RANGE-CALIBRATED GFLOWNETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative Flow Networks (GFlowNets) learn policies for sampling compositional objects in proportion to a specified terminal reward. Applying an inverse temperature to reward values controls concentration, but the same temperature can assign different probabilities to a top quantile on different reward landscapes. To specify target concentration, we propose Range-Calibrated GFlowNets (RC-GFN), which use exponential weights on ranks estimated against a policy-independent reference set. Under a uniform reference measure without score ties, the concentration parameter determines the population target's top-quantile mass in closed form, independently of terminal-space size and reward values. The empirical weights are invariant to strictly increasing score transformations and have a user-controlled log-range bound. For arbitrary reference measures and tied scores, we bound finite-reference target error independently of terminal-space size and separate it from policy-fitting error. Experiments on an enumerable environment examine concentration across landscapes and reference sizes. On molecular and biological sequence design tasks, RC-GFN is competitive with the tested temperature-conditional GFlowNets in both mode discovery and top-candidate reward, and outperforms a representative rank-based GFlowNet.

1 INTRODUCTION

Generative Flow Networks (GFlowNets) generate compositional objects through sequences of actions so that the terminating distribution is proportional to a non-negative reward R (Bengio et al., 2021; 2023). Rather than producing a single optimum, GFlowNets repeatedly sample from diverse high-reward modes and yield a reusable policy for downstream candidate generation. GFlowNets have found applications in small-molecule design (Bengio et al., 2021), biological sequence design (Jain et al., 2022), Bayesian structure learning (Deleu et al., 2022), graph combinatorial optimization (Zhang et al., 2023b), and compiler scheduling (Zhang et al., 2023a). High-reward candidates are often rare, and sampling in proportion to R alone can still place most probability mass in lower-reward regions (Shen et al., 2023). Practitioners therefore sample in proportion to R^β (power form) (Bengio et al., 2021; Cretu et al., 2025; Shen et al., 2023) or $e^{\beta R}$ (exponential form) (Koziarski et al., 2024; Kim et al., 2024a). Both forms use an inverse temperature β to control target concentration, with larger values favoring higher-reward candidates. Because β acts directly on reward values, however, this form of control raises two issues.

Issue 1. The concentration induced by β depends on the reward landscape. A temperature chosen for one task need not induce the same concentration on another (Chen & Mauch, 2024; Zhang et al., 2023a; Pandey et al., 2025). Even when two score functions induce the same ordering, changing the numerical spacing of scores can change the mass assigned to high-ranked candidates at the same β. Reward normalization can also change this correspondence (Kim et al., 2024c; Cretu et al., 2025). With too little concentration, even a perfectly fitted GFlowNet can spend most of its sampling budget outside the high-reward region. Conversely, a sharply concentrated target can impede exploration early in training (Chen & Mauch, 2024).

Issue 2. Specifying top-quantile mass requires landscape-dependent calibration. Within a task, users may want to concentrate on high-reward candidates while retaining alternatives for downstream evaluation (Jain et al., 2023). For example, a user may want half of the target probability to lie on the best 1% of candidates. This specifies how much of the sampling budget should reach a region defined by rank, regardless of the numerical reward scale. Translating this request into β

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.

requires knowing how reward values are distributed across ranks. Annealing β changes the target concentration during optimization (Pandey et al., 2025; Kim et al., 2024b), but does not remove this dependence.

Both issues motivate specifying concentration through a property of the target distribution itself. In maximum-entropy reinforcement learning, for example, a target entropy can be used to tune the temperature automatically (Haarnoja et al., 2019). For top-quantile calibration, ranks provide a coordinate independent of numerical score differences. Rank-based GFlowNets use comparisons between candidates (Chen & Mauch, 2024; Pastor-Pérez et al., 2025). An ordering, however, is compatible with many target distributions. The question is how to choose rank weights for a desired top-quantile mass and quantify the error when those ranks are estimated from a finite reference set.

We propose Range-Calibrated GFlowNets (RC-GFN), which assign exponential weights to ranks estimated against a policy-independent reference. The concentration parameter λ bounds the log-weight range. Under a uniform reference measure without score ties, the population target has a closed-form top-quantile mass that depends only on λ and the quantile. The desired mass can be specified before training, with the same interpretation across reward landscapes. We draw a finite reference once and freeze it before training, giving each candidate a fixed positive reward for standard GFlowNet objectives. Our analysis bounds the resulting target approximation error and separates it from error in fitting the policy. Our contributions are as follows.

- We define a positive exponential rank reward using a frozen reference set. Its weights are invariant to strictly increasing score transformations, and λ bounds their log range.
- Under a uniform reference measure without score ties, we derive an exact correspondence between λ and population top-quantile mass, independent of reward values and terminal-space size. For any top set that is neither empty nor the entire terminal space, each attainable mass determines a unique finite nonnegative λ.
- We bound finite-reference target error independently of terminal-space size for arbitrary reference measures and tied scores. Under a uniform reference measure without ties, a sharper mean-square bound gives a sufficient reference size linear in λ at fixed RMSE tolerance. We separate reference error from policy-fitting error and quantify how reference resolution and equal-score groups constrain top-set mass.
- We test concentration control on the HyperGrid environment and evaluate mode discovery on molecular and biological sequence design tasks. Enumeration on the downstream tasks measures reference and policy-fitting errors separately.

2 Related Work

2.1 Temperature control and exploration

Temperature-conditional GFlowNets train a single model for multiple inverse temperatures. Layer-GFN conditions the forward and backward policies on the temperature (Zhang et al., 2023a), and Logit-GFN learns a temperature-dependent function that directly rescales the logits of the policy (Kim et al., 2024b). Both are designed to improve training across multiple temperatures. These methods allow a single model to cover a family of targets defined on reward values, but the concentration specified by a given β still depends on the numerical encoding of the reward and on the distribution of its values. Other methods control exploration or the sampling policy. QGFN constructs a sampling policy with adjustable greediness from action values (Lau et al., 2024), and LGGFN uses an auxiliary GFlowNet to prioritize the exploration of regions in which the loss of the main model is high (Malek et al., 2026). In contrast, RC-GFN constructs positive, bounded terminal rewards from rank coordinates defined by a policy-independent reference set frozen before training. Under a uniform reference measure without score ties, its concentration parameter determines the population target's top-quantile mass in closed form. This target calibration complements work on accurately fitting a prescribed terminating distribution (Silva et al., 2025; Lei et al., 2026).

2.2 RANK-BASED GFLOWNETS

Rank-based GFlowNets replace a reward prescribed in terms of values with comparisons between candidates. OP-GFN (Chen & Mauch, 2024) learns a reward that is consistent with the order of the terminating states within a batch. Its single-objective analysis of neighboring-pair losses characterizes piecewise geometric rank profiles and shows how optimizing the reward range increases concentration on higher-ranked candidates. GR-GFN (Pastor-Pérez et al., 2025) constructs global ranks over observed candidates to resolve inconsistencies in batchwise Pareto comparisons. It permits fitting rank-derived weights directly or after a softmax transformation. In its online setting, these ranks depend on an evolving set of observed candidates. RC-GFN specifies the rank weights before training using a fixed, policy-independent reference set. These weights define a bounded, strictly positive terminal reward for standard GFlowNet losses. Quantile invariance has been studied in information-geometric optimization (IGO) (Ollivier et al., 2017), and under a uniform reference measure without score ties the population target has the same rank weights as exponential ranking in evolutionary computation (Blickle & Thiele, 1996). Our finite-reference bounds quantify how empirical ranks perturb the population target, independently of terminal-space size. A separate fitting term accounts for the discrepancy between the frozen target and the learned terminating distribution.

3 RANGE-CALIBRATED GFLOWNETS

3.1 PRELIMINARIES

Let $(\mathcal{S}, \mathcal{E}_{\mathrm{c}})$ be a finite directed acyclic graph with unique initial state s_0 and a nonempty set $\mathcal{X} \subseteq \mathcal{S}$ of terminating states representing the objects to be generated. Each $x \in \mathcal{X}$ has a terminating edge (x, s_f) to a separate sink state s_f and may also have other children (Bengio et al., 2023). The edges in $\mathcal{E}_{\mathrm{c}}$ are non-terminating. Every state in $\mathcal{S}$ is reachable from s_0 and can reach a terminating state.

A forward policy $P_F(s' \mid s)$ assigns a probability distribution over the outgoing edges, including the terminating edge when available. Let $\mathcal{T}_x$ be the set of complete trajectories from s_0 ending with (x, s_f), and let $P_F(\tau)$ be the product of its forward transition probabilities. Different trajectories may produce the same object. For a positive terminal reward $r : \mathcal{X} \to (0, \infty)$, the terminating distribution on $\mathcal{X}$ and the reward-proportional target are

$$P_F^{\top}(x) = \sum_{\tau \in \mathcal{T}_x} P_F(\tau), \qquad p_r(x) = \frac{r(x)}{\sum_{y \in \mathcal{X}} r(y)}. \tag{1}$$

Training seeks $P_F^{\top} = p_r$ through flow matching, detailed balance, or trajectory balance (Bengio et al., 2021; 2023; Whittammer et al., 2022). Appendix B gives these objectives. RC-GFN defines the terminal reward from comparisons with a frozen reference set.

3.2 RANK COORDINATE

Write $N = |\mathcal{X}|$, and let $R : \mathcal{X} \to \mathbb{R}$ be a fixed score, with larger values preferred. Before training, draw and freeze

$$Y = (Y_1, \ldots, Y_m), \qquad Y_j \overset{\text{iid}}{\sim} \mu, \tag{2}$$

where $m \geq 1$ and the reference measure μ is a policy-independent probability measure on $\mathcal{X}$. Draws are with replacement, and duplicates are retained. Constructing Y requires m evaluations of R. The base case uses uniform μ.

The empirical and population rank coordinates of a candidate x are

$$\widehat{\Phi}(x) = \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}\{R(Y_j) < R(x)\} \in [0, 1] \tag{3}$$

and

$$\Phi(x) = \mu(\{y : R(y) < R(x)\}). \tag{4}$$

Both use strict comparisons. Under uniform μ, $\Phi(x)$ is the fraction of terminal objects that x outranks. The reference sample defines a comparison scale for every terminal object, including candidates not contained in Y.

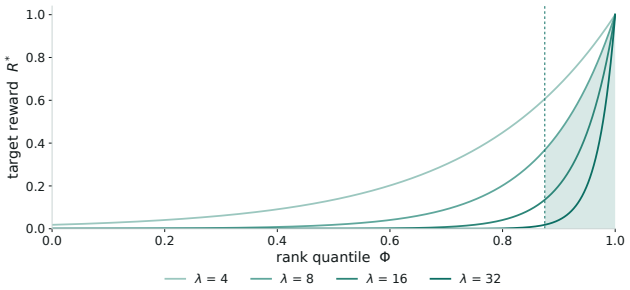

Figure 1: Relationship between the population rank reward R^* and the rank quantile Φ. As a function of Φ, R^* is an exponential curve whose rate of increase is controlled by λ. In the continuous limit, the area under the exponential curve for $\lambda = 8$ that is delimited by the dashed line is proportional to the sampling mass that p^* places on the top rank interval.

Proposition 1. Let g be strictly increasing on $R(\mathcal{X})$. Replacing R by $g \circ R$ does not change $\widehat{\Phi}$ for any candidate.

A strictly increasing g preserves every comparison $R(Y_j) < R(x)$, and hence $\widehat{\Phi}$.

Proposition 2. For each fixed $x \in \mathcal{X}$, $m\widehat{\Phi}(x) \sim \mathrm{Bin}\big(m, \Phi(x)\big)$, and therefore

$$\mathbb{E}\big[\widehat{\Phi}(x)\big] = \Phi(x), \qquad \mathrm{Var}\left(\widehat{\Phi}(x)\right) = \frac{\Phi(x)\big(1-\Phi(x)\big)}{m}. \tag{5}$$

Each indicator in the definition is a Bernoulli variable with parameter $\Phi(x)$, and $Y_1, \ldots, Y_m$ are independent and identically distributed, so their sum follows a binomial distribution.

The empirical coordinate has sampling noise and at most $m + 1$ levels. Section 5 discusses how these affect target accuracy and rank resolution.

3.3 Exponential rank reward

The exponential form assigns the same multiplicative change in weight to the same increase in the rank coordinate. For a finite $\lambda \geq 0$, define

$$t(x) = \lambda\big(\widehat{\Phi}(x) - 1\big), \qquad \widehat{R}(x) = e^{t(x)}, \tag{6}$$

with induced terminal target

$$\widehat{p}(x) = \frac{\widehat{R}(x)}{\widehat{Z}}, \qquad \widehat{Z} = \sum_{y \in \mathcal{X}} \widehat{R}(y). \tag{7}$$

Proposition 3. For any two candidates x and y,

$$\big|\log \widehat{R}(x) - \log \widehat{R}(y)\big| \leq \lambda, \tag{8}$$

and every finite λ guarantees $\widehat{R} \geq e^{-\lambda} > 0$.

Since $\widehat{\Phi} \in [0, 1]$, we have $t \in [-\lambda, 0]$, proving both claims. At $\lambda = 0$ the target is uniform. Increasing λ strengthens its preference for higher coordinates. Proposition 1 also holds for $\widehat{R}$ and $\widehat{p}$, which remain fixed during training once Y and λ are chosen.

3.4 Semantics of λ

To calibrate λ, replace $\widehat{\Phi}$ by Φ and define

$$R^*(x) = e^{\lambda(\Phi(x)-1)}, \qquad p^*(x) = \frac{R^*(x)}{\sum_y R^*(y)}. \tag{9}$$

For fixed λ, Proposition 2 and finite $\mathcal{X}$ give uniform convergence in probability of $\widehat{\Phi}$ to Φ as $m \to \infty$. Continuity of normalized weights then gives $\widehat{p} \to p^*$. Theorem A.3 bounds this finite-reference discrepancy.

First assume uniform μ and no score ties. Then Φ takes the values $0, 1/N, \ldots, (N-1)/N$, giving the same population reward profile across landscapes (Figure 1).

The top-set mass can be computed exactly at finite N. Appendix A.3 gives the continuous rank limit.

Proposition 4. Assume uniform μ and no score ties. For $q \in \{0, 1/N, \ldots, 1\}$, let T_q contain the qN highest-scoring candidates. Its mass under p^* is $(1 - e^{-\lambda q})/(1 - e^{-\lambda})$ for $\lambda > 0$, and q for $\lambda = 0$. It depends only on λ and q.

Index candidates from best to worst as $x_{(1)}, \ldots, x_{(N)}$. Then $\Phi(x_{(j)}) = 1 - j/N$. For $\lambda > 0$, write $a = e^{-\lambda/N}$, so $R^*(x_{(j)}) = a^j$ and

$$p^*(T_q) = \frac{\sum_{j=1}^{qN} a^j}{\sum_{j=1}^{N} a^j} = \frac{1 - a^{qN}}{1 - a^N} = \frac{1 - e^{-\lambda q}}{1 - e^{-\lambda}}. \tag{10}$$

At $\lambda = 0$, all weights are equal.

For $0 < q < 1$, each desired mass in $[q, 1)$ determines a unique finite $\lambda \geq 0$ (Proposition A.4). For example, choosing $\lambda = 1/q$ gives

$$p^*(T_q) = \frac{1 - e^{-1}}{1 - e^{-\lambda}}. \tag{11}$$

This is about 0.632 for $q = 0.1$ or 0.01. With ties and uniform μ, let T be a nonempty proper top set comprising complete score groups. Its attainable masses for finite $\lambda \geq 0$ are exactly $[|T|/N, 1)$, and each determines a unique λ. Calibration depends on the ordered group sizes, not on score values or their spacing (Appendix G).

3.5 Finite-reference accuracy

The frozen target $\widehat{p}$ implements the rank weighting with estimated coordinates. For any fixed reference measure μ, reference size m, and $\delta \in (0, 1)$, Theorem A.3 gives, with probability at least $1 - \delta$ over the iid reference draw,

$$\mathrm{TV}(\widehat{p}, p^*) \leq \tanh\left(\lambda \sqrt{\frac{\log(2/\delta)}{8m}}\right).$$

The bound holds simultaneously for all finite $\lambda \geq 0$ and all terminal sets, allowing nonuniform reference measures and tied scores.

For fixed λ under uniform μ without ties, Proposition A.6 gives $\mathbb{E}_Y \, \mathrm{TV}(\widehat{p}, p^*)^2 \leq \lambda/(4m)$. In particular, the RMSE of $\widehat{p}(T_q)$ about the population mass in Proposition 4 is at most $\frac{1}{2}\sqrt{\lambda/m}$. Thus $m \geq \lambda/(4\varepsilon^2)$ suffices for RMSE at most ε in this setting. Both reference bounds are independent of N. Section 5 relates them to reference selection and policy fitting.

3.6 Algorithm

Algorithm 1 takes λ as input and keeps it fixed during training. Under the assumptions of Proposition 4, λ can be calibrated to a desired mass on a specified top quantile. We tune λ separately for each downstream task (Appendix F).

4 Experiments

We examine concentration control on HyperGrid and evaluate candidate discovery on molecular and biological sequence design tasks. For the downstream tasks, we tune λ separately, using 40 on QM9, 150 on sEH, 25 on TFBind10 and 400 on RNA. We group the compared methods into three classes

Algorithm 1 RC-GFN working procedure

Input terminal space $\mathcal{X}$, reward R, concentration parameter $0 \leq \lambda < \infty$, top quantile $q \in (0, 1]$ for diagnostics, reference-error tolerance ε, reference measure μ (uniform on $\mathcal{X}$ in the base case), training objective (Appendix B), number of training steps T

Output forward policy P_F

1: Choose $m \geq 1$, starting from $m \gtrsim \lambda$, and assess the reference-error tolerance ε using the finite-reference bounds in Section 3.5
2: Draw $Y = (Y_1, \ldots, Y_m) \overset{\text{iid}}{\sim} \mu$, evaluate $R(Y_1), \ldots, R(Y_m)$, sort the values, and freeze them
3: If uniform samples are available, let M be their number and c the number of samples attaining the highest score in the batch, and diagnose ties in the high-scoring region with $\widehat{r} = c/M$. These samples may be shared with Y
4: If μ is uniform, if $q < \max\{1/(m+1), \widehat{r}\}$, flag a possible effect of reference resolution or ties. This condition is a heuristic screen. If the reference design is chosen using these diagnostics, treat all inspected samples as pilot data and independently draw, evaluate, sort and freeze the final iid reference Y after fixing m, regardless of the diagnostic outcome
5: **for** $t = 1, \ldots, T$ **do**
6: Sample a batch of complete trajectories from the current policy, and let $\mathcal{B}$ be the set of their terminating states
7: For each $x \in \mathcal{B}$, evaluate $R(x)$ and obtain $\widehat{\Phi}(x)$ by binary search against the sorted reference values
8: $\widehat{R}(x) \leftarrow e^{\lambda(\widehat{\Phi}(x)-1)}$
9: Update the parameters with the chosen training objective
10: **end for**
11: Sample from the trained P_F

according to how they control concentration. The downstream rank-based comparison uses OP-GFN (Chen & Mauch, 2024) and RC-GFN. Figure 6 also compares GR-GFN (Pastor-Pérez et al., 2025) on HyperGrid. The temperature-controlled GFlowNets are the unconditional GFN-β with a fixed inverse temperature (Shen et al., 2023; Bengio et al., 2021), Layer-GFN (Zhang et al., 2023a), and Logit-GFN (Kim et al., 2024b). The reward-maximization methods are MARS (Xie et al., 2021), which is based on Markov chain Monte Carlo (MCMC) sampling, and three reinforcement learning methods, A2C (Mnih et al., 2016), SQL (Haarnoja et al., 2017) and PPO (Schulman et al., 2017). Unless stated otherwise, results are means $\pm$ 95% confidence intervals over five random seeds. Experimental details are given in Appendix H.

4.1 HyperGrid

The synthetic HyperGrid environment (Bengio et al., 2021) has as its terminal space a D-dimensional grid with side length H. We design three reward landscapes, namely an indicator landscape that takes only three reward values and therefore contains many ties, a cosine landscape, and a variant of the cosine landscape without its Gaussian envelope, which we call spread. The grid sizes cover $(D, H) \in \{(2, 64), (3, 32)\}$. We use HyperGrid to answer the following five research questions (RQs).

- RQ1 Does one value of λ give the same concentration across reward landscapes?
- RQ2 Over what range does the declared concentration match the obtained one?
- RQ3 How does top-set mass error depend on reference size?
- RQ4 Can reference and tie diagnostics identify concentration constraints before training?
- RQ5 How does concentration evolve when the rank target is fixed before training?

We answer RQ1 and RQ2 below and address RQ3 to RQ5 in Appendix C.

Figure 2(a) answers RQ1. With $\lambda = 10$ and $m = 300$, RC-GFN attains similar top-10% masses on the six landscapes, close to the no-tie benchmark from Proposition 4. The differences across landscapes are comparable to the variation across seeds, which includes both reference draws and policy training. At a fixed inverse temperature, GFN-β attains markedly different masses across

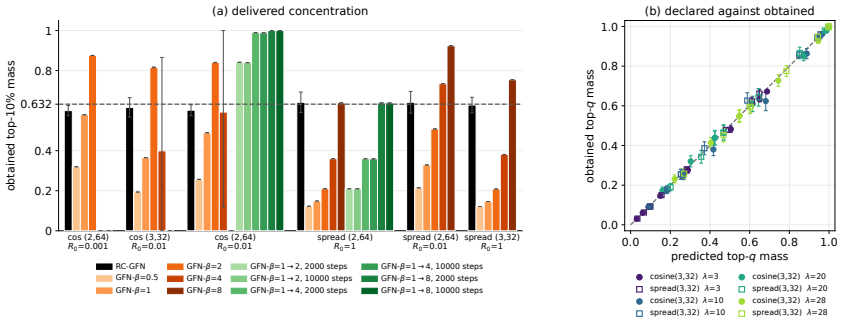

Figure 2: Concentration obtained on HyperGrid. (a) Obtained top 10% sampling mass on six reward landscapes. The dashed line marks the no-tie population benchmark from Proposition 4, approximately 0.632. (b) Obtained top q sampling mass against the same benchmark for four values of λ on the cosine and spread landscapes. The dashed line indicates equality. The landscapes have score ties, so agreement with this benchmark is assessed empirically.

landscapes. With a large inverse temperature, its obtained top-10% mass is especially small on the cosine landscapes. The tested annealing schedules also yield different masses on the cosine and spread landscapes, and lengthening the schedule changes them little. Figure 2(b) answers RQ2 using four values of λ, two landscapes and nine values of q. The measured masses follow the benchmark over the tested range from nearly zero to nearly one, with a small downward offset on the cosine landscape and no systematic growth of the deviation with λ or q. The formula is exact under the assumptions of Proposition 4. These experiments assess its agreement with the learned distributions on tied landscapes.

4.2 Molecular design

We compare the methods on two small-molecule design tasks, QM9 and sEH, in which molecules are assembled from building blocks (Shen et al., 2023; Chen & Mauch, 2024; Kim et al., 2024b). The QM9 task ($|\mathcal{X}| = 161{,}051$) generates small molecular graphs in which each molecule consists of 5 building blocks chosen from 11 types, each with 2 attachment sites, and its oracle is a neural network trained on density functional theory data to predict the HOMO-LUMO gap. The sEH task ($|\mathcal{X}| = 34{,}012{,}224$) also generates small molecular graphs, in which each molecule consists of 6 building blocks chosen from 18 types, again each with 2 attachment sites. Following Kim et al. (2024b), its oracle is a gradient boosting regressor that predicts the binding affinity to soluble epoxide hydrolase, trained on the outputs of the graph neural network of Bengio et al. (2021) as labels. Each method has a budget of 64,000 oracle calls, which corresponds to 2,000 training iterations with a batch size of 32. RC-GFN spends m of these calls on constructing its frozen reference set and the remaining budget on policy training, and the reference calls are included in the cumulative oracle-call axis in Figure 3. Figure 3(a) and (b) show cumulative performance during training, and Table 1 reports the performance evaluated on 10,000 molecules sampled after training.

In Figure 3(a) and (b), RC-GFN is competitive with the strongest baselines on QM9. Its final cumulative mode count is slightly below the best, with an advantage at intermediate call counts. On sEH it leads in cumulative mode count after approximately 15,000 oracle calls. The ratios of the oracle budget to terminal-space size are $64{,}000/161{,}051$ for QM9 and $64{,}000/34{,}012{,}224$ for sEH. RC-GFN's curves additionally count reference calls and modes found during reference construction. Layer-GFN and Logit-GFN aggregate discoveries over the temperatures sampled during training. Within the oracle budget, RC-GFN discovers more cumulative modes than OP-GFN on both tasks. OP-GFN learns reward magnitudes through order comparisons, whereas RC-GFN fits a reward specified before training. OP-GFN's order-preserving optimization can steepen the learned rank profile, while RC-GFN instead keeps this profile fixed through Y and λ, separating target specification from policy fitting. In Table 1, RC-GFN has the highest mean post-training mode count on both tasks. Its mean top-100 reward matches the best temperature-conditional result on QM9 at the reported precision and exceeds both temperature-conditional results on sEH.

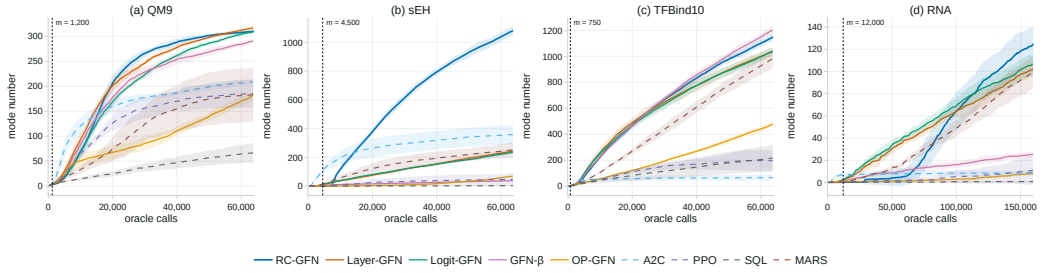

Figure 3: Performance of RC-GFN and the baselines during training on the (a) QM9 and (b) sEH molecular design tasks and the (c) TFBind10 and (d) RNA biological sequence design tasks. The horizontal axis shows the number of oracle calls made during reference-set sampling (RC-GFN only) or training, and the vertical axis shows the cumulative number of modes discovered during reference-set sampling (RC-GFN only) or training. The black dotted line marks the number of oracle calls spent by RC-GFN on constructing the reference set, with reference-set construction to its left and RC-GFN training to its right.

Table 1: Performance on the QM9 and sEH molecular design tasks and the TFBind10 and RNA biological sequence design tasks, evaluated on samples drawn after training. The best is shown in bold and the second best is underlined. The symbol ↑ indicates that higher values are better.

	QM9		sEH		TFBind10		RNA	
Method	Modes (↑)	Top-100 reward (↑)	Modes (↑)	Top-100 reward (↑)	Modes (↑)	Top-100 reward (↑)	Modes (↑)	Top-100 reward (↑)
A2C	0.20 ± 0.39	$\mathbf{0.80 \pm 0.02}$	28.40 ± 24.36	6.55 ± 0.79	0.80 ± 0.39	0.90 ± 0.14	0.60 ± 0.78	0.80 ± 0.16
PPO	4.60 ± 9.02	0.58 ± 0.10	0.00 ± 0.00	6.52 ± 0.30	0.20 ± 0.39	0.17 ± 0.08	3.20 ± 5.79	0.75 ± 0.08
SQL	14.60 ± 5.28	0.52 ± 0.01	0.80 ± 1.14	6.40 ± 0.09	45.40 ± 23.20	0.68 ± 0.07	0.00 ± 0.00	0.64 ± 0.03
MARS	48.60 ± 12.95	0.61 ± 0.03	$\underline{95.20 \pm 40.31}$	$\mathbf{6.94 \pm 0.02}$	197.60 ± 19.77	0.87 ± 0.02	10.60 ± 4.00	0.91 ± 0.01
GFN-β	80.00 ± 4.92	0.62 ± 0.01	6.80 ± 1.69	6.68 ± 0.02	276.60 ± 14.87	0.88 ± 0.01	2.60 ± 1.82	0.81 ± 0.00
Logit-GFN	157.00 ± 4.38	$\underline{0.66 \pm 0.00}$	50.00 ± 8.20	6.86 ± 0.01	$\mathbf{329.80 \pm 17.28}$	$\mathbf{0.91 \pm 0.02}$	10.80 ± 4.44	$\mathbf{0.92 \pm 0.00}$
Layer-GFN	$\underline{158.40 \pm 10.14}$	$\underline{0.66 \pm 0.00}$	53.40 ± 10.55	6.86 ± 0.01	$\underline{311.00 \pm 22.28}$	0.89 ± 0.01	$\underline{15.40 \pm 2.29}$	$\mathbf{0.92 \pm 0.00}$
OP-GFN	65.60 ± 6.69	0.59 ± 0.01	46.00 ± 6.41	6.85 ± 0.01	105.00 ± 7.15	0.79 ± 0.01	2.00 ± 1.64	0.79 ± 0.00
RC-GFN	$\mathbf{177.60 \pm 4.70}$	$\underline{0.66 \pm 0.00}$	$\mathbf{235.40 \pm 19.33}$	$\underline{6.93 \pm 0.01}$	299.00 ± 12.27	$\underline{0.90 \pm 0.01}$	$\mathbf{27.80 \pm 4.82}$	$\underline{0.92 \pm 0.01}$

4.3 Biological sequence design

Additionally, we compare the methods on two biological sequence design tasks, TFBind10 and RNA, in which nucleic acid sequences are constructed one position at a time (Chen & Mauch, 2024; Kim et al., 2024c;b). TFBind10 ($|\mathcal{X}| = 1{,}048{,}576$) generates DNA sequences of length 10 by prepending or appending one nucleotide to the current sequence at each step. Its oracle is the binding free energy of the yeast transcription factor Pho4 measured by Le et al. (2018), in which the ten nucleotides flank a fixed E-box core sequence and all 4^{10} combinations were measured experimentally. The RNA task ($|\mathcal{X}| = 268{,}435{,}456$) generates RNA sequences of length 14 in the same way, and its oracle is the hybridization binding energy between the sequence and a fixed RNA target sequence computed with ViennaRNA (Lorenz et al., 2011). The oracle budgets are 64,000 calls on TFBind10 and 160,000 calls on RNA, corresponding to 2,000 and 5,000 training iterations with a batch size of 32, respectively. As on the molecular tasks, RC-GFN spends m of these calls on its frozen reference set and the remainder on policy training, and the reference calls are included in the cumulative oracle-call axis in Figure 3. Figure 3(c) and (d) show cumulative performance during training, and Table 1 reports the performance evaluated on 10,000 sequences sampled after training.

In Figure 3(c) and (d), the policy-training call count on TFBind10 is small relative to its design space ($64{,}000/1{,}048{,}576$), but the mode set of this task is not filtered for diversity, and one mode is often a near neighbor of another. The cumulative number of discovered modes therefore still reflects the extent of exploration to a considerable degree. On this task RC-GFN is second only to the unconditional baseline with a fixed inverse temperature, and the difference between the two lies within the variation across seeds. The policy-training call count on RNA is also small relative to its design space ($160{,}000/268{,}435{,}456$), and its mode set is filtered to local optima. RC-GFN attains the highest cumulative number of discovered modes on RNA. The reference adds $12{,}000$ oracle calls before policy training and hits no mode on any seed. RC-GFN initially trails the temperature-conditional methods on the cumulative-call axis and catches up later. RC-GFN discovers more

cumulative modes than OP-GFN on both tasks under the implementations and budgets used here, which is consistent with the mechanism described in Section 4.2. In Table 1, RC-GFN has the highest mean mode count on RNA, and its mean top-100 reward matches those of the temperature-conditional methods at the reported precision. On TFBind10, its mean mode count lies below both temperature-conditional methods, while its mean top-100 reward lies between theirs.

5 Discussion

Under a uniform reference measure without score ties, Proposition 4 allows λ to be chosen from a desired population mass on a top quantile, independently of reward values and terminal-space size. For candidate discovery, the preferred concentration can depend on the task and the budget available to fit the target. With tied scores, calibration under uniform μ uses the score-group sizes. For a general reference measure μ, the population mass also depends on the reference mass of each group (Appendix G). The calibration concerns the order induced by R. Its usefulness therefore depends on how well that order reflects the intended objective.

At fixed total variation tolerance and confidence, Theorem A.3 gives a sufficient reference size of order λ^2 for arbitrary μ and tied scores. Under uniform μ without ties, Proposition A.6 gives a sufficient size of order λ at fixed root mean square TV tolerance, or at fixed TV tolerance and confidence by Markov's inequality. The pointwise calculation in (26) also motivates $m \gtrsim \lambda$, while the mean-square bound controls the normalized target. Once Y and λ are fixed, the reference error remains fixed during policy training. Appendix E examines the tradeoff between reference accuracy and policy training under a fixed combined call budget.

Theorem A.3 assumes one iid draw with fixed m, μ and holds simultaneously for all finite $\lambda \geq 0$, so λ may be chosen after seeing Y. Selecting a reference after repeated screening requires accounting for that selection. Appendix C.1 examines top-mass error as m varies at fixed λ. A finite reference also merges candidates into at most $m+1$ rank levels, within which increasing λ cannot distinguish candidates. Under uniform μ and no score ties, objects above every reference point share coordinate 1 and occupy an expected fraction at most $1/(m+1)$ of the space. The reference misses a top set of qN objects with probability $(1-q)^m$ (Proposition A.7). Actual score ties impose a separate mass constraint. Any $L \leq k$ objects from a highest-score group of size k receive exactly L/k of the group's mass, hence at most L/k of the total mass, under either target. Algorithm 1 uses $1/(m+1)$ and the sample-maximum fraction $\widehat{r}$ as heuristic screens for these effects. Appendix C.2 evaluates these diagnostics.

Once the reference is frozen, GFlowNet training seeks to match $\widehat{p}$. Under the positivity and full support assumptions of Proposition A.8, exact zero expected loss gives $P_F^\top = \widehat{p}$. For any learned policy and $\delta \in (0,1)$, Corollary A.9 gives, with probability at least $1-\delta$ over the iid reference draw,

$$\mathrm{TV}(P_F^\top, p^*) \leq \mathrm{TV}(P_F^\top, \widehat{p}) + \tanh\left(\lambda\sqrt{\frac{\log(2/\delta)}{8m}}\right).$$

For uniform μ without score ties, this also bounds the error in the top-quantile mass calibrated by Proposition 4. Appendix D evaluates all three distributions on the downstream tasks. At $m = 30\lambda$, mean reference TV is 0.036–0.055, while mean policy-fitting TV is 0.331–0.865. These measurements locate most of the remaining discrepancy in the terminating distribution on the fitting side. On RNA, fitting TV falls from 0.865 to 0.201 when aggregated by score level. The full TV also measures nonuniform allocation among tied objects.

6 Conclusion

RC-GFN gives its concentration parameter a direct interpretation through population top-quantile mass under a uniform reference measure without score ties. Its exponential weighting of empirical ranks defines a fixed terminal target with a controlled log range and invariance to strictly increasing score transformations. Our finite-reference bounds quantify the approximation for arbitrary reference measures and tied scores, and distinguish it from the error of fitting the target with a GFlowNet. HyperGrid experiments examine concentration across landscapes and reference sizes. On the four downstream tasks, RC-GFN achieves the highest mean post-training mode count on three and competitive top-100 rewards against the tested temperature-conditional GFlowNets.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, assisting with translation, interpreting results, and assisting in the writing of proofs. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, or proposing or refining hypotheses, and supporting qualitative and thematic data analysis is not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27381–27394. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/e614f646836aaed9f89ce58e837e2310-Abstract.html.

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. ISSN 1533-7928. URL http://jmlr.org/papers/v24/22-0364.html.

Tobias Blickle and Lothar Thiele. A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evolutionary Computation*, 4(4):361–394, December 1996. ISSN 1063-6560. doi: 10.1162/evco.1996.4.4.361. URL https://ieeexplore.ieee.org/abstract/document/6791924.

Yihang Chen and Lukas Mauch. Order-Preserving GFlowNets. In *International Conference on Learning Representations*, volume 2024, pp. 53088–53125, May 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ea06e6e9e80f1c3d382317fff67041ac-Abstract-Conference.html.

Miruna Cretu, Charles Harris, Ilia Igashov, Arne Schneuing, Marwin Segler, Bruno Correia, Julien Roy, Emmanuel Bengio, and Pietro Lio. SynFlowNet: Design of Diverse and Novel Molecules with Synthesis Constraints. *International Conference on Learning Representations*, 2025:47101–47126, May 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7495fa446f10e9edef6e47b2d327596e-Abstract-Conference.html.

Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 518–528. PMLR, August 2022. URL https://proceedings.mlr.press/v180/deleu22a.html.

A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator. *The Annals of Mathematical Statistics*, 27(3):642–669, September 1956. ISSN 0003-4851, 2168-8990. doi: 10.1214/aoms/1177728174. URL https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-27/issue-3/Asymptotic-Minimax-Character-of-the-Sample-Distribution-Function-and-of/10.1214/aoms/1177728174.full.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement Learning with Deep Energy-Based Policies. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1352–1361. PMLR, July 2017. URL https://proceedings.mlr.press/v70/haarnoja17a.html.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic Algorithms and Applications, January 2019. URL http://arxiv.org/abs/1812.05905. arXiv:1812.05905 [cs.LG].

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological Sequence Design with GFlowNets. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 9786–9801. PMLR, June 2022. URL https://proceedings.mlr.press/v162/jain22a.html.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-García, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-Objective GFlowNets. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 14631–14653. PMLR, July 2023. URL https://proceedings.mlr.press/v202/jain23a.html.

Hyeonah Kim, Minsu Kim, Sanghyeok Choi, and Jinkyoo Park. Genetic-guided GFlowNets for Sample Efficient Molecular Optimization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 42618–42648. Curran Associates, Inc., 2024a. doi: 10.52202/079017-1350. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4b25c000967af9036fb9b207b198a626-Abstract-Conference.html.

Minsu Kim, Joohwan Ko, Taeyoung Yun, Dinghuai Zhang, Ling Pan, Woo Chang Kim, Jinkyoo Park, Emmanuel Bengio, and Yoshua Bengio. Learning to scale logits for temperature-conditional GFlowNets. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pp. 24248–24270, Vienna, Austria, July 2024b. JMLR.org.

Minsu Kim, Taeyoung Yun, Emmanuel Bengio, Dinghuai Zhang, Yoshua Bengio, Sungsoo Ahn, and Jinkyoo Park. Local Search GFlowNets. *International Conference on Learning Representations*, 2024:42296–42313, May 2024c. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/b8d32f60f69cc1fe461be9712af1528d-Abstract-Conference.html.

Michał Koziarski, Andrei Rekesh, Dmytro Shevchuk, Almer van der Sloot, Piotr Gaiński, Yoshua Bengio, Cheng-Hao Liu, Mike Tyers, and Robert Batey. RGFN: Synthesizable Molecular Generation Using GFlowNets. In *Advances in Neural Information Processing Systems*, volume 37, pp. 46908–46955. Curran Associates, Inc., 2024. doi: 10.52202/079017-1488. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/53704142f230054140418ecd8857f391-Abstract-Conference.html.

Elaine Lau, Stephen Zhewen Lu, Ling Pan, Doina Precup, and Emmanuel Bengio. QGFN: Controllable Greediness with Action Values. In *Advances in Neural Information Processing Systems*, volume 37, pp. 81645–81676. Curran Associates, Inc., 2024. doi: 10.52202/079017-2594. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/948d8ba4e30c8c3a800cf436b31f376e-Abstract-Conference.html.

Daniel D. Le, Tyler C. Shimko, Arjun K. Aditham, Allison M. Keys, Scott A. Longwell, Yaron Orenstein, and Polly M. Fordyce. Comprehensive, high-resolution binding energy landscapes reveal context dependencies of transcription factor binding. *Proceedings of the National Academy of Sciences*, 115(16):E3702–E3711, April 2018. doi: 10.1073/pnas.1715888115. URL https://www.pnas.org/doi/abs/10.1073/pnas.1715888115.

Zengxiang Lei, Ananth Shreekumar, Jonathan Rosenthal, Ruoyu Song, Alvaro A. Cardenas, Daniel J. Fremont, Dongyan Xu, Satish Ukkusuri, and Z. Berkay Celik. Stable GFlowNets with TV Monitoring and Probabilistic Guarantees, August 2026. URL http://arxiv.org/abs/2605.01729. arXiv:2605.01729 [cs.LG].

Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, 6(1):26, November 2011. ISSN 1748-7188. doi: 10.1186/1748-7188-6-26. URL https://doi.org/10.1186/1748-7188-6-26.

Idriss Malek, Aya Laajil, Abhijith Sharma, Eric Moulines, and Salem Lahlou. Loss-Guided Auxiliary Agents for Overcoming Mode Collapse in GFlowNets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(29):24326–24334, March 2026. ISSN 2374-3468. doi: 10.1609/aaai.v40i29.39613. URL https://ojs.aaai.org/index.php/AAAI/article/view/39613.

P. Massart. The Tight Constant in the Dvoretzky-Kiefer-Wolfowitz Inequality. *The Annals of Probability*, 18(3):1269–1283, July 1990. ISSN 0091-1798, 2168-894X. doi: 10.1214/aop/1176990746. URL https://projecteuclid.org/journals/annals-of-probability/volume-18/issue-3/The-Tight-Constant-in-the-Dvoretzky-Kiefer-Wolfowitz-Inequality/10.1214/aop/1176990746.full.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1928–1937. PMLR, June 2016. URL https://proceedings.mlr.press/v48/mniha16.html.

Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. *Journal of Machine Learning Research*, 18(18):1–65, 2017. ISSN 1533-7928. URL http://jmlr.org/papers/v18/14-467.html.

Mohit Pandey, Gopeshh Subbaraj, Artem Cherkasov, Martin Ester, and Emmanuel Bengio. Pretraining Generative Flow Networks with Inexpensive Rewards for Molecular Graph Generation, June 2025. URL http://arxiv.org/abs/2503.06337. arXiv:2503.06337 [cs.LG].

Lluís Pastor-Pérez, Javier Alonso-Garcia, and Lukas Mauch. Global-Order GFlowNets. In *arXiv.org*, April 2025. URL https://arxiv.org/abs/2504.02968v1.

Kenneth J. Rothman and John D. Boice, Jr. *Epidemiologic Analysis with a Programmable Calculator*. Epidemiology Resources, December 1982.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs.LG].

Max W. Shen, Emmanuel Bengio, Ehsan Hajiramezanali, Andreas Loukas, Kyunghyun Cho, and Tommaso Biancalani. Towards Understanding and Improving GFlowNet Training. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 30956–30975. PMLR, July 2023. URL https://proceedings.mlr.press/v202/shen23a.html.

Tiago Silva, Rodrigo Alves, Eliezer de Souza da Silva, Amauri Souza, Vikas Garg, Samuel Kaski, and Diego Mesquita. When do GFlowNets learn the right distribution? *International Conference on Learning Representations*, 2025:65742–65767, May 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a48f8928f78a58399ef0049453c14b02-Abstract-Conference.html.

Sam Sinai, Richard Wang, Alexander Whatley, Stewart Slocum, Elina Locane, and Eric D. Kelsic. AdaLead: A simple and robust adaptive greedy search algorithm for sequence design, October 2020. URL http://arxiv.org/abs/2010.02141. arXiv:2010.02141 [cs.LG].

Esmeralda S Whitammer, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in GFlowNets. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5955–5967. Curran Associates, Inc., 2022. doi: 10.52202/068431-0431. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/27b51baca8377a0cf109f6ecc15a0f70-Abstract-Conference.html.

Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. MARS: Markov Molecular Sampling for Multi-objective Drug Discovery, March 2021. URL http://arxiv.org/abs/2103.10432. arXiv:2103.10432 [q-bio.BM].

David W. Zhang, Corrado Rainone, Markus Peschl, and Roberto Bondesan. Robust Scheduling with GFlowNets, February 2023a. URL http://arxiv.org/abs/2302.05446. arXiv:2302.05446 [cs.AI].

Dinghuai Zhang, Hanjun Dai, Esmeralda S Whitammer, Aaron Courville, Yoshua Bengio, and Ling Pan. Let the Flows Tell: Solving Graph Combinatorial Problems with GFlowNets. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11952–11969. Curran Associates, Inc., 2023b. doi: 10.52202/075280-0522. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/27571b74d6cd650b8eb6cf1837953ae8-Abstract-Conference.html.

A THEORY

A.1 SETUP

Let $\mathcal{X}$ be finite and nonempty, $N = |\mathcal{X}|$, and $R : \mathcal{X} \to \mathbb{R}$ a fixed score, with larger values preferred. Let $Y_1, \dots, Y_m$ be iid with fixed law μ on $\mathcal{X}$, $m \geq 1$, and retain duplicates. Use the coordinates and targets $\widehat{p}, p^*$ of Section 3 for finite $\lambda \geq 0$. Unless stated otherwise, μ may be nonuniform and R may have ties. Both targets sum one weight per terminal object. The reference measure μ enters only through the coordinates.

Probability is over Y. Conditional on Y, $\widehat{R}$ and $\widehat{p}$ are fixed. The notation $P_F^\top$ denotes a forward policy's terminating distribution. For distributions p, r on $\mathcal{X}$,

$$\begin{aligned} \mathrm{TV}(p, r) &:= \frac{1}{2} \sum_x |p(x) - r(x)| = \sup_{A \subseteq \mathcal{X}} |p(A) - r(A)|, \\ D_{\mathrm{KL}}(p \| r) &:= \sum_x p(x) \log \frac{p(x)}{r(x)}. \end{aligned} \tag{12}$$

All logarithms are natural, and the KL expressions below involve positive distributions.

A.2 FINITE-REFERENCE APPROXIMATION

Lemma A.1 (Strict-CDF deviation)**.** For every $\epsilon > 0$,

$$\Pr_Y \left(\sup_{x \in \mathcal{X}} |\widehat{\Phi}(x) - \Phi(x)| > \epsilon \right) \leq 2e^{-2m\epsilon^2}. \tag{13}$$

Proof. Put $Z_j = R(Y_j)$, $F(t) = \Pr(Z_1 \leq t)$, and $F_m(t) = m^{-1} \sum_j \mathbb{1}\{Z_j \leq t\}$. With probability at least $1 - 2e^{-2m\epsilon^2}$, the DKW–Massart inequality (Dvoretzky et al., 1956; Massart, 1990) gives $\sup_t |F_m(t) - F(t)| \leq \epsilon$. On this event,

$$|F_m(t-) - F(t-)| = \lim_{n \to \infty} |F_m(t - 1/n) - F(t - 1/n)| \leq \epsilon$$

for every t. Since $F(t-) = \Pr(Z_1 < t)$ and $F_m(t-) = m^{-1} \sum_j \mathbb{1}\{Z_j < t\}$, evaluation at $t = R(x)$ gives the result, including when F has atoms. □

Normalization removes common shifts of the log weights, so we measure perturbations by their oscillation $\mathrm{osc}(d) := \max_x d(x) - \min_x d(x)$.

Lemma A.2 (Exponential-weight perturbation)**.** Let p be a strictly positive distribution on $\mathcal{X}$, and let $d : \mathcal{X} \to \mathbb{R}$. Define

$$p_d(x) := \frac{p(x) e^{d(x)}}{\mathbb{E}_p e^d}, \qquad \omega := \mathrm{osc}(d). \tag{14}$$

Then

$$\max_x \left| \log \frac{p_d(x)}{p(x)} \right| \leq \omega, \tag{15}$$

$$\max\{D_{\mathrm{KL}}(p \| p_d), D_{\mathrm{KL}}(p_d \| p)\} \leq \min\{\omega, \omega^2/8\}, \tag{16}$$

$$\mathrm{TV}(p, p_d) \leq \tanh(\omega/4). \tag{17}$$

For every $A \subseteq \mathcal{X}$, writing $a = p(A)$ gives

$$\frac{a}{a + (1-a)e^\omega} \leq p_d(A) \leq \frac{a e^\omega}{1 - a + a e^\omega}. \tag{18}$$

Proof. If $\omega = 0$, then d is constant and $p_d = p$. Otherwise, write $\ell = \min_x d(x)$ and $h = \max_x d(x)$. Since $\log \mathbb{E}_p e^d \in [\ell, h]$, the log ratio $\log(p_d(x)/p(x)) = d(x) - \log \mathbb{E}_p e^d$ has absolute value at most $h - \ell = \omega$. Taking expectations proves the linear KL bounds.

For the quadratic bound, set $\psi(t) = \log \mathbb{E}_p e^{td}$. Then $\psi'(t) = \mathbb{E}_{p_{td}} d$ and $\psi''(t) = \mathrm{Var}_{p_{td}}(d) \le \omega^2/4$, since variance is at most the mean squared distance from the midpoint $(\ell + h)/2$. Thus

$$D_{\mathrm{KL}}(p\|p_d) = \psi(1) - \psi(0) - \psi'(0) = \int_0^1 (1-t)\psi''(t)\,dt \le \frac{\omega^2}{8}.$$

Applying the same argument to base distribution p_d and perturbation $-d$ proves the reverse KL bound.

To bound TV, set $W = e^{d-\ell}$, $b = e^{\omega} > 1$, and $z = \mathbb{E}_p W \in [1, b]$. Convexity of $w \mapsto (w-z)_+$ gives $(w-z)_+ \le (w-1)(b-z)/(b-1)$ for $w \in [1, b]$. Consequently,

$$\begin{aligned}\mathrm{TV}(p, p_d) &= \frac{\mathbb{E}_p(W-z)_+}{z} \le \frac{(z-1)(b-z)}{z(b-1)} \\ &= \frac{b+1-z-b/z}{b-1} \le \frac{b+1-2\sqrt{b}}{b-1} = \frac{\sqrt{b}-1}{\sqrt{b}+1} = \tanh(\omega/4).\end{aligned}$$

Equality holds for $W \in \{1, b\}$ with $p(W = b) = 1/(1+\sqrt{b})$, which gives $z = \sqrt{b}$ and proves sharpness.

For $0 < a < 1$, the ratio of the set odds is

$$\frac{p_d(A)}{1-p_d(A)}\frac{1-a}{a} = \frac{\mathbb{E}_p[e^d \mid A]}{\mathbb{E}_p[e^d \mid A^c]} \in [e^{-\omega}, e^{\omega}].$$

Solving for $p_d(A)$ gives (18). For $a \in \{0, 1\}$, the bounds hold with equality because A is empty or all of $\mathcal{X}$. □

Theorem A.3 (Frozen-reference target approximation)**.** Fix $\delta \in (0, 1)$ and put

$$\epsilon_m(\delta) := \sqrt{\frac{\log(2/\delta)}{2m}}, \qquad c := \lambda \epsilon_m(\delta). \tag{19}$$

With probability at least $1 - \delta$ over Y, the following bounds hold simultaneously for every finite $\lambda \ge 0$:

$$\sup_x |\widehat{\Phi}(x) - \Phi(x)| \le \epsilon_m(\delta), \tag{20}$$

$$\sup_x |\log \widehat{R}(x) - \log R^*(x)| \le c, \tag{21}$$

$$\sup_x \left|\log \frac{\widehat{p}(x)}{p^*(x)}\right| \le 2c, \tag{22}$$

$$\max\{D_{\mathrm{KL}}(p^*\|\widehat{p}), D_{\mathrm{KL}}(\widehat{p}\|p^*)\} \le \min\{2c, c^2/2\}, \tag{23}$$

$$\mathrm{TV}(\widehat{p}, p^*) \le \tanh(c/2). \tag{24}$$

In particular, $|\widehat{p}(A) - p^*(A)| \le \tanh(c/2)$ for every $A \subseteq \mathcal{X}$. The more specific bounds (18) also hold with $a = p^*(A)$ and $\omega = 2c$.

Proof. Lemma A.1 gives an event of probability at least $1 - \delta$ on which (20) holds. On this event, the log-weight difference $d(x) = \lambda(\widehat{\Phi}(x) - \Phi(x))$ satisfies $|d(x)| \le c$, hence $\mathrm{osc}(d) \le 2c$. Since $\widehat{p}(x) = p^*(x)e^{d(x)}/\mathbb{E}_{p^*}e^d$, Lemma A.2 gives every target bound. The rank event depends on neither λ nor A, so the conclusions are simultaneous in both. □

For a target TV tolerance $\varepsilon \in (0, 1)$ and a prescribed λ, a sufficient reference size is

$$m \ge \frac{\lambda^2 \log(2/\delta)}{8\,\mathrm{arctanh}^2(\varepsilon)}, \qquad m \in \mathbb{N}, \quad m \ge 1. \tag{25}$$

This makes $c/2 \le \mathrm{arctanh}(\varepsilon)$ and gives a sufficient uniform bound of order λ^2 at fixed confidence and tolerance. The theorem also yields $\mathrm{TV}(\widehat{p}, p^*) \to 0$ in probability for fixed λ as $m \to \infty$. At $\lambda = 0$, both targets are uniform for every Y.

Pointwise precision and parameter selection. By Proposition 2, the standard deviation of the log reward at a fixed candidate is

$$\mathrm{sd}_Y\left[\lambda(\widehat{\Phi}(x)-1)\right]=\lambda\sqrt{\frac{\Phi(x)(1-\Phi(x))}{m}}. \tag{26}$$

For large λ, if $1-\Phi(x)$ is of order $1/\lambda$, this standard deviation is of order $\sqrt{\lambda/m}$, motivating the initial rule $m \gtrsim \lambda$ in Algorithm 1. The calculation concerns one log reward. The normalized target is controlled by (25) for arbitrary μ and tied scores, and by Proposition A.6 in mean square under uniform μ without ties.

For fixed m, μ, Theorem A.3 allows λ and the evaluated set to depend on Y. The theorem does not cover repeated reference selection or a data-dependent stopping size. If m, μ are chosen using pilot data and the final reference is drawn independently, the theorem applies conditionally on the pilot data.

A.3 Calibration and the mass of top sets

Suppose in this subsection that $\mu = \mathrm{Unif}(\mathcal{X})$ and R has no ties. Index objects from best to worst as $x_{(1)}, \dots, x_{(N)}$. For $q = L/N$ with $L \in \{0, \dots, N\}$, let $T_q = \{x_{(1)}, \dots, x_{(L)}\}$, with $T_0 = \varnothing$. Define

$$H_\lambda(q) := \begin{cases} \dfrac{1-e^{-\lambda q}}{1-e^{-\lambda}}, & \lambda > 0, \\ q, & \lambda = 0. \end{cases} \tag{27}$$

The finite geometric sum in Proposition 4 gives $p^*(T_q) = H_\lambda(q)$. Uniform weights give the continuous extension at $\lambda = 0$. Since $\Phi(x_{(j)}) = 1 - j/N$, the population log range is $\lambda(1 - 1/N)$.

Proposition A.4 (Calibration as a parameter choice)**.** For $0 < q < 1$ on the rank grid, $H_\lambda(q)$ is continuous and strictly increasing in $\lambda \geq 0$, with $H_0(q) = q$ and $\lim_{\lambda\to\infty} H_\lambda(q) = 1$. Hence each requested mass $\alpha \in [q, 1)$ determines a unique finite $\lambda \geq 0$.

Proof. Write $u(x_{(j)}) = j/N$. Differentiating the normalized weights $e^{-\lambda u(x)}$ gives

$$\begin{aligned} \frac{d}{d\lambda}p^*(T_q) &= -\operatorname{Cov}_{p^*}(\mathbb{1}_{T_q}, u) \\ &= p^*(T_q)\big(1-p^*(T_q)\big)\left(\mathbb{E}_{p^*}[u \mid T_q^c] - \mathbb{E}_{p^*}[u \mid T_q]\right) > 0. \end{aligned}$$

Both sets are nonempty, all weights are positive, and every u in T_q is smaller than every u in its complement. Continuity follows from the finite sums defining the target. The limit follows from (27). The intermediate value theorem and strict monotonicity give existence and uniqueness. □

For $u = 1 - \Phi$ and fixed λ, Riemann sums give, for every continuous g on $[0, 1]$,

$$\frac{\sum_{j=1}^N g(j/N)e^{-\lambda j/N}}{\sum_{j=1}^N e^{-\lambda j/N}} \xrightarrow[N\to\infty]{} \frac{\int_0^1 g(u)e^{-\lambda u}\,du}{\int_0^1 e^{-\lambda u}\,du}.$$

Thus the law of u under p^* converges weakly, for $\lambda > 0$, to the truncated exponential density

$$p(u) = \frac{\lambda e^{-\lambda u}}{1-e^{-\lambda}}, \qquad u \in [0, 1]. \tag{28}$$

At $\lambda = 0$ the limiting density is uniform. For $\lambda > 0$, the finite weights are those of exponential ranking with base $e^{-\lambda/N}$, up to a common factor and reversal of the rank index (Blickle & Thiele, 1996).

Corollary A.5 (Finite-reference calibration band)**.** On the event of Theorem A.3, the following bounds hold simultaneously for every $q \in \{0, 1/N, \dots, 1\}$ and finite $\lambda \geq 0$, with $a = H_\lambda(q)$ and $c = \lambda\epsilon_m(\delta)$:

$$|\widehat{p}(T_q) - a| \leq \tanh(c/2), \tag{29}$$

$$\frac{a}{a + (1-a)e^{2c}} \leq \widehat{p}(T_q) \leq \frac{ae^{2c}}{1 - a + ae^{2c}}. \tag{30}$$

Proof. Substitute $p^*(T_q) = H_\lambda(q)$ into the set bounds of Theorem A.3. □

Proposition A.6 (Mean-square error under a uniform reference measure)**.** Assume uniform μ and no score ties. For fixed $\lambda \geq 0$ and $m \geq 1$,

$$\mathbb{E}_Y D_{\mathrm{KL}}(p^* \| \widehat{p}) \leq \frac{\lambda}{2m}, \qquad \mathbb{E}_Y \, \mathrm{TV}(\widehat{p}, p^*)^2 \leq \frac{\lambda}{4m}. \tag{31}$$

In particular, for every $q \in \{0, 1/N, \ldots, 1\}$,

$$\left(\mathbb{E}_Y \left[\left(\widehat{p}(T_q) - H_\lambda(q) \right)^2 \right] \right)^{1/2} \leq \frac{1}{2} \sqrt{\frac{\lambda}{m}}. \tag{32}$$

Proof. The case $\lambda = 0$ is immediate. For $\lambda > 0$, index objects from best to worst and set

$$v_j = \Phi(x_{(1)}) - \Phi(x_{(j)}) = \frac{j-1}{N}, \qquad \widehat{v}_j = \widehat{\Phi}(x_{(1)}) - \widehat{\Phi}(x_{(j)}).$$

Then $m\widehat{v}_j$ counts reference draws in $\{x_{(2)}, \ldots, x_{(j)}\}$, empty when $j = 1$, so it is binomial with parameters m, v_j. Subtracting the best object's coordinate changes all weights by a common factor. Hence $p^*(x_{(j)}) \propto e^{-\lambda v_j}$ and $\widehat{p}(x_{(j)}) \propto e^{-\lambda \widehat{v}_j}$.

Put $b = m(1 - e^{-\lambda/m}) \in (0, \lambda]$ and $S(t) = \sum_{j=1}^N e^{-t v_j}$. The binomial moment formula gives

$$\mathbb{E}_Y e^{-\lambda \widehat{v}_j} = \left(1 - v_j + v_j e^{-\lambda/m}\right)^m \leq e^{-b v_j}.$$

The linear term $\lambda \sum_j p^*(x_{(j)})(\widehat{v}_j - v_j)$ in $D_{\mathrm{KL}}(p^* \| \widehat{p})$ has mean zero. Jensen's inequality therefore gives

$$\mathbb{E}_Y D_{\mathrm{KL}}(p^* \| \widehat{p}) = \mathbb{E}_Y \log \frac{\sum_j e^{-\lambda \widehat{v}_j}}{S(\lambda)} \leq \log \frac{S(b)}{S(\lambda)}.$$

For $t > 0$, $S(t) = (1 - e^{-t})/(1 - e^{-t/N})$. Since $1 - e^{-t}$ is increasing and concave and vanishes at zero,

$$\frac{S(b)}{S(\lambda)} = \frac{1 - e^{-b}}{1 - e^{-\lambda}} \frac{1 - e^{-\lambda/N}}{1 - e^{-b/N}} \leq \frac{\lambda}{b} = e^{\lambda/(2m)} \frac{\lambda/(2m)}{\sinh(\lambda/(2m))} \leq e^{\lambda/(2m)}.$$

Taking logarithms proves the KL bound. Pinsker's inequality $\mathrm{TV}(\widehat{p}, p^*)^2 \leq D_{\mathrm{KL}}(p^* \| \widehat{p})/2$ and $|\widehat{p}(T_q) - p^*(T_q)| \leq \mathrm{TV}(\widehat{p}, p^*)$ give the remaining bounds. □

For $\varepsilon, \delta \in (0, 1)$, $m \geq \lambda/(4\varepsilon^2)$ suffices for top-set RMSE at most ε under these assumptions. Markov's inequality also gives $\Pr_Y(\mathrm{TV}(\widehat{p}, p^*) > \varepsilon) \leq \lambda/(4m\varepsilon^2)$, so $m \geq \lambda/(4\delta\varepsilon^2)$ suffices at confidence $1 - \delta$.

A.4 Reference resolution and ties

The empirical coordinate takes at most $m + 1$ values. The top bin, where $\widehat{\Phi} = 1$, has the following distribution under a uniform reference measure without score ties.

Proposition A.7 (Top-bin resolution)**.** Assume a uniform reference measure and no score ties. Let

$$B_m := \{x : R(x) > \max_{1 \leq j \leq m} R(Y_j)\}. \tag{33}$$

For $L \in \{1, \ldots, N\}$ and $q = L/N$,

$$\Pr_Y(|B_m| \geq L) = \Pr_Y(Y_j \notin T_q \text{ for all } j) = (1 - q)^m, \tag{34}$$

$$\mathbb{E}_Y |B_m| = \sum_{k=0}^{N-1} (k/N)^m, \tag{35}$$

$$\frac{1}{m+1} - \frac{1}{N} \leq \mathbb{E}_Y \frac{|B_m|}{N} \leq \frac{1}{m+1}. \tag{36}$$

For any realized reference with $|B_m| \geq L$,

$$\widehat{p}(T_q) = \frac{L}{|B_m|} \widehat{p}(B_m) \leq \frac{L}{|B_m|}, \qquad \lim_{\lambda \to \infty} \widehat{p}(T_q) = \frac{L}{|B_m|}. \tag{37}$$

Proof. The event $|B_m| \geq L$ is equivalent to every draw missing T_q, giving (34) by independence and uniformity. The j-th best object belongs to B_m with probability $((N-j)/N)^m$, since all draws must be strictly worse. Summing over j gives (35). For the increasing function t^m,

$$\frac{1}{N}\sum_{k=0}^{N-1}(k/N)^m \leq \int_0^1 t^m\,dt = \frac{1}{m+1} \leq \frac{1}{N}\sum_{k=1}^{N}(k/N)^m.$$

The sums differ by $1/N$, proving (36). When $|B_m| \geq L$, we have $T_q \subseteq B_m$. Objects in B_m have weight one, whereas those outside have coordinate at most $1-1/m$ and weight at most $e^{-\lambda/m} \to 0$. This proves (37). □

For $0 < q < 1$, the exact integer condition for the reference to hit T_q with probability at least $1-\delta$ is

$$m \geq \left\lceil \frac{\log \delta}{\log(1-q)} \right\rceil. \tag{38}$$

The simpler $m \geq q^{-1}\log(1/\delta)$ suffices by $(1-q)^m \leq e^{-mq}$. This controls a reference hit, whereas (25) controls target error. The bound $1/(m+1)$ on expected relative bin size is not a deterministic feasibility threshold. The mass constraint (37) depends on the realized B_m.

Tie groups and nonuniform reference measures. Let $r_1 < \cdots < r_K$ be the distinct scores, with $C_k = \{x : R(x) = r_k\}$ and $n_k = |C_k|$. For any μ, define

$$a_k := \sum_{i<k}\mu(C_i), \qquad \widehat{a}_k := \frac{1}{m}\sum_{j=1}^{m}\mathbb{1}\{R(Y_j) < r_k\}. \tag{39}$$

Directly grouping the terminal sums yields

$$p^*(C_k) = \frac{n_k e^{\lambda a_k}}{\sum_i n_i e^{\lambda a_i}}, \qquad \widehat{p}(C_k) = \frac{n_k e^{\lambda \widehat{a}_k}}{\sum_i n_i e^{\lambda \widehat{a}_i}}. \tag{40}$$

For a set A selecting L_k objects from C_k, its mass is consequently

$$p^*(A) = \sum_k \frac{L_k}{n_k} p^*(C_k), \qquad \widehat{p}(A) = \sum_k \frac{L_k}{n_k}\widehat{p}(C_k). \tag{41}$$

Thus a fixed subset of L_k tied objects and fractional inclusion of L_k/n_k of the group have the same target mass.

Any $L \leq k$ objects from a highest-score group of size k receive at most L/k mass under either target. A requested mass $\alpha > L/k$ is therefore impossible. Under uniform μ, the population supremum is L/k. If $K > 1$, then a_K is uniquely largest, so dividing (40) by $e^{\lambda a_K}$ makes every lower-level contribution vanish as $\lambda \to \infty$. If $K = 1$, the target is uniform.

The coordinates a_k reflect reference masses, while target weights include multiplicities n_k. Nonuniform μ need not give the grid underlying $H_\lambda(q)$, even without ties. Theorem A.3 still applies. A sample maximum need not be a global maximum, so $\widehat{r}$ in Algorithm 1 diagnoses ties in the sampled region.

A.5 Correctness of the GFlowNet objectives

Consider the objectives in Appendix B on a finite acyclic graph $(\mathcal{S}, \mathcal{E}_c)$ with unique state s_0. Every state is reachable from s_0 and can reach a terminating state in $\mathcal{X} \subseteq \mathcal{S}$. The sink state s_f is separate from $\mathcal{S}$, and each terminating state has one terminating edge (x, s_f). Terminating states may also have other children. Thus

$$\sum_{v:(s,v)\in\mathcal{E}_c} P_F(v \mid s) + \mathbb{1}\{s \in \mathcal{X}\}P_F(s_f \mid s) = 1, \qquad s \in \mathcal{S}, \tag{42}$$

$$\sum_{u:(u,s)\in\mathcal{E}_c} P_B(u \mid s) = 1, \qquad s \in \mathcal{S} \setminus \{s_0\}. \tag{43}$$

The terminating term is included only at terminating states. The process terminates almost surely because the graph is finite and acyclic and every state with no non-terminating outgoing edge belongs to $\mathcal{X}$. Its terminating distribution is

$$P_F^\top(x) = \Pr_{P_F}(\text{the edge } (x, s_f) \text{ is taken}). \tag{44}$$

Proposition A.8 (Reward-proportional sampling at zero loss)**.** Fix λ and condition on the reference Y. For the chosen objective in Appendix B, suppose the flows and the policies are strictly positive on their respective domains and the policies have the above normalizations. Assume that each sampling distribution has full support on the domain specified there. If the expected loss is exactly zero, then

$$P_F^\top(x) = \widehat{p}(x) \quad \text{for every } x \in \mathcal{X}. \tag{45}$$

For any strictly positive terminal reward, all three objectives admit finite zero-loss solutions in an unrestricted positive function class.

Proof. Write $r(x) = \widehat{R}(x) > 0$. Full support and nonnegativity of the squared log residuals imply that zero expected loss forces every residual to vanish.

For FM, the resulting equalities are

$$I_f(s) = O_f(s) \quad (s \in \mathcal{S} \setminus \{s_0\}), \qquad f(x, s_f) = r(x) \quad (x \in \mathcal{X}). \tag{46}$$

Summing $O_f(s) - I_f(s)$ over $\mathcal{S}$ cancels the flows on non-terminating edges. Since $I_f(s_0) = 0$ and all other differences vanish,

$$O_f(s_0) = \sum_x f(x, s_f) = \sum_x r(x). \tag{47}$$

Let $H(s)$ be the visitation probability under normalized outgoing flows. We show $H(s) = O_f(s)/O_f(s_0)$ by topological induction. At s_0, both sides equal one. For $s \neq s_0$, acyclicity makes incoming transitions mutually exclusive, so the induction hypothesis gives

$$H(s) = \sum_{u:(u,s)\in\mathcal{E}_c} H(u)\frac{f(u,s)}{O_f(u)} = \frac{I_f(s)}{O_f(s_0)} = \frac{O_f(s)}{O_f(s_0)}.$$

Thus $P_F^\top(x) = H(x)f(x, s_f)/O_f(x) = r(x)/\sum_z r(z)$, also when the initial state itself is a terminating state.

For DB, zero residuals give

$$F(s)P_F(v \mid s) = F(v)P_B(s \mid v), \qquad F(x)P_F(s_f \mid x) = r(x). \tag{48}$$

Define non-terminating edge flows by $f(s, v) = F(s)P_F(v \mid s)$ and terminating flows by $f(x, s_f) = F(x)P_F(s_f \mid x)$. Forward normalization gives $O_f(s) = F(s)$ for every s. For each non-initial v, summing the first equality over its parents gives

$$I_f(v) = F(v) \sum_{s:(s,v)\in\mathcal{E}_c} P_B(s \mid v) = F(v).$$

These are the FM equalities, with the same normalized policy P_F, so (45) follows.

For TB, zero residuals imply that every complete trajectory terminating at x satisfies

$$ZP_F(\tau) = r(x)P_B(\tau^- \mid x). \tag{49}$$

Let $b(s)$ sum the backward path probabilities over all paths from s_0 to s in $(\mathcal{S}, \mathcal{E}_c)$, with $b(s_0) = 1$ for the empty path. Topological induction gives

$$b(s) = \sum_{u:(u,s)\in\mathcal{E}_c} P_B(u \mid s)b(u) = \sum_{u:(u,s)\in\mathcal{E}_c} P_B(u \mid s) = 1.$$

Summing (49) over paths to x gives $ZP_F^\top(x) = r(x)$, and then summing over objects gives $Z = \sum_x r(x)$ by almost sure termination.

To construct a zero-loss solution, fix a positive normalized backward policy. Set $r_0(s) = r(s)$ on $\mathcal{X}$ and zero elsewhere, and define in reverse topological order

$$F(s) = r_0(s) + \sum_{v:(s,v)\in\mathcal{E}_c} F(v)P_B(s \mid v). \tag{50}$$

Each $F(s)$ is finite and positive because s can reach a positive terminal reward. Set

$$P_F(v \mid s) = \frac{F(v)P_B(s \mid v)}{F(s)}, \qquad P_F(s_f \mid x) = \frac{r(x)}{F(x)}. \tag{51}$$

The recursion ensures forward normalization and DB, hence FM. Multiplying the DB equalities along a path, including its terminating edge, cancels intermediate flows and gives $F(s_0)P_F(\tau) = r(x)P_B(\tau^- \mid x)$, so TB holds with $Z = F(s_0)$. □

A.6 Learned terminating distribution

Corollary A.9 (Error decomposition for the terminating distribution)**.** For any learned forward policy on the above graph, define its discrepancy from the frozen target by

$$\eta_Y := \mathrm{TV}(P_F^\top, \widehat{p}). \tag{52}$$

On the reference event of Theorem A.3,

$$\mathrm{TV}(P_F^\top, p^*) \le \eta_Y + \tanh\left(\frac{\lambda\epsilon_m(\delta)}{2}\right). \tag{53}$$

Under a uniform reference measure and no score ties, every top set T_q defined in Appendix A.3 also satisfies

$$\left|P_F^\top(T_q) - H_\lambda(q)\right| \le \eta_Y + \tanh\left(\frac{\lambda\epsilon_m(\delta)}{2}\right). \tag{54}$$

The learned policy may depend on Y and on training randomness. Under the assumptions of Proposition A.8, exact zero loss gives $\eta_Y = 0$.

Proof. The triangle inequality gives $\mathrm{TV}(P_F^\top, p^*) \le \mathrm{TV}(P_F^\top, \widehat{p}) + \mathrm{TV}(\widehat{p}, p^*)$. Apply Theorem A.3. The variational characterization (12) and $p^*(T_q) = H_\lambda(q)$ give the top-set bound. Both steps hold for every realized policy on the reference event. Proposition A.8 gives the last statement. □

Under the assumptions of Proposition A.6, the triangle inequality and Minkowski's inequality also give

$$\left(\mathbb{E}\,\mathrm{TV}(P_F^\top, p^*)^2\right)^{1/2} \le \left(\mathbb{E}\eta_Y^2\right)^{1/2} + \frac{1}{2}\sqrt{\frac{\lambda}{m}},$$

where expectation includes Y and any training randomness. The fitting error η_Y is not inferred from a nonzero empirical loss. On an enumerable task, computing $p^*(T_q)$ and $\widehat{p}(T_q)$ separately distinguishes reference error from the policy's discrepancy on the same set.

B GFlowNet training objectives

The fixed reward $\widehat{R}$ can be used with flow matching (FM) (Bengio et al., 2021), trajectory balance (TB) (Whitammer et al., 2022), and detailed balance (DB) (Bengio et al., 2023). We use the graph and policy normalizations of Appendix A.5. Terminating states may have both non-terminating and terminating edges, and the sink is separate from $\mathcal{S}$. Expectations with empty sampling domains are omitted.

RC-FM introduces a positive edge flow f. Denote by

$$I_f(s) = \sum_{u\,:\,(u\to s)} f(u,s), \qquad O_f(s) = \sum_{v\,:\,(s\to v)} f(s,v) + \mathbb{1}\{s\in\mathcal{X}\}\, f(s,s_f) \tag{55}$$

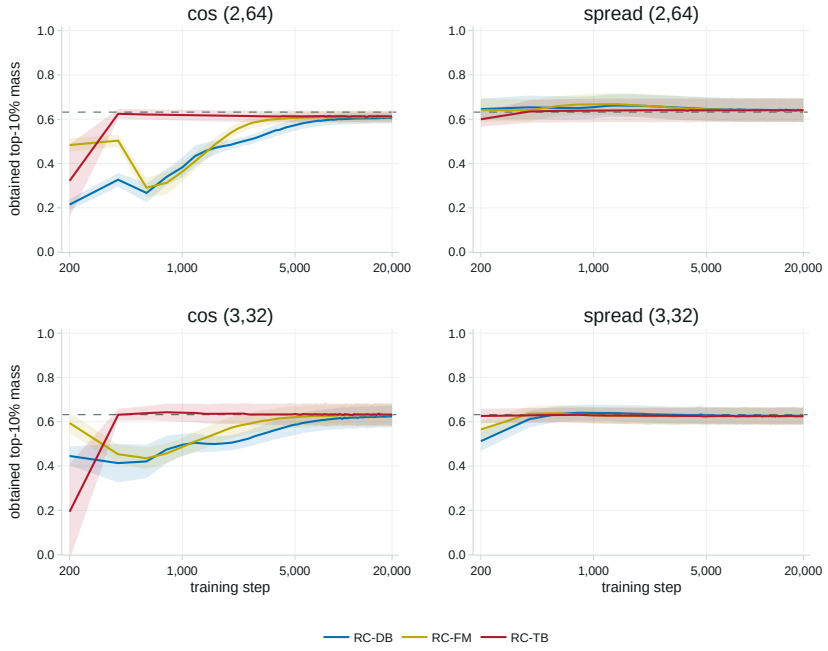

Figure 4: Obtained top 10% sampling mass of RC-GFN under the three training objectives as a function of the number of training steps, on the cosine and spread landscapes of sizes $(2, 64)$ and $(3, 32)$. The horizontal axis is on a logarithmic scale, and the dashed line marks the no-tie population benchmark of approximately 0.632 from Proposition 4.

the inflow and the outflow, where both sums range over non-terminating edges. The loss is

$$\mathcal{L}_{\text{RC-FM}} = \mathbb{E}_s \left[\log^2 \frac{I_f(s)}{O_f(s)} \right] + \mathbb{E}_x \left[\log^2 \frac{f(x, s_f)}{\widehat{R}(x)} \right], \tag{56}$$

where the expectations range over $\mathcal{S} \setminus \{s_0\}$ and $\mathcal{X}$, respectively. The forward policy is $P_F(s' \mid s) = f(s, s')/O_f(s)$, including terminating edges.

RC-TB introduces a positive scalar Z to estimate the partition function $\sum_x \widehat{R}(x)$. For a complete trajectory τ terminating in x_τ, let τ^- denote its non-terminating edges. The loss is

$$\mathcal{L}_{\text{RC-TB}} = \mathbb{E}_\tau \left[\log^2 \frac{Z\, P_F(\tau)}{\widehat{R}(x_\tau)\, P_B(\tau^- \mid x_\tau)} \right], \tag{57}$$

where

$$P_F(\tau) = P_F(s_f \mid x_\tau) \prod_{(s \to s') \in \tau^-} P_F(s' \mid s),$$
$$P_B(\tau^- \mid x_\tau) = \prod_{(s \to s') \in \tau^-} P_B(s \mid s').$$

Empty products equal one.

RC-DB introduces a positive state flow F, with the loss

$$\begin{aligned} \mathcal{L}_{\text{RC-DB}} = \mathbb{E}_{(s,s')} \left[\log^2 \frac{F(s)\, P_F(s' \mid s)}{F(s')\, P_B(s \mid s')} \right] \\ + \mathbb{E}_x \left[\log^2 \frac{F(x)\, P_F(s_f \mid x)}{\widehat{R}(x)} \right], \end{aligned} \tag{58}$$

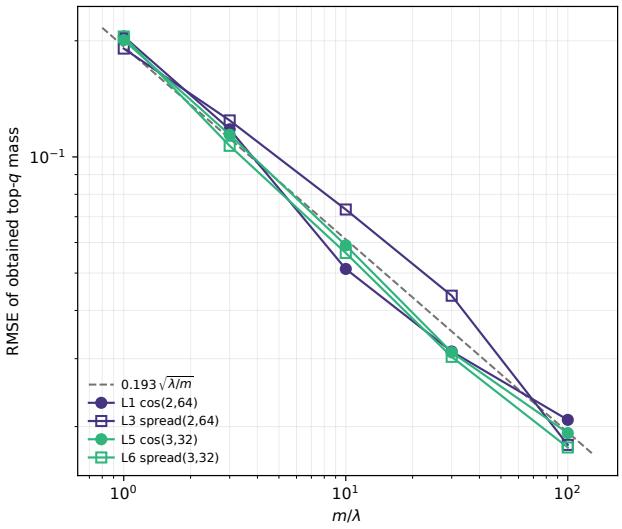

Figure 5: Root-mean-square deviation of the enumerated frozen target's top-10% mass from its declared value, over reference draws at $\lambda = 10$. Both axes are logarithmic, with m/λ on the horizontal axis. The dashed line is $0.193\,(m/\lambda)^{-1/2}$, using a fixed slope of $-1/2$ and the mean fitted coefficient across the four landscapes. The fitted exponents are -0.513, -0.502, -0.520 and -0.535, and the coefficients are 0.1903, 0.2112, 0.1897 and 0.1822.

where the first expectation is taken over non-terminating edges and the second over terminating states.

Under the positivity and full-support assumptions of Proposition A.8, any of these expected losses being zero implies $P_F^{\top} = \widehat{p}$.

B.1 Comparison of the three training objectives

Figure 4 compares the concentration obtained by RC-GFN with the three training objectives on the cosine and spread landscapes. All three objectives lead to top-set masses near the no-tie population benchmark of approximately 0.632 from Proposition 4 in these runs. On the more difficult cosine landscapes, RC-TB converges markedly faster than RC-FM and RC-DB, within about 400 to 500 steps, whereas RC-FM and RC-DB require about 5000 steps.

C Additional HyperGrid experiments

C.1 RQ3. How does top-set mass error depend on reference size?

We enumerate $\widehat{p}$ for repeated reference draws to measure top-set mass error without policy-fitting error. Figure 5 reports RMSE relative to the declared value at $\lambda = 10$ and $q = 10\%$. This statistic combines variation across draws with any systematic deviation from that value. The fitted slopes are close to $-1/2$ over the tested range, supporting $m^{-1/2}$ decay at this fixed λ. The curves are also close across the two tested grid sizes.

For a uniform reference measure without score ties, Proposition A.6 gives top-set RMSE relative to $p^*(T_q) = H_\lambda(q)$ at most $\frac{1}{2}\sqrt{\lambda/m}$. Thus $m \geq \lambda/(4\varepsilon^2)$ is a sufficient reference size for RMSE at most ε under these assumptions. This dependence on λ follows from the bound. Figure 5 varies only m, and its fitted coefficient describes the measured errors.

The cosine and spread formulas in Appendix H.1.1 have score ties. On such landscapes, RMSE about the declared no-tie value also includes the difference between that value and the actual population mass $p^*(T_q)$. Enumerating $p^*(T_q)$ separately isolates this difference from finite-reference error.

Algorithm 2 Heuristic screen at $\lambda = 1/q$

Input top quantile $q \in (0, 1]$, reference size $m \geq 1$, uniform sampler over the terminal space, reward R, number of reference redraws $J = 20$, sample size cap factor $C = 200$

Output diagnostic flag, mean $\bar{\pi}$ and standard deviation s_π of the estimated top-q mass of $\widehat{p}$

$\lambda \leftarrow 1/q$, $q_{\mathrm{ref}} \leftarrow 1/(m+1)$, $M \leftarrow \max\{\lfloor 4\lambda \rfloor, 64\}$
loop
  Draw M terminating states $x_1, \ldots, x_M$ uniformly and evaluate $R(x_i)$
  $c \leftarrow |\{i \mid R(x_i) = \max_{i'} R(x_{i'})\}|$
  Compute the diagnostic interval $[\mathrm{lo}, \mathrm{hi}]$ from c using equation 59
  if $\mathrm{lo}/M > q$ **then** set a tie flag and exit the loop
  if $\mathrm{hi}/M < q$ **then** set a reference flag if $q_{\mathrm{ref}} \geq q$, or no flag otherwise, and exit the loop
  if $M \geq \lfloor C\lambda \rfloor$ **then** mark undetermined and exit the loop
  $M \leftarrow 2M$
end loop
$M' \leftarrow \max\{M, 4m, \lfloor 20/q \rfloor\}$. Draw a separate pool $\mathcal{P}$ of M' uniform samples, retaining duplicates, and let $\mathcal{T}$ contain its $\lceil qM' \rceil$ highest-scoring entries
for $j = 1, \ldots, J$ **do**
  Draw m entries from $\mathcal{P}$ with replacement as $Y^{(j)}$, and compute $\widehat{\Phi}_j(x)$ for every entry $x \in \mathcal{P}$
  $\pi_j \leftarrow \sum_{x \in \mathcal{T}} e^{\lambda(\widehat{\Phi}_j(x) - 1)} \big/ \sum_{x \in \mathcal{P}} e^{\lambda(\widehat{\Phi}_j(x) - 1)}$
end for
Let $\bar{\pi}$ and s_π be the sample mean and standard deviation of $\pi_1, \ldots, \pi_J$
return the diagnostic flag, $\bar{\pi}$ and s_π

C.2 RQ4. Can reference and tie diagnostics identify concentration constraints before training?

The reference and tie diagnostics in Algorithm 1 can be evaluated before training. Under a uniform reference measure without score ties, $q_{\mathrm{ref}} = 1/(m+1)$ bounds the expected fraction of objects above every reference point (Proposition A.7). Comparing it with q indicates the scale of reference resolution. The actual mass ceiling depends on the realized top bin. In particular, $q \leq q_{\mathrm{ref}}$, or equivalently $m \leq \lambda - 1$ at $\lambda = 1/q$, is a screening rule rather than a feasibility threshold.

The tie diagnostic is c/M, where c counts occurrences of the highest score among M uniform samples. This measures ties at the sample maximum, which need not be the global maximum. If the global highest-score group has size k, a subset of $L \leq k$ of its objects receives at most L/k mass under either target (Appendix A.4). Whether this prevents a requested mass therefore depends on that mass as well as on q. The condition $q < k/N$ alone does not establish infeasibility.

Algorithm 2 evaluates both diagnostics at the example setting $\lambda = 1/q$ and estimates the frozen target's top-set mass using a separate uniform pool. Table 2 defines the four outcomes and the corresponding checks. The mass estimate is computed under every outcome. The five cases in Table 3 cover all four outcomes, with predicted masses close to the enumerated values.

The diagnostic interval uses the Byar approximation for a Poisson mean (Rothman & Boice, 1982),

$$\mathrm{lo} = c\left(1 - \frac{1}{9c} - \frac{z}{3\sqrt{c}}\right)^3, \qquad \mathrm{hi} = (c+1)\left(1 - \frac{1}{9(c+1)} + \frac{z}{3\sqrt{c+1}}\right)^3, \qquad z = 1.96, \tag{59}$$

with the lower endpoint truncated at zero. Here $c \geq 1$ because it counts the sample maximum. For a fixed top group of size k, its count in M iid uniform draws is $\mathrm{Bin}(M, k/N)$ and admits a Poisson approximation when k/N is small. The diagnostic instead selects a score level from the same sample and repeatedly doubles M. Thus these endpoints are used as heuristic cutoffs, not as a 95% confidence interval for the global top-group fraction.

The separate pool approximates the uniform measure on terminal objects. Conditional on this pool, $\bar{\pi}$ and s_π summarize variation across reference resamples from its empirical distribution. They do not account for uncertainty in the pool itself. The sums retain sample multiplicities. If a tie crosses the top-q boundary, any choice of the required number of tied entries gives the same target-weight sum.

Table 2: Heuristic diagnostic outcomes and suggested checks. The endpoints lo and hi are computed from equation 59.

Outcome	Condition	Interpretation	Suggested check
Tie flag	$\mathrm{lo}/M > q$	The diagnostic suggests a tied group larger than the requested fraction	Compare the requested mass with the tie-group ceiling
Undetermined	The interval straddles q at the sample cap	The diagnostic remains unresolved	Increase the diagnostic budget if needed
Reference flag	$\mathrm{hi}/M < q$ and $1/(m+1) \geq q$	The expected top-bin scale is at least q	Increase m and assess target error
No flag	$\mathrm{hi}/M < q$ and $1/(m+1) < q$	Neither heuristic screen is triggered	Examine the estimated top-set mass

Table 3: Diagnostic outcomes and estimated top-set masses before training, compared with enumerated values. Predicted values are $\bar{\pi} \pm s_\pi$ from Algorithm 2.

Case	Landscape	q	m	Assessment	Predicted top mass	Enumerated top mass
A	cosine $(3, 32)$	0.100	5	Reference flag	0.522 ± 0.181	0.522 ± 0.220
B	cosine $(3, 32)$	0.100	20	No flag	0.623 ± 0.103	0.651 ± 0.103
C	cosine $(3, 32)$	0.100	300	No flag	0.637 ± 0.044	0.632 ± 0.024
D	indicator $(2, 64)$	0.035	853	Undetermined	0.986 ± 0.006	0.986 ± 0.005
E	indicator $(2, 64)$	0.010	3000	Tie flag	0.289 ± 0.000	0.285 ± 0.000

C.3 RQ5. How does concentration evolve when the rank target is fixed before training?

We track diversity during training through the effective number of candidates n_{eff}. It is the exponential of the Shannon entropy of the empirical distribution $\tilde{p}$ of the $W = 4000$ terminating states sampled most recently,

$$n_{\mathrm{eff}} = \exp\Big(- \sum_{x \in \mathcal{X}} \tilde{p}(x) \log \tilde{p}(x) \Big). \tag{60}$$

It equals 1 when all W samples fall on the same terminating state and increases as the samples spread over more states. It lies between 1 and $\min\{W, N\}$ and measures diversity within this sample window. Using a moving window makes changes late in training visible without accumulating earlier samples.

In Figure 6, RC-GFN's top-set mass stabilizes near the benchmark fixed by λ and q. The OP-GFN and GR-GFN implementations used here reach top-set masses near one and much smaller windowed effective counts than RC-GFN. In particular, OP-GFN's effective count continues to fall after its top-set mass has nearly stabilized. A stable top-set mass can therefore conceal further concentration within that set. The OP-GFN behavior is consistent with the steepening of learned rank profiles described by Chen & Mauch (2024). GR-GFN permits several rank-derived fitting targets (Pastor-Pérez et al., 2025). RC-GFN fixes its target throughout training, so changes in its learned distribution reflect fitting a prescribed target.

D Error analysis of the terminating distribution

For a fixed iid reference design, Corollary A.9 gives, with probability at least $1 - \delta$ over the reference draw,

$$\mathrm{TV}(P_F^\top, p^*) \leq \mathrm{TV}(P_F^\top, \widehat{p}) + \mathrm{TV}(\widehat{p}, p^*) \leq \mathrm{TV}(P_F^\top, \widehat{p}) + \tanh\left(\lambda \sqrt{\frac{\log(2/\delta)}{8m}} \right). \tag{61}$$

The first term is policy-fitting error and the second is reference approximation error. The triangle inequality holds for every policy and reference draw. The probability statement concerns the bound on the reference term. This section computes the three total variations exactly on enumerable terminal spaces with $\delta = 0.05$. We take the trained RC-GFN models of the downstream experiments of

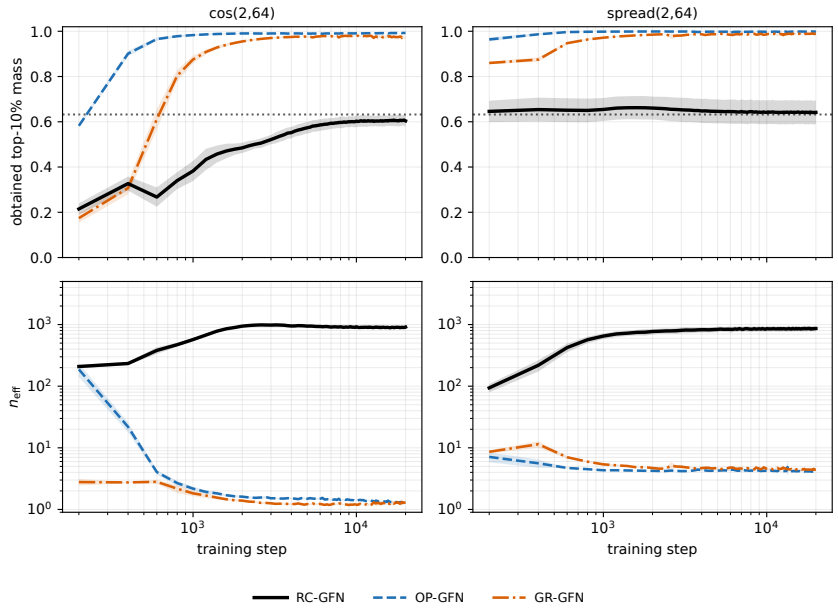

Figure 6: Concentration during training for RC-GFN, OP-GFN and GR-GFN on HyperGrid. The top row shows top-10% sampling mass, with the dotted line marking the no-tie population benchmark of approximately 0.632 from Proposition 4. The bottom row shows the windowed effective count n_{eff} on a logarithmic vertical axis.

Table 4: Total variation decomposition on the trained policies. Here $|u(\mathcal{X})|/|\mathcal{X}|$ is the number of distinct values of the objective function u on the terminal space divided by the number of terminating states, and smaller values indicate more ties. The bound is the one given by Theorem A.3 at $\delta = 0.05$. To separate the effect of score ties, we also report $\mathrm{TV}(P_F^{\top}, \widehat{p})$ merged by reward level, which first sums the mass that $P_F^{\top}$ and $\widehat{p}$ place on each reward value and then takes the total variation, so that it ignores differences of allocation within one reward level.

Task	λ	m	$\lvert u(\mathcal{X})\rvert/\lvert\mathcal{X}\rvert$	$\mathrm{TV}(P_F^{\top}, p^*)$	$\mathrm{TV}(P_F^{\top}, \widehat{p})$	Merged TV	$\mathrm{TV}(\widehat{p}, p^*)$	Bound
QM9	40	1,200	0.822	0.327 ± 0.006	0.331 ± 0.006	0.326 ± 0.006	0.045 ± 0.020	0.655
sEH	150	4,500	0.624	0.398 ± 0.012	0.399 ± 0.016	0.387 ± 0.018	0.037 ± 0.009	0.908
TFBind10	25	750	0.0108	0.702 ± 0.010	0.700 ± 0.013	0.441 ± 0.013	0.036 ± 0.011	0.551
RNA	400	12,000	1.46×10^{-6}	0.864 ± 0.004	0.865 ± 0.004	0.201 ± 0.010	0.055 ± 0.015	0.986

the main text, that is five random seeds on each of QM9, sEH, TFBind10 and RNA at the λ of the task and $m = 30\lambda$, and compute their terminating distributions $P_F^{\top}$ exactly. The prepend and append MDP of Appendices H.2.1 and H.3.1 can terminate only at length L, so $P_F^{\top}$ can be propagated forward layer by layer, the probability of a state of length $l+1$ being the sum over incoming actions of the parent probability times the action probability. If two actions produce the same child, both contributions are included. The transition probabilities come from the trained P_F, with the logits clipped to $[-20, 20]$ and renormalized as during sampling and with the exploration probability set to zero, which matches the setting of Table 1. The distribution $\widehat{p}$ is reconstructed from the reference set frozen for that seed, and p^* uses strict ranks over the whole space. Table 4 reports the three total variations and Table 5 the masses that the three distributions place on two top sets.

Table 4 shows that reference error is much smaller than policy-fitting error at these operating points. Across the four tasks, mean reference TV ranges from 0.036 to 0.055, whereas mean fitting TV ranges from 0.331 to 0.865. The learned distribution is therefore nearly as far from p^* as it is from $\widehat{p}$. At $m = 30\lambda$, improving the fit to the frozen target offers more room to reduce the error in the terminating distribution than increasing the reference size.

Table 5: Mass that the three distributions place on top sets. For each task the upper row uses $q = 1/\lambda$ and the lower row the q at which the task reports its top mass. The set T_q contains the $\lceil qN \rceil$ terminating states with the highest reward and includes the entire boundary tie group, so $|T_q|/N$ can exceed q. The population mass $p^*(T_q)$ does not vary across seeds. The value $H_\lambda(q)$ is the no-tie benchmark from Proposition 4, evaluated at the nominal q.

Task	q	$\lvert T_q\rvert/N$	$P_F^\top(T_q)$	$\widehat{p}(T_q)$	$p^*(T_q)$	$H_\lambda(q)$
QM9	0.025	0.025	0.610 ± 0.021	0.624 ± 0.043	0.632	0.632
	0.005	0.005	0.188 ± 0.017	0.183 ± 0.021	0.181	0.181
sEH	0.00667	0.00667	0.773 ± 0.022	0.622 ± 0.021	0.632	0.632
	0.001	0.001	0.267 ± 0.026	0.143 ± 0.008	0.139	0.139
TFBind10	0.04	0.047	0.292 ± 0.007	0.673 ± 0.022	0.689	0.632
	0.0055	0.0055	0.034 ± 0.001	0.124 ± 0.004	0.131	0.128
RNA	0.0025	0.00258	0.594 ± 0.010	0.638 ± 0.048	0.655	0.632
	0.00509	0.00526	0.764 ± 0.008	0.881 ± 0.022	0.885	0.869

The reference errors in the table are below the high-probability bounds, which are much larger than the measured errors. Since $\widehat{p}$ and p^* are constant within each score level, their TV can be computed exactly by grouping the weights in equation 40, without policy training. For 1,000 further independent reference draws per task, none exceeds the displayed bound. The mean reference TV lies between 0.046 and 0.047 across tasks, and its 95th percentile lies between 0.082 and 0.085. These measurements quantify the looseness of the general bound at the tested operating points. Appendix E varies m within each task.

Table 5 shows that fitting error can shift concentration in either direction. The learned top-set masses are close to $\widehat{p}(T_q)$ and $p^*(T_q)$ on QM9, exceed them on sEH, and fall below them on TFBind10 and RNA. The differences between $p^*(T_q)$ and $H_\lambda(q)$ on the latter two tasks arise at the population level, before reference sampling or training. The no-tie formula uses the nominal quantile, whereas these tasks have score ties and the evaluated sets include entire boundary tie groups.

Merging by score level separates two aspects of the comparison. For the groups C_k of Appendix A.4,

$$\frac{1}{2}\sum_k \left|P_F^\top(C_k) - \widehat{p}(C_k)\right| \le \frac{1}{2}\sum_k \sum_{x \in C_k} \left|P_F^\top(x) - \widehat{p}(x)\right| = \mathrm{TV}(P_F^\top, \widehat{p}).$$

The inequality follows by applying the triangle inequality within each group. The merged distance measures allocation across score levels, while the full distance also detects nonuniform allocation within them. Both targets are uniform within each level, but a learned policy need not be. Merging changes the measured fitting TV little on QM9 and sEH. On TFBind10 it reduces the mean from 0.700 to 0.441, leaving a substantial discrepancy across levels. On RNA the reduction is from 0.865 to 0.201, so score-level concentration is much closer to the target than object-level TV alone suggests. The tie-inclusive top sets in Table 5 depend only on level masses and are insensitive to allocation within a level.

E Choosing the reference size

Algorithm 1 uses $m \gtrsim \lambda$ as an initial rule. Appendix D finds that at $m = 30\lambda$, reference error is much smaller than policy-fitting error on the four downstream tasks. To examine the reference budget, we fix λ and the total oracle budget and vary m. The remaining budget is used for training in batches of 32, with all other settings unchanged.

Table 6 reports both m/λ and mq, where q is the nominal top fraction used for evaluation. Under a uniform reference measure without score ties, the expected top-bin fraction is at most $1/(m+1)$ and a reference misses a top set of fraction q with probability $(1-q)^m$ (Proposition A.7). Thus mq describes the scale of tail coverage. It is not a deterministic threshold for resolving that tail. For the tie-inclusive sets used here, the exact miss probability is $(1 - |T_q|/N)^m$.

Table 6: Ablation over the reference size. QM9 uses $\lambda = 40$ and $q = 0.005$, sEH uses $\lambda = 150$ and $q = 0.001$, TFBind10 uses $\lambda = 25$ and $q = 0.0055$, and RNA uses $\lambda = 400$ and $q = 0.00509$. The top q sampling mass is the mass that the terminating distribution of the trained policy places on T_q, with T_q as defined in Table 5. Reference share is the fraction of the combined reference and policy-training call budget spent on the reference.

Task	m	m/λ	mq	Reference share	Cumulative modes	Top q mass	$\mathrm{TV}(\widehat{p}, p^*)$
QM9	40	1	0.20	0.06%	308 ± 8	0.177 ± 0.044	0.204 ± 0.028
	120	3	0.60	0.19%	305 ± 9	0.195 ± 0.031	0.152 ± 0.044
	400	10	2.00	0.63%	306 ± 4	0.189 ± 0.021	0.083 ± 0.041
	1,200	30	6.00	1.88%	310 ± 4	0.188 ± 0.017	0.045 ± 0.020
sEH	150	1	0.15	0.23%	$1{,}036 \pm 146$	0.294 ± 0.160	0.256 ± 0.065
	450	3	0.45	0.70%	$1{,}120 \pm 23$	0.279 ± 0.037	0.140 ± 0.022
	1,500	10	1.50	2.34%	$1{,}093 \pm 28$	0.296 ± 0.037	0.057 ± 0.007
	4,500	30	4.50	7.03%	$1{,}082 \pm 31$	0.267 ± 0.026	0.037 ± 0.009
TFBind10	25	1	0.14	0.04%	$1{,}153 \pm 58$	0.036 ± 0.006	0.251 ± 0.102
	75	3	0.41	0.12%	$1{,}168 \pm 28$	0.033 ± 0.002	0.128 ± 0.026
	250	10	1.37	0.39%	$1{,}157 \pm 29$	0.032 ± 0.003	0.064 ± 0.026
	750	30	4.12	1.17%	$1{,}148 \pm 28$	0.034 ± 0.001	0.036 ± 0.011
RNA	40	0.1	0.20	0.03%	93 ± 16	0.564 ± 0.157	0.751 ± 0.094
	120	0.3	0.61	0.08%	123 ± 21	0.660 ± 0.085	0.551 ± 0.169
	400	1	2.04	0.25%	135 ± 17	0.717 ± 0.028	0.237 ± 0.061
	1,200	3	6.11	0.75%	122 ± 12	0.739 ± 0.029	0.147 ± 0.037
	12,000	30	61.08	7.50%	125 ± 16	0.777 ± 0.009	0.054 ± 0.015

Reference construction uses at most 7.5% of the combined reference and policy-training call budget in this ablation. The cumulative mode counts are similar across reference sizes on QM9 and TFBind10. On sEH the mean count remains similar but its uncertainty increases at the smallest reference size. On RNA the mean is lower at the smallest reference size. Top-set mass is also more variable with smaller references, especially on sEH and RNA, and its mean decreases on RNA. These comparisons measure the combined effect of changing m and reallocating the remaining budget to training. They indicate that the variability of attained concentration can be more sensitive to reference size than the number of discovered modes.

The measured reference TV decreases with m on all four tasks and is similar across tasks at matched m/λ. Under a uniform reference measure without ties, Proposition A.6 bounds its root mean square by $\frac{1}{2}\sqrt{\lambda/m}$. The downstream tasks have ties, so the cross-task agreement here is empirical. Theorem A.3 supplies the high-probability guarantee that also covers ties, with a sufficient reference size of order λ^2 at fixed tolerance and confidence. Together with the HyperGrid measurements in Figure 5, the ablation supports $m \gtrsim \lambda$ as a practical starting scale. With a fixed combined call budget, increasing m trades reference accuracy against the calls available for policy training.

F Sensitivity to λ

This section examines the sensitivity of RC-GFN to the concentration parameter λ on the four downstream tasks. Figure 7 shows the cumulative number of modes discovered during training for each value of λ, and Figure 8 reports the number of modes and the top-100 reward of 10,000 candidates sampled after training. On all four tasks, the value of λ that maximizes the cumulative number of discovered modes is no larger than the value that maximizes the final sampling quality. On QM9 and TFBind10, whose design spaces are smaller, a relatively small λ attains the best values of both metrics. On sEH and RNA, whose design spaces are larger, a larger λ is needed for both metrics to reach their best values.

G Calibration with tied scores

Theorem A.3 bounds finite-reference error for arbitrary μ and tied scores. Under uniform μ without ties, Proposition 4 gives the closed-form calibration, and Proposition A.6 gives the mean-square

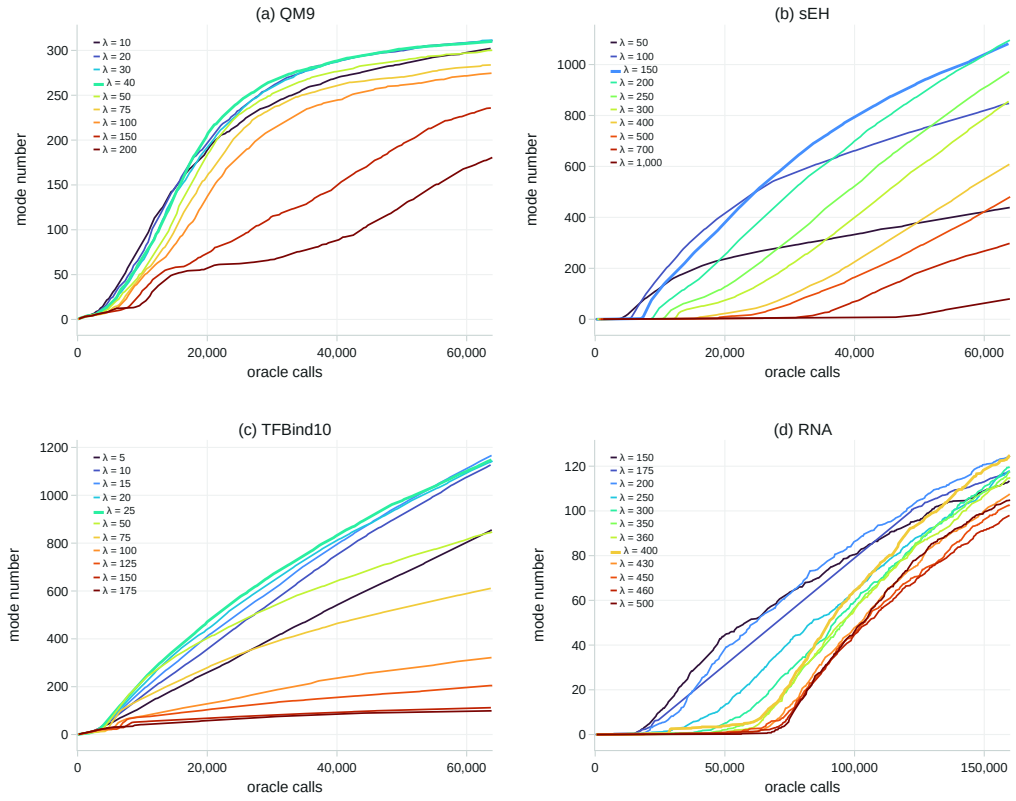

Figure 7: Cumulative number of discovered modes on the four downstream tasks for different values of λ. The thick curve marks the value used in the main text, namely $\lambda = 40$ on QM9, $\lambda = 150$ on sEH, $\lambda = 25$ on TFBind10 and $\lambda = 400$ on RNA. Each curve is the mean over five random seeds, and confidence intervals are omitted for clarity.

bound for fixed λ. Calibration with ties can be expressed through the score groups defined in Appendix A.4. For a top set $T = \bigcup_{k \geq k_0} C_k$, equation 40 gives

$$p^*(T) = \frac{\sum_{k \geq k_0} n_k e^{\lambda a_k}}{\sum_i n_i e^{\lambda a_i}}.$$

Under uniform μ, $a_{k+1} - a_k = n_k/N > 0$. If T is nonempty and proper, every coordinate in T therefore exceeds every coordinate outside it. The differentiation argument in Proposition A.4 then gives $dp^*(T)/d\lambda > 0$. The mass is continuous in λ, equals $|T|/N$ at $\lambda = 0$, and tends to one as $\lambda \to \infty$, when the highest-score group receives all the mass. Hence the attainable masses for finite $\lambda \geq 0$ are exactly $[|T|/N, 1)$, each attained at a unique λ. If the ordered score-group sizes are known, this value can be computed numerically. Without ties, the expression reduces to $H_\lambda(|T|/N)$. For nonuniform μ, the group formula remains valid, but its value depends on both the reference masses defining a_k and the object counts n_k, and cannot be determined from $\mu(T)$ alone.

Selecting only part of a score group imposes a different constraint. By equation 41, any L objects from the highest-score group of size n_K receive L/n_K of that group's mass under either target, and thus at most L/n_K of the total mass. The top sets in Table 5 instead include their boundary score group in full, so their actual fraction $|T_q|/N$ can exceed the nominal q. Ties also change the normalized weights across score levels, even without enlarging a boundary group. Both effects can contribute to the difference between $p^*(T_q)$ and $H_\lambda(q)$. Since these evaluated sets contain complete score groups, their masses depend only on the total probability assigned to each group. The score-level TV in Appendix D measures discrepancies in these group probabilities, while object-level TV also detects differences in allocation within groups.

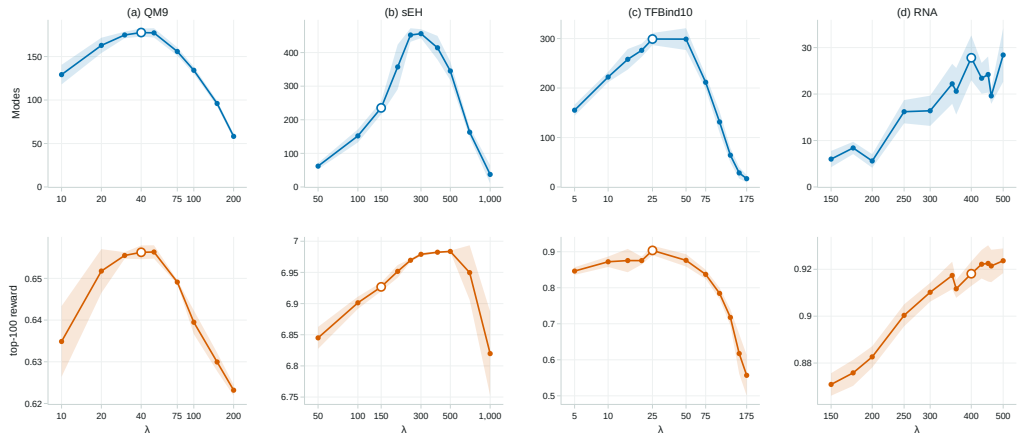

Figure 8: Final sampling quality on the four downstream tasks for different values of λ, evaluated on 10,000 candidates sampled after training. The top row shows the number of modes and the bottom row the top-100 reward, with λ on a logarithmic horizontal axis. Hollow markers indicate the values of λ used in the main text, namely $\lambda = 40$ on QM9, $\lambda = 150$ on sEH, $\lambda = 25$ on TFBind10 and $\lambda = 400$ on RNA.

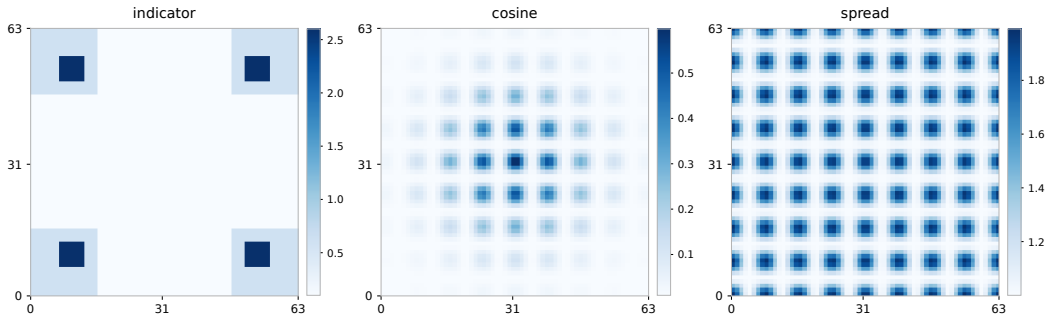

Figure 9: Illustration of the three HyperGrid reward landscapes, drawn at the size $(D, H) = (2, 64)$.

H Experimental configuration

H.1 HyperGrid

H.1.1 Reward landscapes

The three landscapes differ in their tie structure and in the distribution of their modes. The indicator landscape takes only three reward values, and its tied top candidates occupy about 3.5% of the whole space, so it exercises the tie diagnostic in RQ4. The cosine and spread landscapes also contain structural ties induced by reflections about the grid midpoint and permutations of coordinates. Unless an explicit tie-breaking convention is applied, they therefore fall under the tied-score analysis of Appendix A.4. The high reward states of the cosine landscape concentrate in a single basin at the center of the grid, whereas removing the Gaussian envelope spreads the peaks of the spread landscape over several separated basins, so the two landscapes together examine whether the correspondence between λ and the concentration is independent of the shape of the landscape (RQ1 to RQ3 and RQ5).

H.1.2 Implementation details

All methods on HyperGrid share the same network architecture, parameterization and optimizer, and differ only in the supervision applied at the terminating states. A state enters the network as a K-hot

Table 7: Parameters of the three HyperGrid reward landscapes. Here $a_d = |x_d/(H-1) - 0.5| \in [0, 0.5]$ is the distance from the normalized coordinate of dimension d to the midpoint of that dimension, and φ is the probability density function of the standard normal distribution. For cosine and spread, the two grid sizes and the two values of R_0 are listed in the same order. Figure 2(a) also includes cosine $(2, 64)$ with $R_0 = 0.01$ and spread $(2, 64)$ with $R_0 = 1$.

Landscape	Objective function	(D, H)	R_0	R_1	R_2
indicator	$R_0 + R_1 \prod_d \mathbb{1}\{a_d > 0.25\} + R_2 \prod_d \mathbb{1}\{0.3 < a_d < 0.4\}$	$(2, 64)$	0.1	0.5	2.0
cosine	$R_0 + R_1 \prod_d \big(\cos(50a_d) + 1\big)\, \varphi(5a_d)$	$(2, 64)$, $(3, 32)$	0.001, 0.01	1.0	N/A
spread	$R_0 + R_1 \prod_d \frac{1}{2}\big(\cos(50a_d) + 1\big)$	$(2, 64)$, $(3, 32)$	0.01, 1.0	1.0	N/A

vector, in which every coordinate is encoded as a one-hot vector of length H and the D vectors are concatenated, so that the input dimension is DH. The forward policy P_F, the backward policy P_B and the state flow F share one encoder, which is an MLP with 2 hidden layers of 256 units and ReLU activations. Each of the three has its own linear output layer on top of the encoder. P_F outputs $D + 1$ logits, one for advancing along each of the D dimensions and one for the terminating action, P_B outputs D logits, and $\log F$ outputs a scalar. The backward policy is trained and carries no KL regularization. All experiments in the main text and all five research questions use the detailed balance parameterization, and Appendix B.1 reports the flow matching and the trajectory balance parameterizations. The optimizer is Adam with a learning rate of 0.001 and no clipping of the gradient norm. Every training step samples 200 complete trajectories from the current forward policy, evaluates the loss on all transitions and terminating states of this batch, and performs one parameter update.

Write $u(x)$ for the objective function in Table 7, which is strictly positive on every landscape because $R_0 > 0$. GFN-β uses the standard detailed balance loss with the terminal reward in the power form of Section 1, $R(x) = u(x)^\beta$, so that its target distribution is $p(x) \propto u(x)^\beta$, with $\beta \in \{0.5, 1, 2, 4, 8\}$ in Figure 2(a). RC-GFN uses the RC-DB loss of Appendix B with coefficient 1 on both the detailed balance residual and the reward-matching term. The reference set is drawn before training as m states sampled independently with replacement from the uniform distribution over the terminal space, and stays frozen for the whole of training. The random seed determines both the initialization of the networks and the draw of the reference set, so the five seeds of one configuration correspond to five independent reference draws (Section 5).

OP-GFN adds an order-preserving term to the same detailed balance residual. This term acts on the log terminating flow $\log F(x) + \log P_F(s_f \mid x)$ with coefficient $\lambda_{\mathrm{OP}} = 0.1$. This is the value used by OP-GFN with OP-DB on HyperGrid and also the best performing value in its ablation over λ_{OP} (Chen & Mauch, 2024). The order-preserving term of OP-GFN is averaged over all pairs of terminating states within a batch.

H.2 Molecular design

H.2.1 Environments

Both molecular tasks use the prepend and append MDP, which reduces the generation of a molecular graph to the generation of a sequence of building blocks (Shen et al., 2023; Chen & Mauch, 2024; Kim et al., 2024b). Every building block carries two attachment sites. A state is a string of building blocks, a forward action adds one building block at the beginning or at the end of the string, the only forward action allowed once the string has length L is termination, and a backward action deletes the first or the last building block. The terminating states are therefore all block sequences of length exactly L, so that $|\mathcal{X}| = B^L$ with B the number of available building blocks, and the molecule of a terminating state is obtained by connecting its blocks in order.

QM9 ($|\mathcal{X}| = 161{,}051$, $|u(\mathcal{X})| = 132{,}304$). The objective function u is the prediction of the HOMO-LUMO gap by a proxy neural network. Every molecule consists of $L = 5$ building blocks drawn from $B = 11$ types, which gives $11^5 = 161{,}051$ terminating states. The reward is $R(x) = \max\{u(x), 10^{-3}\}$.

sEH ($|\mathcal{X}| = 34{,}012{,}224$, $|u(\mathcal{X})| = 21{,}228{,}996$). The objective function u is a proxy model that predicts the binding affinity to soluble epoxide hydrolase. This proxy is a gradient boosting regressor fit-

ted to the predictions of the graph neural network proxy of Bengio et al. (2021). Every molecule consists of $L = 6$ building blocks drawn from $B = 18$ types. The reward is $R(x) = \max\{u(x), 10^{-3}\}$.

H.2.2 IMPLEMENTATION DETAILS

All methods use the same environment and are evaluated on 10,000 post-training samples. Baseline method parameters follow the settings of their original work. The policy network uses the relative edge flow parameterization (SSR) of Shen et al. (2023), which scores a state together with each of its child states. A state is encoded position by position as one-hot vectors of building blocks that are concatenated, the positions beyond the current length are padded with zeros, and the input dimension is BL. The encoder is an MLP with 1 hidden layer of 1024 units that maps a state to a 1024 dimensional embedding. The scorer is likewise an MLP with 1 hidden layer of 1024 units, takes the concatenation of a state embedding and a child embedding as input, and outputs a scalar logit. The forward policy P_F is the softmax over the logits of all admissible child states, with the logits clipped to $[-20, 20]$. The backward policy P_B uses the same architecture, is normalized over the parent states, and shares no parameters with P_F. The state flow $\log F$ is given by another MLP with 1 hidden layer of 1024 units. All MLPs use LeakyReLU activations.

All GFlowNet baselines and RC-GFN use the detailed balance parameterization. GFN-β, Logit-GFN and Layer-GFN use the power form reward of Section 1 on both molecular tasks. GFN-β uses a fixed β, which is 5 on QM9 and 6 on sEH. These are the values used for the two tasks in the original work of Logit-GFN. Logit-GFN and Layer-GFN first encode the inverse temperature β into a 32 dimensional vector with an MLP with 2 hidden layers of 32 units. Logit-GFN then maps this vector to a scalar with another MLP of the same size and multiplies the logit of the scorer by it (Kim et al., 2024b), whereas Layer-GFN concatenates this vector to the input of the scorer (Zhang et al., 2023a). The state flow network of both takes β as an additional input. The values of β used for training and for exploration are drawn from a uniform distribution, $U[5, 15]$ on QM9 and $U[6, 30]$ on sEH. OP-GFN adds an order-preserving term to the detailed balance residual with coefficient $\lambda_{\mathrm{OP}} = 0.1$. The reward maximization baselines use the scaled $R(x)^\beta$ as the terminal return. Their policy network is the same as P_F, and A2C and PPO have in addition a value network with 1 hidden layer of 1024 units. The entropy regularization coefficient of A2C and PPO is 0.01, the clipping parameter of PPO is 0.2, and the temperature of SQL is 0.01.

Every training round first samples 32 online trajectories from the current policy, where each step is replaced with probability $\varepsilon = 0.10$ by a uniform choice among the admissible child states, and performs K parameter updates on this batch. It then forms an offline batch of 32 trajectories by the reward prioritized replay training (PRT) of Shen et al. (2023), drawing 16 terminating states from the top 10% and 16 from the bottom 90% of the rewards among all terminating states seen so far, sampling their trajectories backwards with P_B, and performing K further updates. Here $K = 1$ on QM9 and $K = 5$ on sEH. The four reward maximization baselines do not use PRT. PPO performs $2K$ updates on the online batch alone, A2C and SQL draw 32 samples uniformly from all past samples and perform $2K$ updates, and MARS draws 32 samples among those it accepts and performs K updates. The optimizer is Adam with a learning rate of 10^{-4} for all networks, and the gradient norm of every network is clipped at 10.

H.3 BIOLOGICAL SEQUENCE DESIGN

H.3.1 ENVIRONMENTS

Both biological sequence tasks use the same prepend and append MDP (Shen et al., 2023; Chen & Mauch, 2024; Kim et al., 2024b). A state is a string of nucleotides, a forward action adds one nucleotide at the beginning or at the end of the string, the only forward action allowed once the string has length L is termination, and a backward action deletes the first or the last nucleotide. The terminating states are therefore all sequences of length exactly L, so that $|\mathcal{X}| = 4^L$.

TFBind10 ($|\mathcal{X}| = 1{,}048{,}576$, $|u(\mathcal{X})| = 11{,}314$). The objective function u is the measured $-\Delta\Delta G$ of the binding between the yeast transcription factor Pho4 and DNA, and the measurement covers all combinations of the five nucleotides on each side of a fixed core sequence (Le et al., 2018). Every sequence has length $L = 10$. The reward is $R(x) = \max\{u(x), 10^{-3}\}$.

Table 8: Mode sets of the four downstream tasks.

Task	Candidate set	Similarity filter	Modes	Fraction of the terminal space	Source
QM9	Top 0.5%, 805 states	Tanimoto similarity below 0.5	364	0.226%	Logit-GFN (Kim et al., 2024b)
sEH	Top 0.1%, 34,012 states	Tanimoto similarity below 0.7	3,385	0.00995%	Logit-GFN (Kim et al., 2024b)
TFBind10	Top 0.5%, 5,764 states	None	5,764	0.550%	OP-GFN (Chen & Mauch, 2024)
RNA	Top 0.5%, 1,412,510 states	Local optima at Hamming distance 1	1,590	0.000592%	Kim et al. (2024c)

RNA ($|\mathcal{X}| = 268{,}435{,}456$, $|u(\mathcal{X})| = 392$). This task is L14_RNA1 of the FLEXS benchmark (Sinai et al., 2020). Every sequence has length $L = 14$ over the alphabet $\{A, U, C, G\}$. The objective function u is the binding energy between the sequence and a fixed target RNA of length 100, computed with the duplexfold routine of ViennaRNA (Lorenz et al., 2011) and divided by the binding energy between the target and its reverse complement scaled by the length ratio $14/100$. The reward is $R(x) = \max\{u(x), 10^{-3}\}$. The scaling constant is determined from a fixed sample of 200,000 uniformly random sequences that all methods share.

H.3.2 Implementation details

All methods use the same environment and evaluation budget. Baseline method parameters follow the settings of their original work. The policy network uses the same SSR parameterization as in Appendix H.2.2. A state is encoded position by position as one-hot vectors of nucleotides that are concatenated, the positions beyond the current length are padded with zeros, and the input dimension is $4L$. The encoder maps a state to an embedding, the scorer takes the concatenation of a state embedding and a child embedding as input and outputs a scalar logit, and the forward policy P_F is the softmax over the logits of all admissible child states, with the logits clipped to $[-20, 20]$. The backward policy P_B uses the same architecture, is normalized over the parent states, and shares no parameters with P_F. The network sizes differ between the two tasks. On TFBind10 the encoder has 1 hidden layer of 512 units and the embedding has 512 dimensions, the scorer has 2 hidden layers of 512 units, and the state flow network and the value network have 3 hidden layers of 512 units. On RNA every network has 1 hidden layer of 1024 units and the embedding has 1024 dimensions. All MLPs use LeakyReLU activations.

All GFlowNet baselines and RC-GFN use the detailed balance parameterization. GFN-β, Logit-GFN and Layer-GFN use the power form reward of Section 1 on both sequence tasks. GFN-β uses a fixed β, which is 3 on TFBind10 and 8 on RNA. These are the values used for the two tasks in the original works (Chen & Mauch, 2024; Kim et al., 2024b). Logit-GFN and Layer-GFN encode β and condition on it in the same way as in Appendix H.2.2, the state flow network of both takes β as an additional input, and the values of β used for training and for exploration are drawn from $U[3, 9]$ on TFBind10 and from $U[8, 24]$ on RNA. OP-GFN adds an order-preserving term to the same detailed balance residual with coefficient $\lambda_{\mathrm{OP}} = 0.1$. The reward maximization baselines use the scaled $R(x)^\beta$ as the terminal return. Their policy network is the same as P_F, and A2C and PPO have in addition a value network. The entropy regularization coefficient of A2C and PPO is 0.01, the clipping parameter of PPO is 0.2, and the temperature of SQL is 0.01.

Every training round first samples 32 online trajectories from the current policy, where each step is replaced with probability $\varepsilon = 0.01$ by a uniform choice among the admissible child states, and performs one parameter update on this batch. It then forms an offline batch of 32 trajectories by the PRT of Shen et al. (2023), drawing 16 terminating states from the top 10% and 16 from the bottom 90% of the rewards among all terminating states seen so far, sampling their trajectories backwards with P_B, and performing one further update. The four reward maximization baselines do not use PRT. PPO performs two updates on the online batch alone, A2C and SQL draw 32 samples uniformly from all past samples and perform two updates, and MARS draws 32 samples among those it accepts and performs one update. The optimizer is Adam with a learning rate of 10^{-4} for all networks, and the gradient norm of every network is clipped at 10.

H.4 Evaluation metrics

All four downstream tasks use the number of discovered modes as their main evaluation metric. The mode set of a task is a fixed set of terminal states determined before training starts. It is defined in two steps. The first step takes the terminal states with the highest values of the objective function u as candidates, and the second step removes candidates that are close to one another according to

a similarity criterion specific to the task, with TFBind10 as the exception. Table 8 summarizes the candidate set, the similarity filter and the number of modes of each task.

H.5 Computing infrastructure

All experiments were executed as Slurm jobs on a heterogeneous academic cluster. The QM9 and sEH experiments were single-GPU jobs, which ran on NVIDIA L40S (46 GB) and on H100 NVL (96 GB) cards. Each of these jobs was allocated 8 CPU cores and 24 GB of host memory. The HyperGrid, TFBind10 and RNA runs were CPU-only jobs, which were typically allocated 8 cores and 24 GB of host memory. The compute nodes run AlmaLinux 9.6 or 9.8, and the GPU nodes use NVIDIA driver 580.173.02. The HyperGrid experiments use Python 3.9.25, PyTorch 2.2.1 built against CUDA 12.1 with cuDNN 8.9.2, NumPy 1.24.4 and the torchgfn fork released with OP-GFN (Chen & Mauch, 2024). The molecular and biological sequence experiments use Python 3.9.25, PyTorch 2.2.1 built against CUDA 12.1 with cuDNN 8.9.2, NumPy 1.26.4, SciPy 1.13.1, RDKit 2025.09.2, scikit-learn 1.0.2, the RNA-binding landscape of FLEXS 0.2.8 (Sinai et al., 2020) and ViennaRNA 2.7.2 (Lorenz et al., 2011).