by the reward itself, placing probability mass across the distinct rewarding modes rather than spreading it uniformly. These results show that MIPS can scale to large outcome spaces while maintaining reward-aligned diversity and substantially reducing sensitivity to the hyperparameter G.

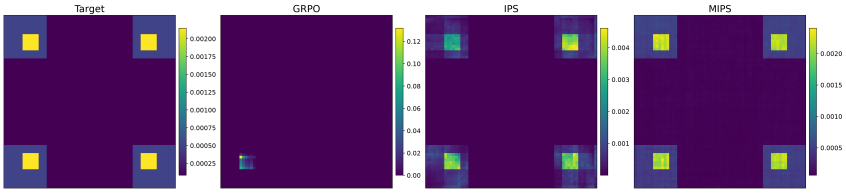
Stability and convergence of marginal-probability estimation. Because Hypergrid marginals are exactly computable by dynamic programming, we directly evaluate the accuracy of the



(a) $G = 64$



(b) $G = 512$

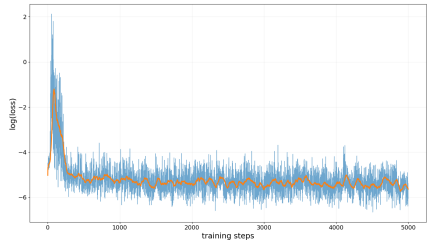


(c) $G = 2048$

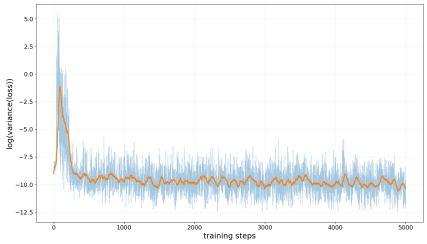
Figure 3: **Outcome distributions on the** 64×64 **Hypergrid.** The reward-distribution target and the exact outcome distributions learned by GRPO, IPS-GRPO, and MIPS-GRPO are shown for increasing rollout-group sizes G.

MIPS estimator during training. For each trajectory sampled during training, we measure $\ell_i = (P_{\theta,\phi}(o_i) - P_\theta(o_i))^2$, where $P_{\theta,\phi}(o_i)$ is the estimated marginal and $P_\theta(o_i)$ is the exact marginal under the current policy. Figure 4 reports the batch mean and variance of ℓ_i across training on the log scale.

Both decrease rapidly after an initial transient and remain low thereafter. This empirically supports stable convergence of the coupled policy and trajectory-conditional model to a low-error regime.



(a) Log mean squared probability error.



(b) Log variance of the squared probability error.

Figure 4: **Marginal-probability estimation.** Raw measurements and smoothed trends are shown.

4.1.1 Ablation of conditional scaling

We ablate the use of the trajectory-conditional model $c_\phi(\tau \mid o)$ in the MIPS scaling factor. In the full method, the scaling term uses the conditional trajectory model $c_\phi(\tau \mid o)$ to account for how trajectories are distributed given the outcome. In the ablated variant, this conditional information is removed from the scaling computation, so the estimator relies only on the trajectory likelihood $P_\theta(\tau)$. This is equivalent to setting $c_\phi(\tau \mid o) = 1$. Any other constant yields the same behavior because the scaled rewards are group-normalized when computing advantages.

Table 1 shows that removing conditional scaling substantially degrades alignment with the reward-distribution target across all Hypergrid configurations. Both Jensen–Shannon divergence (JSD) and total variation (TV) distance increase sharply in every setting. These results show that the benefit of MIPS does not arise from trajectory probability scaling alone. Conditioning the scaling factor on the outcome is important for estimating the marginal outcome probability and, consequently, for matching the target reward distribution.

Table 1: **Ablation of conditional scaling on Hypergrids.** JSD and TV distance from the reward-distribution target. We compare full MIPS with a variant that does not use the trajectory-conditional model in the scaling factor.

		JSD ↓		TV ↓	
H	G	MIPS	No $c_\phi(\tau \mid o)$	MIPS	No $c_\phi(\tau \mid o)$
32	32	**0.0032**	0.5740	**0.0617**	0.9147
32	64	**0.0040**	0.5592	**0.0697**	0.9013
64	32	**0.0077**	0.6315	**0.0996**	0.9622
64	64	**0.0052**	0.6428	**0.0815**	0.9689

4.2 Molecular synthesis

Molecular synthesis can be viewed as a sequential decision-making problem where an agent builds a molecule through a series of chemically valid reactions. At each step, the agent selects a reaction and compatible building blocks, and the final molecule receives a reward based on its desired properties. This gives a natural reinforcement-learning formulation, where the sequence of synthesis decisions forms a trajectory and the resulting molecule is the outcome. The number of possible trajectories grows combinatorially with the available reactions, building blocks, and synthesis steps, creating an enormous search space. Moreover, different synthesis trajectories can lead to the same molecule, so many trajectories may correspond to a single outcome. The goal is therefore not only to find the best molecule, but to explore this large combinatorial space and discover diverse, high-quality molecules.

Environment. We use the reaction-based environment, sEH proxy task, and reward construction of RGFN (Koziarski et al., 2024) without modification. Starting from an initial building block, the policy repeatedly selects a reaction template, a compatible building block, and one of the resulting products until choosing to stop. The chemistry contains 17 reaction types and 350 low-cost building blocks.

Thus, as the number of reaction steps increases, many different trajectories can contribute probability to the same molecule, making this a challenging setting for marginal outcome-probability estimation. We compare GRPO, IPS-GRPO, MIPS-GRPO, and RGFN trained with its published trajectory-balance configuration.

Evaluation. After $4{,}000$ training iterations, we draw $50{,}000$ molecules from each final policy. Molecules are identified by canonical SMILES, and uniqueness is measured only over valid samples. To measure diversity specifically within the high-quality region, we follow the mode criterion of Koziarski et al. (2024): unique molecules with sEH proxy score above 7 are processed in decreasing score order, and the Leader algorithm selects a new mode when its Tanimoto similarity to every previously selected mode is at most 0.5. We additionally count distinct Murcko scaffolds among molecules with proxy score above 7 to measure whether high-quality samples span different molecular cores. Finally, we report the mean sEH proxy over all samples and over the discovered modes to assess whether broader coverage is accompanied by maintaining the molecular quality.

Coverage and quality. Table 2 compares the methods in terms of overall molecular diversity, high-reward mode coverage, structural diversity, and proxy quality.

GRPO collapses almost completely, producing essentially a single unique molecule per run. IPS-GRPO improves overall uniqueness modestly, generating 17 unique molecules on average, but still fails to recover any high-reward modes. RGFN maintains much broader support, generating $34{,}744$ unique molecules (69.5% of valid samples) and recovering 18 distinct high-reward modes.

Table 2: **Final-policy molecular synthesis results.** Statistics from $50{,}000$ samples across 3 seeds. The selected metrics measure overall coverage, high-quality diversity, structural diversity, and molecular quality.

Metric	What it measures	GRPO	IPS-GRPO	RGFN	MIPS-GRPO
Unique molecules (count) ↑	Overall molecular diversity	1 ± 0	17 ± 11	$34{,}744 \pm 1{,}270$	$\mathbf{45{,}794 \pm 2{,}332}$
High-reward modes ↑	Distinct high-quality regions	0 ± 0	0 ± 0	18 ± 1	$\mathbf{70 \pm 2}$
Murcko scaffolds (proxy > 7) ↑	High-quality structural diversity	0 ± 0	0 ± 0	25 ± 6	$\mathbf{149 \pm 42}$
Mean proxy (all samples)	Overall molecular quality	6.54 ± 0.37	4.68 ± 1.31	4.19 ± 0.37	4.47 ± 0.13
Mean proxy (modes) ↑	Quality of discovered modes	—	—	7.17 ± 0.05	$\mathbf{7.22 \pm 0.08}$

MIPS-GRPO further expands this coverage, generating $45{,}794$ unique molecules (91.8% of valid samples), along with 70 high-reward modes and 149 high-reward Murcko scaffolds. The increase in scaffolds indicates that these additional modes span different molecular cores rather than consisting only of small variations of similar structures.

This broader coverage does not come at the cost of proxy quality. MIPS-GRPO achieves a mean proxy score of 4.47 across all generated molecules, compared with 4.19 for RGFN, while its discovered modes have a mean proxy score of 7.22, compared with 7.17 for RGFN. Together, these results show that MIPS-GRPO expands both overall and high-reward coverage while maintaining comparable molecular quality.

4.3 Phylogenetic tree construction

Phylogenetic tree construction can be formulated as a sequential decision-making problem in which an agent builds a tree by repeatedly merging pairs of subtrees and assigning branch lengths. The resulting sequence of construction decisions forms a trajectory, while the completed phylogenetic tree is the outcome and receives a reward based on how well it explains the observed sequence data. This is a challenging combinatorial search problem because the number of possible tree topologies grows rapidly with the number of taxa, and many different construction trajectories can lead to the same final tree. The task therefore provides a structured setting in which to test whether a policy can retain diverse tree hypotheses while continuing to favor high-quality ones.

Setup and evaluation. We use a 10-taxon setting derived from the DS1 benchmark and the bottom-up construction process of PhyloGFN (Zhou et al., 2024): starting from individual taxa, the policy merges pairs of subtrees and assigns branch lengths until a complete tree is formed. Rewards are based on the Jukes–Cantor log-likelihood (Jukes & Cantor, 1969). We compare GRPO, IPS-GRPO, MIPS-GRPO, and PhyloGFN using 10^6 samples from each final policy.

We measure coverage by the number of distinct tree signatures, where a signature combines tree topology and branch-length assignment, and assess tree quality using the mean log-likelihood (MLL) of sampled trees. Alignment with the reward distribution is measured by the Pearson correlation between outcome log-probability of the policy and log-likelihood of the generated tree, both on each method's own samples (sampling r), and on a common set of trees when the corresponding densities are available (shared-tree r). For MIPS-GRPO, the tree log-probability is estimated using Equation (1), with the trained policy and conditional model providing $P_\theta(\tau)$ and $c_\phi(\tau \mid o)$.

Table 3: **Phylogenetic tree construction with** 10 **taxa.** Results from 10^6 samples per final policy (mean $\pm$ standard deviation over three seeds).

Method	Unique signatures / 1M ↑	MLL	Sampling r ↑	Shared-tree r ↑
GRPO	1 ± 0	-4318.4 ± 0.0	—	—
IPS-GRPO	$263{,}532 \pm 24{,}466$	-4294.4 ± 3.7	0.140 ± 0.010	—
PhyloGFN	$1{,}000{,}000 \pm 0$	-4366.5 ± 2.2	0.836 ± 0.044	0.936 ± 0.012
MIPS-GRPO	$1{,}000{,}000 \pm 0$	-4372.9 ± 0.9	$\mathbf{0.967 \pm 0.003}$	$\mathbf{0.962 \pm 0.010}$

Coverage and quality. Table 3 shows that MIPS-GRPO and PhyloGFN achieve broad coverage, whereas IPS-GRPO covers far fewer trees and GRPO collapses to a single tree. Broader coverage

comes with lower MLL: instead of concentrating on a few high-likelihood trees, the policy spreads probability across a wider range of trees, including lower-likelihood ones.

MIPS-GRPO shows the strongest reward alignment, with a sampling correlation of $r = 0.967$ compared with 0.836 for PhyloGFN and 0.140 for IPS-GRPO. It also has the highest correlation on the shared tree set (Figure 5), so this advantage does not depend on each method's own samples. Missing correlations could not be estimated because the corresponding policies generated too few distinct trees. Overall, MIPS-GRPO combines broad topology coverage with a sampling density that increases consistently with tree likelihood.

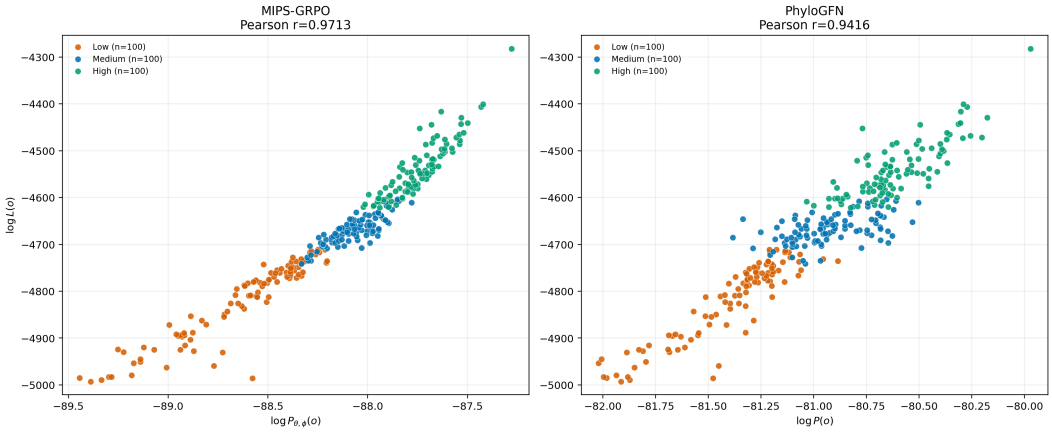


Figure 5: **Probability–likelihood alignment on a shared tree set.** Terminal log-likelihood versus outcome log-probability for MIPS-GRPO (left) and PhyloGFN (right) on a common set of 300 trees (50 per method per log-likelihood band). A common set removes the dependence of the correlation on each method's own sampling distribution. MIPS-GRPO spreads its mass over a larger outcome space, so its log-probabilities are lower and absolute values are not comparable across panels. Pearson r is reported above each panel.

Taken together, these results show that MIPS does more than increase stochasticity in phylogenetic construction. Compared with GRPO and IPS-GRPO, it covers a far broader set of trees while assigning higher probability to higher-likelihood ones, and it achieves similarly broad coverage to PhyloGFN while exhibiting stronger reward alignment.

5 Discussion

Our results show that MIPS can maintain diverse, high-quality outcomes as the outcome space grows, effectively mitigating mode collapse in large outcome spaces. On Hypergrids, MIPS remains closely aligned with the reward distribution across different outcome-space sizes and group sizes. The marginal-estimation results and ablation studies further show that the trajectory-conditional model is important for accurately estimating marginal outcome probabilities. This behavior also extends to larger, structured problems. In molecular synthesis, MIPS discovers diverse, high-reward molecules and molecular scaffolds while maintaining high mode quality. In phylogenetic inference, MIPS achieves broad tree coverage while assigning higher probability to higher-likelihood trees, resulting in stronger alignment between the policy and the reward distribution over outcomes.

MIPS is limited by the accuracy of its learned trajectory-conditional model and by the assumptions underlying the current formulation, which considers finite, acyclic, deterministic constructive processes with positive rewards. Future work could extend MIPS beyond these assumptions, particularly to continuous outcome spaces in which outcomes may not be uniquely defined but instead exhibit varying degrees of similarity. Such settings could motivate marginal-probability corrections that account for similarity or local structure among outcomes rather than relying on exact outcome identity.

More broadly, this study opens the door to using RL algorithms effectively for search and sampling tasks, rather than only for reward maximization, through an appropriate choice of advantage calculation.

AI USE STATEMENT

We used generative AI tools to assist with editing and polishing the writing of this paper, including improving clarity, grammar, and presentation. We also used generative AI tools for retrieval and discovery, including identifying potentially relevant related work. All AI-assisted text, citations, and literature suggestions were reviewed and verified by the authors. The authors take responsibility for the final content of this work.

REFERENCES

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in neural information processing systems*, 34:27381–27394, 2021.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023. URL http://jmlr.org/papers/v24/22-0364.html.

Tristan Deleu, Padideh Nouri, Nikolay Malkin, Doina Precup, and Yoshua Bengio. Discrete probabilistic inference as control in multi-path environments. In Negar Kiyavash and Joris M. Mooij (eds.), *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pp. 997–1021. PMLR, 15–19 Jul 2024. URL https://proceedings.mlr.press/v244/deleu24a.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Thomas H. Jukes and Charles R. Cantor. Evolution of protein molecules. In H. N. Munro (ed.), *Mammalian Protein Metabolism*, volume 3, pp. 21–132. Academic Press, New York, 1969. doi: 10.1016/B978-1-4832-3211-9.50009-7.

Hohyun Kim, Seunggeun Lee, and Min-Hwan Oh. Symmetry-aware GFlowNets. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30334–30362. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kim25s.html.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 20620–20653, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5a68d05006d5b05dd9463dd9c0219db0-Paper-Conference.pdf.

Michał Koziarski, Andrei Rekesh, Dmytro Shevchuk, Almer van der Sloot, Piotr Gaiński, Yoshua Bengio, Cheng-Hao Liu, Mike Tyers, and Robert A Batey. Rgfn: Synthesizable molecular generation using gflownets. *Advances in Neural Information Processing Systems*, 37:46908–46955, 2024.

Chenyi Li, Yuan Zhang, Bo Wang, Guoqing Ma, Wei Tang, Haoyang Huang, and Nan Duan. Setpo: Set-level policy optimization for diversity-preserving llm reasoning. *arXiv preprint arXiv:2602.01062*, 2026.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35: 5955–5967, 2022.

Puhua Niu, Shili Wu, Mingzhou Fan, and Xiaoning Qian. Gflownet training by policy gradients. *arXiv preprint arXiv:2408.05885*, 2024.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):48, 2017.

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. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Ling Pan, Dinghuai Zhang, Aaron Courville, Longbo Huang, and Yoshua Bengio. Generative augmented flow networks. *arXiv preprint arXiv:2210.03308*, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Abhijeet Sinha, Sundari Elango, and Dianbo Liu. Expected return causes outcome-level mode collapse in reinforcement learning and how to fix it with inverse probability scaling. *arXiv preprint arXiv:2601.21669*, 2026.

Daniil Tiapkin, Nikita Morozov, Alexey Naumov, and Dmitry P Vetrov. Generative flow networks as entropy-regularized rl. In *International Conference on Artificial Intelligence and Statistics*, pp. 4213–4221. PMLR, 2024.

Ming Yang Zhou, Zichao Yan, Elliot Layne, Nikolay Malkin, Dinghuai Zhang, Moksh Jain, Mathieu Blanchette, and Yoshua Bengio. Phylogfn: Phylogenetic inference with generative flow networks. In *International Conference on Learning Representations*, volume 2024, pp. 35370–35403, 2024.

Yiheng Zhu, Jialu Wu, Chaowen Hu, Jiahuan Yan, Tingjun Hou, Jian Wu, et al. Sample-efficient multi-objective molecular optimization with gflownets. *Advances in neural information processing systems*, 36:79667–79684, 2023.

Anna Zykova-Myzina, Timofei Gritsaev, Daniil Tiapkin, and Nikita Morozov. Proximal policy optimization for amortized discrete sampling. *arXiv preprint arXiv:2606.15793*, 2026.

A MIPS-GRPO POLICY UPDATE

Algorithm 1 gives the update used for MIPS-GRPO.

Algorithm 1 MIPS-GRPO

Require: policy π_θ; conditional model c_ϕ; group size G; clipping parameter ϵ; policy epochs K; conditional-model epochs K_c; entropy coefficient β for entropy $\mathcal{H}$

for each update **do**
  $\pi_{\text{old}} \leftarrow \pi_\theta$
  Sample G trajectories $\tau_i = (s_{i,0}, a_{i,0}, \ldots, s_{i,T_i}) \sim \pi_{\text{old}}$
  Let o_i denote the terminal outcome of τ_i
  without gradients:
  for $i = 1, \ldots, G$ **do**
    $\ell_i^\pi \leftarrow \sum_{t=0}^{T_i-1} \log \pi_{\text{old}}(a_{i,t} \mid s_{i,t})$
    $\ell_i^c \leftarrow \sum_{t=1}^{T_i} \log c_\phi(s_{i,t-1}, a_{i,t-1} \mid s_{i,t})$
    $\log w_i \leftarrow \log R(o_i) + \ell_i^c - \ell_i^\pi$
  end for
  $\widehat{A}_i \leftarrow \dfrac{w_i - \text{mean}_j(w_j)}{\text{std}_j(w_j)}$
  Stop gradients through $\widehat{A}_i$
  for $k = 1, \ldots, K$ **do**
    $\rho_{i,t}(\theta) \leftarrow \dfrac{\pi_\theta(a_{i,t} \mid s_{i,t})}{\pi_{\text{old}}(a_{i,t} \mid s_{i,t})}$
    $\mathcal{H}_\pi(\theta) \leftarrow \dfrac{1}{G}\sum_{i=1}^{G}\dfrac{1}{T_i}\sum_{t=0}^{T_i-1}\mathcal{H}(\pi_\theta(\cdot \mid s_{i,t}))$
    $\mathcal{L}_\pi(\theta) \leftarrow -\dfrac{1}{G}\sum_{i=1}^{G}\dfrac{1}{T_i}\sum_{t=0}^{T_i-1}\min\Big(\rho_{i,t}(\theta)\widehat{A}_i,\ \text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_i\Big) - \beta\,\mathcal{H}_\pi(\theta)$
    Take one optimizer step on θ
  end for
  for $k_c = 1, \ldots, K_c$ **do**
    $\mathcal{L}_c(\phi) \leftarrow -\dfrac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{T_i}\log c_\phi(s_{i,t-1}, a_{i,t-1} \mid s_{i,t})$
    Take one optimizer step on ϕ
  end for
end for

B FURTHER CONNECTIONS TO GFLOWNETS AND SYMMETRY

Trajectory balance. The trajectory-conditional model c_ϕ represents a distribution over construction paths, as a GFlowNet backward policy does through transitions from a terminal object to the initial state. In trajectory balance (TB), the forward and backward probabilities satisfy (Malkin et al., 2022)

$$ZP_F(\tau) = R(o)P_B(\tau \mid o).$$

At exact balance, $P_F(\tau)/P_B(\tau \mid o) = R(o)/Z = P_F(o)$. MIPS uses the same probability-ratio identity, but to estimate the *current* policy's marginal terminal probability for IPS, rather than to enforce this target-reward constraint. Its trajectory-conditional model tracks the policy's conditional paths through likelihood fitting; the policy receives the resulting IPS correction through the GRPO surrogate.

Policy-gradient GFlowNets. GFlowNets can also be trained through policy gradients and entropy-regularized control (Niu et al., 2024; Tiapkin et al., 2024; Deleu et al., 2024). Zykova-Myzina et al. (2026) derive Entropic PPO for GFlowNet sampling, using log-backward-probability rewards, entropy regularization, and optionally likelihood-based backward-policy learning. MIPS instead augments conventional reward-based RL: a trajectory-conditional model estimates marginal outcome probabilities for IPS, whose scaled rewards define group-relative advantages.

Multiplicity and symmetry. Kim et al. (2025) address biases from graph symmetries and equivalent actions through symmetry-aware reward scaling. MIPS instead corrects construction-path multiplicity when estimating outcome probabilities from trajectory likelihoods. For outcomes defined up to symmetry, the policy and conditional model must use the same equivalence classes; symmetry and MIPS corrections thus address related but distinct biases.

C Experimental details

C.1 Hypergrid scalability

Table 4: **Jensen–Shannon divergence on the complete Hypergrid sweep (↓).** Error between the exact learned terminal distribution and the reward-distribution target. Each entry is the mean ± standard deviation across repeated runs.

H	Method	$G = 16$	$G = 32$	$G = 64$	$G = 128$	$G = 256$	$G = 512$	$G = 1024$	$G = 2048$
4	GRPO	0.4535 ± 0.0005	0.4539 ± 0.0000	0.3981 ± 0.1115	0.3983 ± 0.1113	0.2985 ± 0.1059	0.2985 ± 0.1060	0.2457 ± 0.0223	0.2429 ± 0.0235
	IPS-GRPO	0.1277 ± 0.0006	0.0420 ± 0.0018	0.0019 ± 0.0005	$\mathbf{0.0003 \pm 0.0003}$	$\mathbf{0.0001 \pm 0.0000}$	$\mathbf{0.0001 \pm 0.0000}$	$\mathbf{0.0001 \pm 0.0000}$	$\mathbf{0.0000 \pm 0.0000}$
	MIPS-GRPO	$\mathbf{0.0014 \pm 0.0005}$	$\mathbf{0.0006 \pm 0.0004}$	$\mathbf{0.0004 \pm 0.0002}$	0.0006 ± 0.0004	0.0002 ± 0.0001	$\mathbf{0.0001 \pm 0.0000}$	$\mathbf{0.0001 \pm 0.0000}$	0.0001 ± 0.0000
8	GRPO	0.5062 ± 0.0002	0.5047 ± 0.0033	0.5066 ± 0.0002	0.5067 ± 0.0002	0.5068 ± 0.0001	0.5068 ± 0.0001	0.5066 ± 0.0004	0.4525 ± 0.0625
	IPS-GRPO	0.2430 ± 0.1222	0.0999 ± 0.0046	0.0781 ± 0.0010	0.0750 ± 0.0003	0.0173 ± 0.0027	$\mathbf{0.0005 \pm 0.0002}$	$\mathbf{0.0001 \pm 0.0001}$	$\mathbf{0.0001 \pm 0.0001}$
	MIPS-GRPO	$\mathbf{0.0042 \pm 0.0021}$	$\mathbf{0.0016 \pm 0.0006}$	$\mathbf{0.0011 \pm 0.0004}$	$\mathbf{0.0008 \pm 0.0004}$	$\mathbf{0.0009 \pm 0.0003}$	0.0006 ± 0.0005	0.0003 ± 0.0001	0.0002 ± 0.0001
16	GRPO	0.6069 ± 0.0011	0.6081 ± 0.0003	0.6086 ± 0.0002	0.6088 ± 0.0001	0.6089 ± 0.0000	0.6089 ± 0.0000	0.6089 ± 0.0000	0.6090 ± 0.0000
	IPS-GRPO	0.4514 ± 0.0007	0.4386 ± 0.0016	0.4401 ± 0.0002	0.1091 ± 0.0010	0.1058 ± 0.0010	0.0689 ± 0.0011	0.0035 ± 0.0002	$\mathbf{0.0003 \pm 0.0001}$
	MIPS-GRPO	$\mathbf{0.0046 \pm 0.0006}$	$\mathbf{0.0030 \pm 0.0009}$	$\mathbf{0.0030 \pm 0.0015}$	$\mathbf{0.0011 \pm 0.0003}$	$\mathbf{0.0009 \pm 0.0003}$	$\mathbf{0.0006 \pm 0.0004}$	$\mathbf{0.0008 \pm 0.0003}$	$\mathbf{0.0003 \pm 0.0002}$
32	GRPO	0.5541 ± 0.0039	0.5557 ± 0.0053	0.5563 ± 0.0020	0.5577 ± 0.0039	0.5573 ± 0.0016	0.5572 ± 0.0003	0.5578 ± 0.0004	0.5576 ± 0.0002
	IPS-GRPO	0.5306 ± 0.0018	0.5303 ± 0.0009	0.4540 ± 0.0017	0.4356 ± 0.0007	0.4317 ± 0.0009	0.1003 ± 0.0011	0.0914 ± 0.0008	0.0800 ± 0.0006
	MIPS-GRPO	$\mathbf{0.0064 \pm 0.0030}$	$\mathbf{0.0043 \pm 0.0014}$	$\mathbf{0.0033 \pm 0.0006}$	$\mathbf{0.0018 \pm 0.0009}$	$\mathbf{0.0014 \pm 0.0004}$	$\mathbf{0.0010 \pm 0.0002}$	$\mathbf{0.0011 \pm 0.0009}$	$\mathbf{0.0005 \pm 0.0001}$
64	GRPO	0.5725 ± 0.0078	0.5793 ± 0.0099	0.5734 ± 0.0046	0.5753 ± 0.0042	0.5733 ± 0.0026	0.5711 ± 0.0033	0.5713 ± 0.0025	0.5747 ± 0.0040
	IPS-GRPO	0.5347 ± 0.0033	0.5370 ± 0.0019	0.5369 ± 0.0021	0.5265 ± 0.0014	0.4647 ± 0.0017	0.4356 ± 0.0003	0.4330 ± 0.0003	0.2039 ± 0.0939
	MIPS	$\mathbf{0.0097 \pm 0.0009}$	$\mathbf{0.0074 \pm 0.0018}$	$\mathbf{0.0050 \pm 0.0011}$	$\mathbf{0.0036 \pm 0.0008}$	$\mathbf{0.0022 \pm 0.0007}$	$\mathbf{0.0016 \pm 0.0003}$	$\mathbf{0.0009 \pm 0.0001}$	$\mathbf{0.0007 \pm 0.0001}$
128	GRPO	0.6712 ± 0.0006	0.6715 ± 0.0001	0.6211 ± 0.1004	0.6080 ± 0.0742	0.6084 ± 0.0030	0.6026 ± 0.0070	0.5991 ± 0.0034	0.5981 ± 0.0041
	IPS-GRPO	0.5292 ± 0.0025	0.5312 ± 0.0029	0.5291 ± 0.0009	0.5247 ± 0.0018	0.5237 ± 0.0011	0.5200 ± 0.0017	0.4809 ± 0.0003	0.4344 ± 0.0017
	MIPS-GRPO	$\mathbf{0.0185 \pm 0.0019}$	$\mathbf{0.0108 \pm 0.0016}$	$\mathbf{0.0066 \pm 0.0008}$	$\mathbf{0.0040 \pm 0.0003}$	$\mathbf{0.0032 \pm 0.0006}$	$\mathbf{0.0021 \pm 0.0005}$	$\mathbf{0.0015 \pm 0.0003}$	$\mathbf{0.0010 \pm 0.0001}$
256	GRPO	0.6849 ± 0.0000	0.6849 ± 0.0000	0.6849 ± 0.0000	0.6849 ± 0.0000	0.6849 ± 0.0000	0.6849 ± 0.0000	0.6849 ± 0.0000	0.6849 ± 0.0000
	IPS-GRPO	0.5464 ± 0.0029	0.5515 ± 0.0026	0.5459 ± 0.0032	0.5437 ± 0.0009	0.5426 ± 0.0016	0.5371 ± 0.0009	0.5312 ± 0.0027	0.5261 ± 0.0028
	MIPS-GRPO	$\mathbf{0.0270 \pm 0.0026}$	$\mathbf{0.0175 \pm 0.0025}$	$\mathbf{0.0102 \pm 0.0017}$	$\mathbf{0.0073 \pm 0.0005}$	$\mathbf{0.0050 \pm 0.0010}$	$\mathbf{0.0028 \pm 0.0002}$	$\mathbf{0.0024 \pm 0.0005}$	$\mathbf{0.0014 \pm 0.0002}$
512	GRPO	0.6902 ± 0.0000	0.6902 ± 0.0000	0.6902 ± 0.0000	0.6902 ± 0.0000	0.6902 ± 0.0000	0.6902 ± 0.0000	0.6902 ± 0.0000	0.6902 ± 0.0000
	IPS-GRPO	0.5437 ± 0.0029	0.5650 ± 0.0022	0.5721 ± 0.0035	0.5664 ± 0.0014	0.5605 ± 0.0006	0.5585 ± 0.0020	0.5537 ± 0.0003	0.5402 ± 0.0008
	MIPS-GRPO	$\mathbf{0.0428 \pm 0.0039}$	$\mathbf{0.0250 \pm 0.0027}$	$\mathbf{0.0160 \pm 0.0012}$	$\mathbf{0.0106 \pm 0.0006}$	$\mathbf{0.0071 \pm 0.0003}$	$\mathbf{0.0045 \pm 0.0004}$	$\mathbf{0.0032 \pm 0.0002}$	$\mathbf{0.0025 \pm 0.0003}$

Table 5: **Total variation distance on the complete Hypergrid sweep (↓).** Error between the exact learned terminal distribution and the reward-distribution target. Each entry is the mean ± standard deviation across repeated runs.

H	Method	$G = 16$	$G = 32$	$G = 64$	$G = 128$	$G = 256$	$G = 512$	$G = 1024$	$G = 2048$
4	GRPO	0.8332 ± 0.0002	0.8333 ± 0.0000	0.7708 ± 0.1251	0.7709 ± 0.1248	0.6562 ± 0.1197	0.6561 ± 0.1198	0.5982 ± 0.0198	0.5937 ± 0.0208
	IPS-GRPO	0.3314 ± 0.0002	0.2151 ± 0.0037	0.0508 ± 0.0084	$\mathbf{0.0193 \pm 0.0094}$	$\mathbf{0.0111 \pm 0.0007}$	$\mathbf{0.0085 \pm 0.0007}$	$\mathbf{0.0084 \pm 0.0008}$	$\mathbf{0.0051 \pm 0.0019}$
	MIPS-GRPO	$\mathbf{0.0417 \pm 0.0070}$	$\mathbf{0.0260 \pm 0.0119}$	$\mathbf{0.0247 \pm 0.0086}$	0.0252 ± 0.0111	0.0153 ± 0.0054	0.0126 ± 0.0007	0.0112 ± 0.0029	0.0137 ± 0.0014
8	GRPO	0.8837 ± 0.0001	0.8831 ± 0.0014	0.8839 ± 0.0000	0.8839 ± 0.0000	0.8839 ± 0.0000	0.8839 ± 0.0000	0.8838 ± 0.0001	0.8259 ± 0.0670
	IPS-GRPO	0.5626 ± 0.1608	0.2948 ± 0.0110	0.2163 ± 0.0032	0.2116 ± 0.0001	0.1166 ± 0.0064	$\mathbf{0.0236 \pm 0.0042}$	$\mathbf{0.0119 \pm 0.0041}$	$\mathbf{0.0120 \pm 0.0033}$
	MIPS-GRPO	$\mathbf{0.0692 \pm 0.0224}$	$\mathbf{0.0430 \pm 0.0082}$	$\mathbf{0.0360 \pm 0.0099}$	$\mathbf{0.0291 \pm 0.0102}$	$\mathbf{0.0329 \pm 0.0064}$	0.0243 ± 0.0113	0.0170 ± 0.0020	0.0175 ± 0.0038
16	GRPO	0.9597 ± 0.0005	0.9601 ± 0.0001	0.9602 ± 0.0001	0.9603 ± 0.0000	0.9603 ± 0.0000	0.9604 ± 0.0000	0.9604 ± 0.0000	0.9604 ± 0.0000
	IPS-GRPO	0.8293 ± 0.0006	0.8186 ± 0.0020	0.8207 ± 0.0001	0.2922 ± 0.0025	0.2885 ± 0.0006	0.2492 ± 0.0018	0.0629 ± 0.0024	0.0204 ± 0.0028
	MIPS-GRPO	$\mathbf{0.0755 \pm 0.0077}$	$\mathbf{0.0616 \pm 0.0112}$	$\mathbf{0.0617 \pm 0.0173}$	$\mathbf{0.0363 \pm 0.0040}$	$\mathbf{0.0333 \pm 0.0062}$	$\mathbf{0.0268 \pm 0.0087}$	$\mathbf{0.0310 \pm 0.0064}$	$\mathbf{0.0201 \pm 0.0044}$
32	GRPO	0.9179 ± 0.0059	0.9227 ± 0.0039	0.9203 ± 0.0018	0.9215 ± 0.0006	0.9222 ± 0.0001	0.9226 ± 0.0000	0.9226 ± 0.0000	0.9226 ± 0.0000
	IPS-GRPO	0.9009 ± 0.0020	0.9007 ± 0.0006	0.8183 ± 0.0016	0.8137 ± 0.0006	0.8114 ± 0.0011	0.2845 ± 0.0045	0.2526 ± 0.0008	0.2429 ± 0.0008
	MIPS-GRPO	$\mathbf{0.0863 \pm 0.0215}$	$\mathbf{0.0727 \pm 0.0102}$	$\mathbf{0.0633 \pm 0.0065}$	$\mathbf{0.0461 \pm 0.0134}$	$\mathbf{0.0407 \pm 0.0069}$	$\mathbf{0.0342 \pm 0.0049}$	$\mathbf{0.0350 \pm 0.0145}$	$\mathbf{0.0246 \pm 0.0031}$
64	GRPO	0.9246 ± 0.0043	0.9300 ± 0.0066	0.9284 ± 0.0035	0.9282 ± 0.0030	0.9266 ± 0.0018	0.9260 ± 0.0027	0.9264 ± 0.0012	0.9284 ± 0.0034
	IPS-GRPO	0.9002 ± 0.0030	0.9032 ± 0.0028	0.9044 ± 0.0026	0.8959 ± 0.0012	0.8308 ± 0.0024	0.8126 ± 0.0002	0.8122 ± 0.0002	0.4926 ± 0.1538
	MIPS-GRPO	$\mathbf{0.1076 \pm 0.0044}$	$\mathbf{0.0957 \pm 0.0142}$	$\mathbf{0.0789 \pm 0.0084}$	$\mathbf{0.0674 \pm 0.0075}$	$\mathbf{0.0518 \pm 0.0077}$	$\mathbf{0.0439 \pm 0.0033}$	$\mathbf{0.0328 \pm 0.0029}$	$\mathbf{0.0291 \pm 0.0022}$
128	GRPO	0.9868 ± 0.0004	0.9870 ± 0.0001	0.9467 ± 0.0803	0.9401 ± 0.0559	0.9465 ± 0.0019	0.9429 ± 0.0032	0.9401 ± 0.0011	0.9386 ± 0.0028
	IPS-GRPO	0.8905 ± 0.0031	0.8949 ± 0.0029	0.8942 ± 0.0017	0.8927 ± 0.0026	0.8937 ± 0.0017	0.8915 ± 0.0011	0.8501 ± 0.0004	0.8089 ± 0.0022
	MIPS-GRPO	$\mathbf{0.1515 \pm 0.0107}$	$\mathbf{0.1137 \pm 0.0091}$	$\mathbf{0.0872 \pm 0.0056}$	$\mathbf{0.0679 \pm 0.0026}$	$\mathbf{0.0625 \pm 0.0067}$	$\mathbf{0.0492 \pm 0.0070}$	$\mathbf{0.0422 \pm 0.0047}$	$\mathbf{0.0338 \pm 0.0018}$
256	GRPO	0.9955 ± 0.0000	0.9955 ± 0.0000	0.9955 ± 0.0000	0.9955 ± 0.0000	0.9955 ± 0.0000	0.9955 ± 0.0000	0.9955 ± 0.0000	0.9955 ± 0.0000
	IPS-GRPO	0.8975 ± 0.0017	0.9067 ± 0.0028	0.9068 ± 0.0032	0.9065 ± 0.0014	0.9071 ± 0.0018	0.9039 ± 0.0013	0.8997 ± 0.0032	0.8957 ± 0.0019
	MIPS-GRPO	$\mathbf{0.1814 \pm 0.0106}$	$\mathbf{0.1435 \pm 0.0106}$	$\mathbf{0.1100 \pm 0.0095}$	$\mathbf{0.0945 \pm 0.0042}$	$\mathbf{0.0766 \pm 0.0085}$	$\mathbf{0.0568 \pm 0.0028}$	$\mathbf{0.0526 \pm 0.0060}$	$\mathbf{0.0409 \pm 0.0024}$
512	GRPO	0.9985 ± 0.0000	0.9985 ± 0.0000	0.9985 ± 0.0000	0.9985 ± 0.0000	0.9985 ± 0.0000	0.9985 ± 0.0000	0.9985 ± 0.0000	0.9985 ± 0.0000
	IPS-GRPO	0.8967 ± 0.0015	0.9108 ± 0.0018	0.9194 ± 0.0022	0.9186 ± 0.0012	0.9165 ± 0.0003	0.9157 ± 0.0014	0.9130 ± 0.0002	0.9054 ± 0.0014
	MIPS-GRPO	$\mathbf{0.2323 \pm 0.0098}$	$\mathbf{0.1735 \pm 0.0089}$	$\mathbf{0.1390 \pm 0.0061}$	$\mathbf{0.1129 \pm 0.0041}$	$\mathbf{0.0917 \pm 0.0021}$	$\mathbf{0.0720 \pm 0.0039}$	$\mathbf{0.0604 \pm 0.0027}$	$\mathbf{0.0541 \pm 0.0034}$

Table 6: Hypergrid reproducibility settings. H and G are swept; every other value is fixed. Seeds are $1, 2, 3$.

Setting	Value
Grid	$D = 2$, $H \in \{4, 8, 16, 32, 64, 128, 256, 512\}$
Coordinates	$\mathrm{linspace}(-1, 1, H)$
Actions	Increment one coordinate, or stop
Termination	Stop, or all coordinates at $H - 1$
Policy	MLP, $HD \to 128 \to 128 \to D + 1$, ReLU
Optimizer	Adam, step size 10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$
Updates	10,000, one full-batch step each
Batch size G	$\{16, 32, 64, 128, 256, 512, 1024, 2048\}$
Entropy coefficient	0.1

C.2 Molecular synthesis

Table 7: **Hyperparameters for molecular synthesis.** GRPO, IPS-GRPO, and MIPS-GRPO share the same policy architecture and training settings; MIPS-GRPO additionally trains a separate conditional model c_ϕ. RGFN uses its published configuration.

Hyperparameter	Value
Shared	
Policy network	Graph transformer, 5 layers, 4 heads, hidden dim 64
Graph features	Node dim 71, edge dim 4
Output heads	Phase-specific MLPs with GELU
Maximum reactions	4
Reward	$\exp(\beta \cdot \text{proxy})$, $\beta = 8$
Optimizer	Adam
Training iterations	4,000
Group size G	100
Replay buffer	None
Exploration	None (on-policy)
Policy-ratio clip ϵ	0.2
Gradient clip (policy)	10.0
Learning rate (GRPO, IPS-GRPO)	1×10^{-3}
MIPS-GRPO	
Conditional model c_ϕ	Separate network, same architecture
Learning rate (c_ϕ)	1×10^{-3}
Fitting epochs per update (c_ϕ)	4
Gradient clip (c_ϕ)	1.0

Evaluation details. In addition to the metrics in the main text, we report four quantities. Mode candidates are unique molecules with a proxy score above 7 before similarity filtering. Thus, if a method has no mode candidates, it has not sampled any high-reward molecules. Following Koziarski et al. (2024), we also count Murcko scaffolds among molecules with proxy score above 8. For the discovered modes, we report two standard chemistry metrics: quantitative estimate of drug-likeness (QED) and synthetic accessibility (SA). QED ranges from 0 to 1, with higher values indicating greater drug-likeness. SA ranges from 1 to 10, with lower values indicating easier synthesis. Our evaluation differs from Koziarski et al. (2024) in that we count modes using samples from the final policy rather than all molecules visited during training.

Table 8: **Per-seed molecular synthesis results.** Statistics of 50,000 samples from each final policy (3 seeds). Unique fractions are computed over valid samples, and scaffold counts are taken among molecules above the given proxy score. Mode-level quality metrics are averaged over discovered modes and are undefined when no mode is found.

		Diversity							Quality			
Method	Seed	Unique	Unique (%)	Modes	Candidates	Scaffolds (> 7)	Scaffolds (> 8)	Valid (%)	Mean proxy	Mode proxy	Mode QED	Mode SA
GRPO	0	1	0.002	0	0	0	0	100.0	6.60	—	—	—
	1	1	0.002	0	0	0	0	100.0	6.88	—	—	—
	2	1	0.002	0	0	0	0	100.0	6.15	—	—	—
IPS-GRPO	0	6	0.012	0	0	0	0	100.0	3.35	—	—	—
	1	17	0.034	0	0	0	0	100.0	5.97	—	—	—
	2	27	0.055	0	0	0	0	100.0	4.72	—	—	—
RGFN	0	35,711	71.5	19	27	25	0	99.9	4.41	7.12	0.234	3.11
	1	33,306	66.6	17	36	30	0	100.0	4.40	7.22	0.319	3.01
	2	35,216	70.5	18	19	19	0	100.0	3.76	7.16	0.369	2.95
MIPS-GRPO	0	47,022	94.1	71	132	128	1	100.0	4.39	7.30	0.441	2.87
	1	43,105	86.2	72	236	197	1	100.0	4.62	7.21	0.289	2.80
	2	47,255	95.2	68	122	122	0	100.0	4.40	7.15	0.301	2.91

C.3 PHYLOGENETIC INFERENCE

Environment details. We use the first 10 taxa of the DS1 alignment (1,949 sites). Branch lengths are chosen from 50 discrete values, $0.001, 0.002, \ldots, 0.050$, for all methods.

Training details. All four methods share the forward network. GRPO, IPS-GRPO, and MIPS-GRPO each process 4,096 on-policy trajectories per iteration for 10,000 iterations, i.e. 40.96M trajectories without replay. PhyloGFN processes 3,277 on-policy and 819 replayed trajectories per iteration for the same 10,000 iterations, also 40.96M in total. MIPS-GRPO additionally trains a separate conditional model c_ϕ. Table 9 lists all hyperparameters.

Table 9: **Hyperparameters for phylogenetic inference (DS1 subset, 10 taxa).**

Hyperparameter	Value
Shared (all four methods)	
Policy network	Transformer, 6 layers, 4 heads; MLP output heads, hidden 256
Training iterations	10,000
Exploration probability	Linear from 0.5 to 0.001 over 200 iterations
GRPO, IPS-GRPO, MIPS-GRPO	
Group size G	4,096
Policy epochs K	1
Learning rate (policy)	10^{-4} (Adam)
Clip parameter ϵ	0.2
Entropy coefficient β	0
MIPS-GRPO	
Conditional model c_ϕ	MLP, 2 hidden layers of width 128, over valid merges; input features describe the step, the merged subtrees, and the terminal tree
Conditional-model epochs K_c	4
Learning rate (c_ϕ)	10^{-3} (Adam)
PhyloGFN	
Loss / backward policy	Trajectory balance / fixed uniform
Learning rate (model / $\log Z$)	10^{-4} / 10^{-2} (Adam)

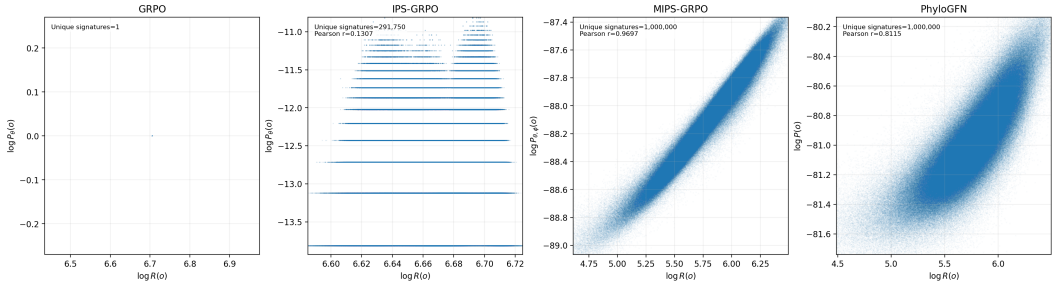


Figure 6: **Outcome log-probability versus log-reward.** Each panel shows 10^6 trees sampled from the final policy. GRPO places all probability on a single tree ($\log P_\theta(o) = 0$), and IPS-GRPO's probabilities barely vary with reward. MIPS-GRPO and PhyloGFN cover a broad set of trees, with MIPS-GRPO's estimated probability $P_{\theta,\phi}(o)$ following the reward most closely. Axis ranges differ across panels.

Table 10: **Per-seed phylogenetic results.** Unique counts among 10^6 samples from each final policy.

Method	Seed	Unique topologies	Unique signatures	Sampling r	Marginal log-lik.	MLL	Shared-tree r
GRPO	0	1	1	—	−4318.36	−4183.27	—
	1	1	1	—	−4318.43	−4182.40	—
	2	1	1	—	−4318.43	−4182.61	—
IPS-GRPO	0	13	291,750	0.131	−4293.30	−4212.82	—
	1	11	248,245	0.150	−4298.43	−4214.99	—
	2	13	250,600	0.140	−4291.36	−4211.82	—
PhyloGFN	0	985,326	1,000,000	0.812	−4367.72	−4672.82	0.942
	1	985,263	1,000,000	0.886	−4363.99	−4671.59	0.922
	2	985,322	1,000,000	0.809	−4367.77	−4672.78	0.943
MIPS-GRPO	0	955,839	1,000,000	0.970	−4372.49	−4663.80	0.971
	1	955,918	1,000,000	0.968	−4373.98	−4664.66	0.952
	2	955,800	1,000,000	0.964	−4372.22	−4663.60	0.964