CAPACITY-AWARE SAMPLING FOR FROZEN GENERATIVE FLOW NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

GFlowNets are trained so that the terminal distribution is proportional to reward, yet scientific discovery deployments draw only a finite pool of samples from a frozen GFlowNet. We show that reward-proportional i.i.d. deployment is generally misaligned with finite-budget coverage and cannot adapt local flow to capacity as the sample pool grows. We first derive the finite-pool coverage target, which is generally not reward-proportional. To approximate this target, we propose Capacity-Aware Sampling (CAS), a local reweight-and-renormalize rule applied to the frozen policy. The penalty combines two deployment-time signals: a soft threshold on low-probability actions, and capacity-relative congestion of intermediate resources under the current sample history. CAS requires no retraining, no extra reward calls, and does not delete valid actions. Across five frozen GFlowNet benchmarks, a single default CAS setting improves finite-budget mode discovery over standard i.i.d. sampling on every task and achieves the best or near-best final-budget performance against tuned history-blind inference baselines. At $N = 2000$, CAS increases discovered modes from 51.8 to 252.8 on QM9STR and from 4.4 to 131.8 on SEHSTR.

1 INTRODUCTION

Many scientific discovery problems are not solved by returning a single best candidate. In molecular design, biological sequence optimization, and other black-box design tasks, a useful model should produce a pool of high-quality and diverse candidates for downstream screening. Generative Flow Networks (GFlowNets) are designed for multi-modal discovery: a trained forward policy generates objects so that the terminal marginal is proportional to reward (Bengio et al., 2021; 2023). Most work therefore improves the training process, including trajectory balance (Malkin et al., 2022), subtrajectory balance (Madan et al., 2023), replay and guided training (Shen et al., 2023), and local search (Kim et al., 2024). We study the later deployment question: once the GFlowNet is frozen and the user can draw only N samples, should inference remain i.i.d.?

Standard frozen deployment fails under finite budgets in two ways. First, reward-proportional i.i.d. sampling ignores diminishing returns in terminal coverage: once a region has already received substantial sampling budget, additional probability buys progressively fewer new discoveries. Second, it has no mechanism for matching realized trajectory flow to local capacity. A high-capacity prefix can productively support many trajectories, whereas the same amount of flow can congest a low-capacity state or transition. Yet i.i.d. deployment keeps sampling from the same frozen logits regardless of how the current pool has been allocated. CAS turns this static policy into a feedback-controlled allocator: it measures realized usage against frozen-policy capacity and redirects subsequent sampling away from locally saturated resources while preserving flow through high-capacity structure.

We propose to view a frozen GFlowNet as a local capacity model for finite-pool construction. The intuition comes from congestion and route-choice models in transportation science: travelers choose routes to minimize travel time and avoid congestion, where congestion is measured by how a link or a network is being used compared to its capacity, or volume-to-capacity ratio (Wardrop, 1952; Beckmann et al., 1956; Bureau of Public Roads, 1964). In the same spirit, the frozen forward policy supplies a reference flow, the current sample pool supplies the realized flow, and the ratio measures usage relative to capacity.

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.

During generation, CAS keeps the frozen forward policy as a reference and adjusts it locally via a per-action penalty. The penalty has two terms: a history-aware congestion term that detects overuse of transitions or child states relative to their frozen-policy capacity, and a soft threshold that downweights very low-probability actions without removing them. The resulting sampler requires no retraining, no additional reward evaluations, and no change to the valid action set.

Contributions. (i) We formulate frozen GFlowNet deployment as a finite-pool construction problem and show that the optimal fixed i.i.d. allocation for finite-pool coverage has a threshold form that generally differs from the reward-proportional training target. (ii) We propose CAS, a history-aware inference rule that combines a soft threshold on low-probability actions and the capacity-relative penalty of intermediate resources, with the frozen forward policy as the reference distribution. (iii) Across five frozen GFlowNet benchmarks, CAS improves mode discovery over standard i.i.d. sampling on every task without retraining or additional reward calls, while maintaining reward quality.

2 WHY FINITE-POOL DEPLOYMENT DIFFERS FROM I.I.D. SAMPLING

2.1 FROZEN GFLOWNET DEPLOYMENT

A GFlowNet defines a stochastic construction process on a DAG $\mathcal{G} = (\mathcal{S}, \mathcal{A})$, with source state s_0 and sink state s_f. A state $s \in \mathcal{S}^f := \mathrm{Par}(s_f)$ is called a terminating state, and a complete trajectory has the form $\tau = (s_0, s_1, \ldots, s_n, s_{n+1} = s_f)$ with $s_n \in \mathcal{S}^f$. A forward policy $P_F(\cdot \mid s)$ assigns a distribution over the children of each $s \neq s_f$. In a deterministic environment, this is equivalent to a distribution $P_F(a \mid s)$ over valid actions $a \in \mathcal{A}(s)$, where each action a leads to a unique child state $T(s, a) \in \mathrm{Child}(s)$. A frozen forward policy induces

$$p_0(\tau) = \prod_{t=0}^{n} P_F(s_{t+1} \mid s_t), \qquad p_0(x) = \sum_{\tau : x \to s_f \in \tau} p_0(\tau), \qquad x \in \mathcal{S}^f. \tag{1}$$

We call $p_0(x)$ the *terminal marginal*; it corresponds to the terminating state probability P_T of Bengio et al. (2023). The asymptotic training target is $p_0(x) \propto R(x)$ for a non-negative reward (Bengio et al., 2021; 2023). In this paper, P_F is fixed. Standard frozen inference draws $\tau_1, \ldots, \tau_N \overset{\text{i.i.d.}}{\sim} p_0$ and returns the terminal pool $P_N = \{x(\tau_1), \ldots, x(\tau_N)\}$, where $x(\tau)$ denotes the terminating state visited by τ. We focus on the resulting pool P_N rather than expectations under p_0.

2.2 FINITE-BUDGET COVERAGE AND LOCAL FLOW ALLOCATION

Under i.i.d. sampling from p_0, a finite budget N is spent in proportion to the frozen policy's mass. This creates a terminal-level coverage mismatch and leaves local flow blind to realized load relative to frozen-policy capacity.

(i) Terminal coverage loss. Let $X_1, \ldots, X_N \overset{\text{i.i.d.}}{\sim} q$ on a finite set $\mathcal{X}$, and let $U_N = |\{X_1, \ldots, X_N\}|$ denote the number of distinct modes covered by N i.i.d. sampling. Then

$$\mathbb{E}[U_N] = \sum_{x \in \mathcal{X}} \left[1 - (1 - q(x))^N\right]. \tag{2}$$

The expression is concave in q: extra mass on a region with high $q(x)$ contributes little additional coverage once $Nq(x)$ is large.

We now extend equation 2 to a weighted setting where each mode $m \in \mathcal{M}$ has importance $w_m > 0$.

Theorem 1 (Finite-pool coverage has a threshold optimum under i.i.d. sampling)**.** *Let q be a distribution over modes $m \in \mathcal{M}$ with weights $w_m > 0$, and let $F_N(q) = \sum_{m \in \mathcal{M}} w_m \left[1 - (1 - q_m)^N\right]$ denote the expected weighted number of distinct modes covered by N i.i.d. sampling. For $N > 1$, F_N is concave on the simplex. Its maximizer satisfies*

$$q_m^*(\alpha) = \left[1 - \left(\frac{\alpha}{w_m}\right)^{1/(N-1)}\right]_+, \qquad \sum_{m \in \mathcal{M}} q_m^*(\alpha) = 1, \tag{3}$$

for a threshold $\alpha \geq 0$. In particular, modes whose weight w_m falls at or below the cutoff α receive zero probability, and the finite-pool optimum is generally not proportional to w_m.

The proof is given in Appendix A.1. The exact optimizer q^* in Theorem 1 depends on the mode set $\mathcal{M}$ and weights w_m through the threshold α; realizing it directly in a frozen GFlowNet would additionally require control over the terminal marginal. The theorem identifies the finite-pool principle that motivates our local sampler: finite-pool discovery has diminishing returns, so after a region has received enough budget, additional probability has lower marginal coverage value.

(ii) Capacity-relative congestion. Shared intermediate resources have unequal capacity under the frozen policy. The same realized load can be well within capacity on a high-capacity prefix but congest a low-capacity state or transition. Under i.i.d. sampling, pairwise co-use accumulates quadratically with N; the following identity quantifies this reuse.

Proposition 1 (Pairwise resource reuse under i.i.d. sampling)**.** *Let $\tau_1, \dots, \tau_N \overset{\text{i.i.d.}}{\sim} p$ be trajectories sampled from a distribution p, and let $\mathcal{J}$ be a set of resources. For each resource $j \in \mathcal{J}$, let $B_{j,\tau} \in \{0,1\}$ indicate whether trajectory τ uses j, let $z_p(j) = \mathbb{P}_{\tau\sim p}(B_{j,\tau} = 1)$ be the probability that a single trajectory uses j, and let $C_N(j) = \sum_{i=1}^{N} B_{j,\tau_i}$ be the number of the N trajectories that use j. For nonnegative weights w_j, define the weighted pairwise reuse*

$$L_N = \sum_{j\in\mathcal{J}} w_j \binom{C_N(j)}{2}, \tag{4}$$

which counts the number of trajectory pairs that both use j, weighted by w_j. Then

$$\mathbb{E}[L_N] = \binom{N}{2} \sum_{j\in\mathcal{J}} w_j \, z_p(j)^2. \tag{5}$$

The expected pairwise reuse grows quadratically in N and is dominated by resources with the largest $z_p(j)$.

The quadratic growth of L_N creates increasing reuse pressure as the pool grows. Capacity determines where that pressure matters: high-capacity resources are expected to carry more flow, while the same load can congest a low-capacity resource. This motivates the capacity-normalized feedback used by CAS.

3 CAPACITY-AWARE SAMPLING

3.1 METHOD OVERVIEW

CAS turns a frozen GFlowNet into a history-aware finite-budget sampler without retraining. After each completed trajectory, it compares realized local usage with the flow expected under the frozen policy, combines this capacity-relative signal with a soft threshold on low-probability actions, and reweights the frozen logits for the next trajectory. Algorithm 1 summarizes the complete procedure.

Definition 1 (frozen-policy local capacity)**.** *In this paper, "capacity" means the amount of local sample flow that the frozen policy itself would assign to a resource under the current pool history. For a transition (s, a), this capacity is $C_{k-1}(s)P_F(a \mid s)$, where $C_{k-1}(s)$ is the number of previous trajectories that visited state s. For a child state, it is the aggregate frozen inflow from all parents.*

3.2 TERMINAL REALLOCATION INDUCES LOCAL KL UPDATES

Theorem 1 characterizes reallocation at the terminal level, while a frozen GFlowNet operates through local decisions along trajectories. We connect these levels by seeking a trajectory distribution that matches the terminal target while minimizing KL divergence to p_0.

Lemma 1 (Minimum-KL terminal targets induce local KL updates)**.** *Let $M : \mathcal{S}^f \to \mathcal{M}$ be a fixed terminal-mode map and write $M(\tau) = M(x(\tau))$. Let $p_0^{\mathcal{M}}(m)$ denote the probability that a trajectory drawn from p_0 has mode m. For any q with $\operatorname{supp}(q) \subseteq \operatorname{supp}(p_0^{\mathcal{M}})$, among all trajectory distributions whose terminal-mode marginal is q, the unique minimizer of $D_{\mathrm{KL}}(Q\|p_0)$ is*

$$Q_q^\dagger(\tau) = p_0(\tau)\frac{q_{M(\tau)}}{p_0^{\mathcal{M}}(M(\tau))}. \tag{6}$$

Let $r_q(m) = q_m/p_0^{\mathcal{M}}(m)$ and define the frozen-continuation value

$$V_q(s,a) = \mathbb{E}_{\tau\sim p_0}\big[r_q(M(\tau)) \mid \tau \text{ uses } (s,a)\big]. \tag{7}$$

For any visited non-sink state $s \neq s_f$ with $\sum_{b\in\mathcal{A}(s)} P_F(b \mid s)V_q(s,b) > 0$, the conditional action distribution under $Q_q^\dagger$ is

$$\pi_q^\dagger(a \mid s) = \frac{P_F(a \mid s)V_q(s,a)}{\sum_{b\in\mathcal{A}(s)} P_F(b \mid s)V_q(s,b)}. \tag{8}$$

Equivalently, $\pi_q^\dagger(\cdot \mid s)$ is the solution of

$$\arg\min_{\pi\in\Delta(\mathcal{A}(s))} \left[D_{\mathrm{KL}}\big(\pi \,\|\, P_F(\cdot \mid s)\big) + \sum_{a\in\mathcal{A}(s)} \kappa_q^\dagger(s,a)\pi(a) \right], \qquad \kappa_q^\dagger(s,a) = -\log V_q(s,a). \tag{9}$$

up to an additive constant in $\kappa_q^\dagger(s,\cdot)$.

The minimum-KL construction in equation 6 is an information projection onto a fixed terminal marginal (Csiszár, 1975), and its local form in equation 8 corresponds to a Doob h-transform (Doob, 1957) specialized to the frozen-GFlowNet DAG. In equation 9, the KL term comes from decomposing the trajectory-level KL into per-state contributions; the linear term is the local penalty needed to move toward a terminal target. For $q = q^*$ in Theorem 1, the exact penalty requires terminal ratios $q_m^*/p_0^{\mathcal{M}}(m)$ and their continuation values propagated backward through the DAG. CAS retains the KL-regularized form and constructs finite, observable penalties from two deployment-time signals: low-probability actions and capacity-relative congestion under the current history.

3.3 LOCAL CAPACITY-AWARE RESPONSE

Before drawing trajectory k, the sampler has history $h_{k-1} = (\tau_1, \ldots, \tau_{k-1})$. At a visited state s, each valid action $a \in \mathcal{A}(s)$ leads to a child state $u = T(s,a)$. CAS instantiates the variational form in equation 9 using a finite deployment diagnostic $\kappa_{k-1}(s,a)$ constructed from the frozen policy and sample history:

$$\tilde{\pi}_k(\cdot \mid s) = \arg\min_{\pi\in\Delta(\mathcal{A}(s))} \left[D_{\mathrm{KL}}\big(\pi \,\|\, P_F(\cdot \mid s)\big) + \sum_{a\in\mathcal{A}(s)} \kappa_{k-1}(s,a)\pi(a) \right]. \tag{10}$$

Proposition 2 (Local Gibbs response)**.** *Fix a nonterminal state s. Let $P_F(\cdot \mid s)$ be a forward policy on $\mathcal{A}(s)$, and let $\kappa(a) \in \mathbb{R} \cup \{+\infty\}$ for each action a, with $e^{-\infty} = 0$, and assume $\sum_{b\in\mathcal{A}(s)} P_F(b \mid s)e^{-\kappa(b)} > 0$. For the local objective in Eq. 10, the minimizer is given by*

$$\pi^*(a) = \frac{P_F(a \mid s)e^{-\kappa(a)}}{\sum_{b\in\mathcal{A}(s)} P_F(b \mid s)e^{-\kappa(b)}}. \tag{11}$$

In particular, $\pi^*(a) = 0$ whenever $P_F(a \mid s) = 0$, and $\pi^* = P_F$ whenever κ is finite and constant.

Thus CAS subtracts $\kappa_{k-1}(s,a)$ from the frozen log-probability $\log P_F(a \mid s)$ and renormalizes. Valid actions with nonzero frozen probability retain support and are reweighted by finite factors.

Remark 1 (History-dependent pool generation)**.** *For a fixed history h_{k-1}, the conditional rule $\tilde{\pi}_k(\cdot \mid s)$ is Markovian along the next trajectory. The joint law of the whole pool $(\tau_1, \ldots, \tau_N)$ is history-dependent because the sampling rule adapts after each completed trajectory. Accordingly,* CAS *operates as an inference-only finite-pool sampler with history-dependent deployment dynamics. The penalty κ_{k-1} is computed from the completed history before trajectory k starts, and the count tables are updated only after that trajectory terminates.*

3.4 CAPACITY-RELATIVE CONGESTION AND SOFT THRESHOLDING

The penalty has two terms. The first term is capacity-relative congestion. For a transition (s,a), let $C_{k-1}(s)$ be the number of previous trajectories that visited s, and let $C_{k-1}(s,a)$ be the number that took action a at s. The expected transition capacity and volume-to-capacity ratio are

$$\bar{C}^{\mathrm{trans}}_{k-1}(s,a) = C_{k-1}(s)P_F(a \mid s), \qquad \rho^{\mathrm{trans}}_{k-1}(s,a) = \frac{C_{k-1}(s,a)}{\max\{1, \bar{C}^{\mathrm{trans}}_{k-1}(s,a)\}}. \tag{12}$$

The denominator converts raw load into congestion: high-capacity actions can carry more traffic before being penalized, while low-capacity actions reach the same congestion level after fewer uses. The $\max\{1, \cdot\}$ floor avoids unstable ratios at the beginning of sampling.

For state-level feedback, let $u = T(s,a)$ be the child state. A child reachable from several parents should aggregate the expected inflow from all of them:

$$\bar{C}^{\mathrm{state}}_{k-1}(u) = \sum_{(v,b):T(v,b)=u} C_{k-1}(v)P_F(b \mid v), \qquad \rho^{\mathrm{state}}_{k-1}(s,a) = \frac{C_{k-1}(u)}{\max\{1, \bar{C}^{\mathrm{state}}_{k-1}(u)\}}. \tag{13}$$

This aggregate denominator avoids comparing the total visit count of u against the expected contribution of a single parent.

The second term is a soft threshold on low-probability actions, implemented through the inverse-probability ratio $p_{\min}/P_F(a \mid s)$.

For either ratio above, define

$$\widetilde{\rho}_{k-1}(s,a) = \max\left\{\rho_{k-1}(s,a),\ \frac{p_{\min}}{P_F(a \mid s)}\right\}, \tag{14}$$

for actions with $P_F(a \mid s) > 0$. The first term measures observed overuse; the second is a soft threshold on low-probability actions. Because both quantities are expressed as demand relative to frozen-policy capacity, the maximum combines comparable signals. The softness of the penalty avoids permanently removing low-probability actions that may still lead to useful continuations.

We convert the effective ratio into a BPR-style penalty (Bureau of Public Roads, 1964):

$$\kappa^{\mathrm{cap}}_{k-1}(s,a) = \lambda\left[\left(\frac{\widetilde{\rho}_{k-1}(s,a)}{t}\right)^{\beta} - 1\right]_+, \tag{15}$$

where $[z]_+ = \max\{z, 0\}$, $\lambda \geq 0$ controls the penalty strength, $t > 0$ is the activation threshold, and $\beta > 0$ controls the growth rate above the threshold. This BPR-style form gives a thresholded capacity response: the penalty is zero below t and grows with the effective ratio above it. Algorithm 1 gives pseudocode for the full sequential sampler.

Proposition 3 (Interior reallocation can beat i.i.d. and hard truncation)**.** *Consider a two-region finite-pool coverage model. A fraction α of draws goes to region H, which contains K_H useful modes; each draw in H is useful with probability θ_H. The remaining fraction $1-\alpha$ goes to region T, with K_T useful modes and per-draw usefulness θ_T. Applying equation 2 to each region, the expected discovered modes are*

$$F_N(\alpha) = K_H\left[1 - \left(1 - \frac{\alpha\theta_H}{K_H}\right)^N\right] + K_T\left[1 - \left(1 - \frac{(1-\alpha)\theta_T}{K_T}\right)^N\right]. \tag{16}$$

Standard i.i.d. sampling corresponds to $\alpha = p_H$, while hard truncation of T corresponds to $\alpha = 1$. If $F'_N(p_H) > 0$ and $F'_N(1) < 0$, then an interior head share $\alpha^* \in (p_H, 1)$ strictly improves over both.

The two conditions $F'_N(p_H) > 0$ and $F'_N(1) < 0$ typically hold when $\theta_H > \theta_T$ but $\theta_T > 0$: the head region is more useful per draw, yet the tail still contains modes worth covering. Both i.i.d. sampling and hard truncation then leave coverage on the table. The result identifies the finite-budget regime targeted by soft thresholding: probability should be shifted toward the head without eliminating the tail.

Algorithm 1 Capacity-Aware Sampling for a frozen GFlowNet

Require: frozen forward policy P_F, budget N, parameters (λ, β, t), feedback type $\text{type} \in \{\text{state}, \text{transition}\}$, soft threshold $p_{\min} \in [0, 1]$.

Initialize state counts $C_0(s) \leftarrow 0$ and transition counts $C_0(s, a) \leftarrow 0$.
for $k = 1, \ldots, N$ **do**
  $s \leftarrow s_0$; initialize trajectory $\tau_k \leftarrow (s_0)$.
  while $s \notin \mathcal{S}^f$ **do**
    for each valid action $a \in \mathcal{A}(s)$, with $P_F(a|s) > 0$ **do**
      $u \leftarrow T(s, a)$.
      if $\text{type} = \text{transition}$ **then**
        Compute $\rho_{k-1}(s, a)$ using eq. (12). (Trans-CAS)
      else
        Compute $\rho_{k-1}(s, a)$ using eq. (13). (State-CAS)
      end if
      $\widetilde{\rho}_{k-1}(s, a) \leftarrow \max\{\rho_{k-1}(s, a),\, p_{\min}/P_F(a \mid s)\}$.
      $\kappa(a) \leftarrow \lambda[(\widetilde{\rho}_{k-1}(s, a)/t)^{\beta} - 1]_+$.
      $\tilde{\ell}(a \mid s) \leftarrow \log P_F(a \mid s) - \kappa(a)$.
    end for
    Sample $a \sim \text{softmax}(\tilde{\ell}(\cdot \mid s))$.
    Append $(s, a, T(s, a))$ to τ_k.
    $s \leftarrow T(s, a)$.
  end while
  Update all state and transition counts using τ_k.
end for
return $\{\tau_1, \ldots, \tau_N\}$.

4 EXPERIMENTS

Protocol. We evaluate frozen GFlowNet checkpoints on BAG, TFBIND8, TFBIND10, QM9STR, and SEHSTR. Task names follow Shen's benchmark convention: TFBIND8 corresponds to SIX6, TFBIND10 corresponds to PHO4, QM9STR and SEHSTR are the string-MDP versions of QM9 and sEH. For the main comparison, all five checkpoints follow the released implementation of Shen et al. (2023) and its task-specific reward transformations, construction MDPs, and mode predicates. All samplers use the same checkpoint, valid-action mask, reward function, sample budget, and matched seed set within each task. No method updates model parameters or calls the reward model during generation. We report mean ± standard error (SE) over five matched seeds, following Shen's convention. The main matched-budget comparison uses $N = 2000$. The per-task reward exponent, construction MDP, and mode definition are defined in Appendix B.1.

For the history-blind baselines, we use tuned global settings selected from the sensitivity sweep in Appendix B.4: temperature sampling with $T = 0.5$, Top-k sampling with $k = 3$, nucleus sampling with $p = 0.8$, and ϵ-greedy with $\epsilon = 0.3$. Each baseline uses one setting across all tasks, rather than a separate per-task optimum.

Table 1: Matched-budget comparison at $N = 2000$ over five matched seeds (42–46). Entries are mean ± sample SE, computed as $s/\sqrt{5}$ with sample standard deviation s using $\text{ddof} = 1$. Rows are grouped by metric: mode discovery and mean top-10 reward.

Metric	Task	Standard	Temp. $T = 0.5$	Top-k $k = 3$	Nucleus $p = 0.8$	ϵ-greedy	State-CAS	Trans-CAS
Mode discovery	BAG	18.0 ± 3.2	36.0 ± 2.1	44.0 ± 0.7	31.8 ± 1.3	14.4 ± 2.0	45.0 ± 3.4	42.4 ± 2.2
	TFBIND8	47.8 ± 2.1	81.6 ± 2.3	13.6 ± 0.7	81.8 ± 1.6	32.4 ± 2.4	77.2 ± 2.6	81.0 ± 1.7
	TFBIND10	13.4 ± 1.6	17.8 ± 0.9	17.0 ± 1.0	13.0 ± 1.4	12.4 ± 2.1	19.8 ± 2.5	23.0 ± 1.7
	QM9STR	51.8 ± 2.8	127.6 ± 3.7	55.6 ± 0.4	111.0 ± 3.8	41.2 ± 2.7	247.2 ± 3.0	252.8 ± 7.6
	SEHSTR	4.4 ± 1.6	11.2 ± 1.2	119.4 ± 0.4	10.6 ± 1.5	2.8 ± 0.7	131.8 ± 4.5	126.6 ± 6.5
Mean top-10 reward	BAG	29.6 ± 0.40	30.0 ± 0.00	30.0 ± 0.00	30.0 ± 0.00	30.0 ± 0.00	30.0 ± 0.00	30.0 ± 0.00
	TFBIND8	9.6 ± 0.04	9.8 ± 0.02	9.3 ± 0.01	9.8 ± 0.01	9.6 ± 0.04	9.8 ± 0.02	9.8 ± 0.01
	TFBIND10	7.7 ± 0.13	7.8 ± 0.13	7.6 ± 0.06	7.4 ± 0.11	7.7 ± 0.14	7.9 ± 0.11	7.9 ± 0.10
	QM9STR	20.5 ± 2.65	100.0 ± 0.00	100.0 ± 0.00	42.0 ± 4.47	18.1 ± 2.27	100.0 ± 0.00	100.0 ± 0.00
	SEHSTR	7.1 ± 0.16	7.4 ± 0.03	9.0 ± 0.00	7.4 ± 0.05	6.9 ± 0.04	8.4 ± 0.06	8.5 ± 0.06

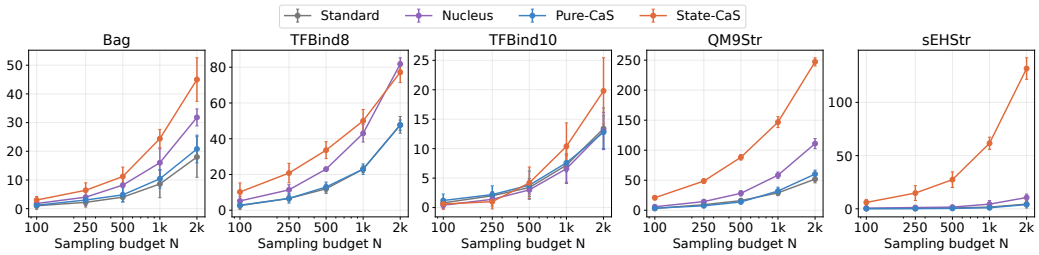


Figure 1: Mode discovery vs. budget for standard i.i.d., fixed nucleus ($p = 0.8$), PURE-CAS, and State-CAS. State-CAS yields substantial gains on BAG, QM9STR, and SEHSTR, remains competitive with nucleus on TFBIND8 and TFBIND10.

For CAS, we use one default setting shared across tasks and feedback types, $(\lambda, \beta, t, p_{\min}) = (1, 1, 1, 0.2)$, unless otherwise stated. Appendix B.5 shows that tuning these parameters can further change performance, so the main table is a conservative default-setting comparison rather than a per-task tuned CAS result.

Mode discovery at matched budget. A mode is a distinct terminal object satisfying the task-specific predicate (Appendix B.1). Table 1 gives the endpoint comparison. The best capacity-aware variant improves mode discovery over i.i.d. sampling on every benchmark while preserving the support of the frozen policy. Different components expose two regimes. On BAG, the soft threshold captures most of the gain, while history feedback adds little. On TFBIND8, QM9STR, and SEHSTR, history feedback provides substantial gains beyond the soft threshold, showing that adaptive local reallocation matters beyond static filtering. TFBIND10 shows a smaller but still positive endpoint gain. Appendix B.6 gives the matched-seed uncertainty for the headline gains. The gain remains after excluding modes already present in the training archive and across exploration-oriented GFlowNet training methods (Appendices B.2 and B.3).

Why static decoding falls short. Temperature sampling and ϵ-greedy change local entropy but expose no feedback loop from realized congestion to the next draw; ϵ-greedy in fact underperforms standard sampling. Nucleus sampling is a strong history-blind baseline on tasks where the useful modes lie inside its kept set (TFBIND8, QM9STR), but it applies the same truncation to every draw and barely improves over standard sampling on SEHSTR (10.6 vs. 4.4). Top-k is the most uneven baseline: it is the strongest non-CAS method on SEHSTR (119.4) but collapses on TFBIND8 (13.6, well below the standard 47.8), depending on whether the useful modes lie in the local head of P_F. Across the five tasks, no single fixed transformation of the frozen logits is robust, while the history-aware variants come close to the best per-task baseline on every task.

Top-10 reward quality is largely preserved. The top-10 reward rows of Table 1 show that CAS never falls below standard sampling and ties or near-ties the best baseline on four of the five tasks; the only visible gap is on SEHSTR, where Top-k reaches 9.0 versus 8.4–8.5 for CAS. The capacity-aware variants therefore expand the candidate pool without sacrificing the high-reward objects.

Table 2: Component ablation at $N = 2000$ (mode discovery, mean $\pm$ sample SE over five matched seeds, 42–46). SE is computed as $s/\sqrt{5}$ with sample standard deviation s using ddof $= 1$. The table separates hard truncation, soft threshold, state-level congestion, and transition-level congestion. PURE-CAS uses state feedback with $p_{\min} = 0$; RAWCOUNT-CAS uses raw state-visit counts.

Method	Threshold	State cong.	Trans. cong.	BAG	TFBIND8	TFBIND10	QM9STR	SEHSTR
Standard	✗	✗	✗	18.0 ± 3.2	47.8 ± 2.1	13.4 ± 1.6	51.8 ± 2.8	4.4 ± 1.6
Nucleus $p = 0.8$	Hard	✗	✗	31.8 ± 1.3	81.8 ± 1.6	13.0 ± 1.4	111.0 ± 3.8	10.6 ± 1.5
RAWCOUNT-CAS	✗	✓	✗	25.2 ± 3.9	35.0 ± 0.9	13.0 ± 1.1	38.8 ± 3.6	3.6 ± 0.7
Soft Threshold	Soft	✗	✗	50.8 ± 3.5	69.0 ± 2.5	20.6 ± 2.7	184.0 ± 3.0	107.4 ± 4.9
PURE-CAS	✗	✓	✗	20.8 ± 2.2	47.6 ± 1.3	12.8 ± 1.2	60.2 ± 3.0	4.0 ± 0.4
State-CAS	Soft	✓	✗	45.0 ± 3.4	77.2 ± 2.6	19.8 ± 2.5	247.2 ± 3.0	131.8 ± 4.5
Trans.-CAS	Soft	✗	✓	42.4 ± 2.2	81.0 ± 1.7	23.0 ± 1.7	252.8 ± 7.6	126.6 ± 6.5

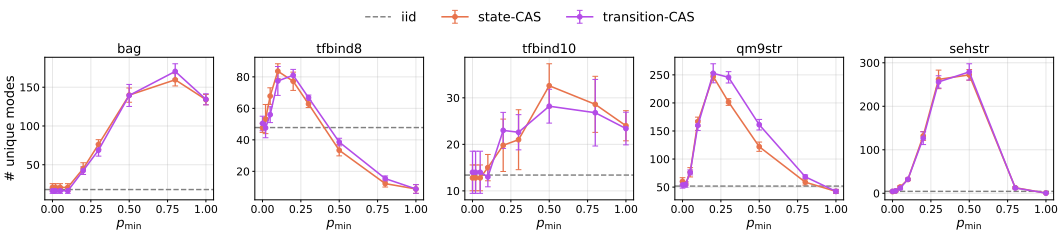


Figure 2: Sensitivity to the soft threshold $p_{\min}$ at $N = 2000$. The x-axis shows different values of $p_{\min}$, and the y-axis reports the number of unique modes discovered.

Which term matters? Table 2 separates the contribution of each CAS component. On BAG, soft threshold alone outperforms the history-aware variants. BAG (Shen et al., 2023) has unordered multiset states, which causes the capacity ratio to degenerate into a terminal-popularity signal, see Appendix B.7. On the other tasks, history-aware feedback gives clear additional gains over soft threshold: relative to Soft threshold, the best capacity-aware variant improves TFBIND8 from 69.0 to 81.0, QM9STR from 184.0 to 252.8, and SEHSTR from 107.4 to 131.8. PURE-CAS without the threshold is close to standard sampling on several tasks, so capacity feedback is most useful after the low-capacity tail has been softened. The RAWCOUNT-CAS control is inconsistent and often worse, indicating that raw usage alone is a poor congestion signal without normalization by frozen-policy capacity.

Budget scaling. Figure 1 shows mode discovery across the budget sweep, where the endpoint pattern persists across sampling budgets. On QM9STR and SEHSTR, State-CAS separates sharply from standard i.i.d. and nucleus sampling as the pool grows. On TFBIND8 and TFBIND10, State-CAS strongly improves over i.i.d. and remains competitive with nucleus. On BAG, the threshold component carries most of the gain across all budgets and adding history feedback gives little extra. TFBIND10 stays close to the baselines throughout the sweep, matching its weak endpoint result in Table 2.

Sensitivity to the soft threshold. Figure 2 sweeps $p_{\min}$ from 0 to 1 at $N = 2000$. The mode-discovery curve peaks at an interior value of $p_{\min}$ on every task, consistent with the trade-off in Proposition 3: an intermediate soft reweighting can achieve strictly better finite-pool coverage than either i.i.d. allocation or hard truncation.

Raw overlap is not congestion. Table 3 reports two raw structural metrics. Duplicate rate measures how often generation revisits an already generated terminal object, while prefix overlap measures how much generated trajectories reuse the same nonterminal prefix states. On TFBIND8, QM9STR, and SEHSTR, CAS increases prefix overlap relative to standard sampling; on TFBIND8 and QM9STR, duplicate rate also increases. Yet mode discovery improves sharply. This makes the distinction explicit: raw overlap measures structural sharing, not congestion. CAS acts on load relative to frozen-policy capacity, allowing high-capacity shared structure to carry more flow while redirecting sampling away from locally congested alternatives. On QM9STR, for example, prefix overlap rises from 0.526 to 0.638 while discovered modes increase from 51.8 to 252.8.

Table 3: Duplicate rate and prefix overlap at matched budget $N = 2000$. Each entry reports mean $\pm$ sample SE over five matched seeds (42–46), computed as $s/\sqrt{5}$ with sample standard deviation s using $\mathrm{ddof} = 1$. Nucleus sampling uses $p = 0.8$.

	Duplicate rate				Prefix overlap			
Task	Standard	Nucleus	State-CAS	Trans-CAS	Standard	Nucleus	State-CAS	Trans-CAS
BAG	0.148 ± 0.002	0.178 ± 0.002	0.089 ± 0.002	0.130 ± 0.004	0.576 ± 0.002	0.609 ± 0.001	0.558 ± 0.002	0.568 ± 0.002
TFBIND8	0.037 ± 0.001	0.117 ± 0.001	0.297 ± 0.005	0.255 ± 0.004	0.639 ± 0.001	0.730 ± 0.001	0.793 ± 0.001	0.778 ± 0.001
TFBIND10	0.000 ± 0.000	0.005 ± 0.001	0.000 ± 0.000	0.001 ± 0.000	0.474 ± 0.001	0.521 ± 0.001	0.475 ± 0.001	0.479 ± 0.001
QM9STR	0.014 ± 0.001	0.037 ± 0.001	0.108 ± 0.003	0.089 ± 0.003	0.526 ± 0.001	0.604 ± 0.001	0.638 ± 0.001	0.629 ± 0.000
SEHSTR	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.000	0.002 ± 0.000	0.338 ± 0.000	0.381 ± 0.000	0.430 ± 0.001	0.430 ± 0.001

5 Related Work

GFlowNets and finite-pool deployment. GFlowNets were introduced as amortized samplers for discrete objects with probability proportional to reward (Bengio et al., 2021; 2023), with training objectives such as trajectory balance and subtrajectory balance (Malkin et al., 2022; Madan et al., 2023). Pan et al. (2023) improve training by exploiting local credit on partial trajectories, and Tiapkin et al. (2024) cast GFlowNets as maximum-entropy reinforcement learning. Other work improves training through replay, local search, Thompson sampling, or distributed decomposition (Shen et al., 2023; Kim et al., 2024; Silva et al., 2025). Batch active learning and BatchGFN optimize set-valued utilities during learning (Malik et al., 2023). All of these methods modify how P_F is learned. CAS instead modifies how a frozen P_F is sampled from at deployment.

Decoding, diversity, and non-i.i.d. generation. Temperature scaling, top-k, and nucleus sampling are widely used history-blind decoding rules (Fan et al., 2018; Holtzman et al., 2019). They transform local logits in the same way for every draw and cannot adapt to how the current pool has loaded local resources relative to capacity. Post-hoc subset methods such as determinantal point processes promote diversity after candidates are generated (Kulesza & Taskar, 2012), while particle guidance, metadynamics-style history penalties, Stein variational particle methods, and diverse beam methods couple samples during generation (Corso et al., 2024; Laio & Parrinello, 2002; Liu & Wang, 2016; Vijayakumar et al., 2016; Meister et al., 2021). CAS is closest to these non-i.i.d. finite-pool methods, but its feedback is specific to frozen GFlowNet DAGs: it adjusts local probabilities according to load relative to frozen-policy capacity, rather than using a generic distance or kernel between completed samples.

Congestion models. The capacity-relative penalty borrows the volume-to-capacity formulation from transportation network equilibrium (Wardrop, 1952; Beckmann et al., 1956; Bureau of Public Roads, 1964), and the BPR form of equation 15 comes from the same literature. Stochastic user equilibrium adds an entropy term and logit route choice (Fisk, 1980; Daganzo & Sheffi, 1977), and Recursive Logit computes soft trajectory responses on graphs (Fosgerau et al., 2013). This provides the design principle for CAS: the frozen policy defines local capacity, and the inference-time correction responds to realized load relative to that capacity.

6 Limitations and Discussion

Firstly, CAS intentionally biases the sampler away from p_0. Users who need unbiased estimates under the frozen marginal should use standard sampling or add importance correction. The method also depends on the frozen policy as a capacity model. If P_F is poorly calibrated, the ratios in equation 12–equation 13 become noisy. The threshold is soft for the same reason: a rare action may still reach a valuable mode and should not be deleted from the support. Secondly, the effectiveness of history feedback depends on whether the state representation preserves trajectory-level usage information. In the sequence-structured and graph-structured benchmarks (TFBind8/10, QM9Str, sEHStr), intermediate-state counts retain such information and support capacity-relative feedback. In Bag, unordered multiset states merge distinct action histories, causing the capacity ratio to track terminal-marginal popularity rather than trajectory-level load (Appendix B.7).

7 Conclusion

We revisited a question that begins after GFlowNet training ends: given a frozen forward policy P_F and a finite sampling budget N, how should one construct the candidate pool? We showed that finite-budget i.i.d. sampling exhibits diminishing returns in terminal coverage, does not adapt local allocation to realized usage relative to frozen-policy capacity, and that the optimal fixed i.i.d. allocation for finite-pool coverage is generally not reward-proportional. CAS addresses these deployment-time effects through a local KL-regularized reweighting of the frozen policy, combining a soft threshold on low-probability actions with capacity-relative history feedback. CAS requires no retraining or additional reward queries and preserves the support of the frozen policy. Across five benchmarks, it improves mode discovery over i.i.d. sampling at matched budget, establishing finite-budget deployment as a distinct allocation problem beyond reward-proportional training.

AI USE STATEMENT

Generative AI tools assisted with literature searches, theoretical consistency checks, formulation and editing of proofs, language editing, and LaTeX preparation. The authors are responsible for the final content.

REFERENCES

Martin Beckmann, Charles B McGuire, and Christopher B Winsten. Studies in the economics of transportation. Technical report, 1956.

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

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023.

Bureau of Public Roads. Traffic assignment manual. Technical report, U.S. Department of Commerce, Urban Planning Division, Washington, D.C., 1964.

Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: non-iid diverse sampling with diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 22480–22507, 2024.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

Carlos F. Daganzo and Yossi Sheffi. On stochastic models of traffic assignment. *Transportation Science*, 11(3):253–274, 1977.

Pedro Dall'Antonia, Tiago da Silva, Daniel Csillag, Salem Lahlou, and Diego Mesquita. Avoid what you know: Divergent trajectory balance for gflownets. *arXiv preprint arXiv:2602.17827*, 2026.

Joseph L Doob. Conditional brownian motion and the boundary limits of harmonic functions. *Bulletin de la Société mathématique de France*, 85:431–458, 1957.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

Caroline Fisk. Some developments in equilibrium traffic assignment. *Transportation Research Part B: Methodological*, 14(3):243–255, 1980.

Mogens Fosgerau, Emma Frejinger, and Anders Karlstrom. A link based network route choice model with unrestricted choice set. *Transportation Research Part B: Methodological*, 56:70–80, 2013.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Minsu Kim, Taeyoung Yun, Emmanuel Bengio, Dinghuai Zhang, Yoshua Bengio, Sungsoo Ahn, and Jinkyoo Park. Local search gflownets. In *International Conference on Learning Representations*, volume 2024, pp. 42296–42313, 2024.

Minsu Kim, Sanghyeok Choi, Taeyoung Yun, Emmanuel Bengio, Leo Feng, Jarrid Rector-Brooks, Sungsoo Ahn, Jinkyoo Park, Nikolay Malkin, and Yoshua Bengio. Adaptive teachers for amortized samplers. In *International Conference on Learning Representations*, volume 2025, pp. 9315–9341, 2025.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, 2002.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

Kanika Madan, Jarrid Rector-Brooks, Maksym Korablyov, Emmanuel Bengio, Moksh Jain, Andrei Cristian Nica, Tom Bosc, Yoshua Bengio, and Nikolay Malkin. Learning gflownets from partial episodes for improved convergence and stability. In *International Conference on Machine Learning*, pp. 23467–23483. PMLR, 2023.

Kanika Madan, Alex Lamb, Emmanuel Bengio, Glen Berseth, and Yoshua Bengio. Towards improving exploration through sibling augmented gflownets. In *International Conference on Learning Representations*, volume 2025, pp. 89636–89654, 2025.

Shreshth A Malik, Salem Lahlou, Andrew Jesson, Moksh Jain, Nikolay Malkin, Tristan Deleu, Yoshua Bengio, and Yarin Gal. Batchgfn: Generative flow networks for batch active learning. *arXiv preprint arXiv:2306.15058*, 2023.

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

Clara Meister, Martina Forster, and Ryan Cotterell. Determinantal beam search. In *Proceedings of ACL-IJCNLP*, pp. 6551–6562, 2021.

Ling Pan, Nikolay Malkin, Dinghuai Zhang, and Yoshua Bengio. Better training of gflownets with local credit and incomplete trajectories. In *International Conference on Machine Learning*, pp. 26878–26890. PMLR, 2023.

Max W Shen, Emmanuel Bengio, Ehsan Hajiramezanali, Andreas Loukas, Kyunghyun Cho, and Tommaso Biancalani. Towards understanding and improving gflownet training. In *International conference on machine learning*, pp. 30956–30975. PMLR, 2023.

Tiago Silva, Amauri H Souza, Omar Rivasplata, Vikas Garg, Samuel Kaski, and Diego Mesquita. Generalization and distributed learning of gflownets. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

John Glen Wardrop. Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers*, 1(3):325–362, 1952.

A Proofs and Supporting Theory

A.1 Finite-pool coverage

Proof of Theorem 1. Consider the optimization problem

$$\max_{q\in\Delta(\mathcal{M})} F_N(q) = \sum_{m\in\mathcal{M}} w_m\big[1-(1-q_m)^N\big], \qquad w_m>0, \quad N>1, \tag{17}$$

where $\Delta(\mathcal{M}) = \{q : \sum_m q_m = 1,\ q_m \geq 0\}$.

Concavity. Computing second derivatives,

$$\frac{\partial^2 F_N}{\partial q_m\,\partial q_\ell} = \begin{cases} -N(N-1)\,w_m(1-q_m)^{N-2}, & m=\ell, \\ 0, & m\neq\ell. \end{cases} \tag{18}$$

The Hessian is therefore diagonal with entries in $[-N(N-1)\max_m w_m,\ 0]$ on the simplex, hence negative semidefinite. Consequently F_N is concave on $\Delta(\mathcal{M})$, and strictly concave on the relative interior of any face on which all active coordinates satisfy $q_m < 1$. KKT conditions are therefore both necessary and sufficient for global optimality.

KKT system. The Lagrangian with simplex equality constraint and nonnegativity multipliers $\mu_m \geq 0$ is

$$\mathcal{L}(q,\lambda,\mu) = \sum_{m\in\mathcal{M}} w_m\big[1-(1-q_m)^N\big] - \lambda\left(\sum_{m\in\mathcal{M}} q_m - 1\right) + \sum_{m\in\mathcal{M}} \mu_m q_m. \tag{19}$$

Stationarity, primal feasibility, and complementary slackness give, for each m,

$$N w_m (1-q_m)^{N-1} = \lambda - \mu_m, \tag{20}$$

$$q_m \geq 0, \qquad \mu_m \geq 0, \qquad \mu_m q_m = 0, \tag{21}$$

$$\sum_{m\in\mathcal{M}} q_m = 1. \tag{22}$$

Active and inactive coordinates. Define $\alpha := \lambda/N \geq 0$. We split coordinates into two cases.

Active case ($q_m > 0$). Complementary slackness gives $\mu_m = 0$, so equation 20 becomes $(1-q_m)^{N-1} = \alpha/w_m$, yielding

$$q_m = 1 - \left(\tfrac{\alpha}{w_m}\right)^{1/(N-1)}. \tag{23}$$

Since $q_m > 0$, this requires $w_m > \alpha$.

Inactive case ($q_m = 0$). Stationarity gives $\mu_m = \lambda - N w_m \geq 0$, i.e. $w_m \leq \alpha$.

The two cases are consistent, and together imply that the optimum has the threshold form

$$q_m^*(\alpha) = \Big[1 - \big(\tfrac{\alpha}{w_m}\big)^{1/(N-1)}\Big]_+. \tag{24}$$

Existence and uniqueness of the threshold α. Define

$$g(\alpha) := \sum_{m\in\mathcal{M}} \Big[1-(\alpha/w_m)^{1/(N-1)}\Big]_+, \qquad \alpha\in[0,\ w_{\max}], \tag{25}$$

where $w_{\max} = \max_m w_m$. Each summand is continuous in α and weakly decreasing (strictly decreasing on $[0, w_m]$ and zero on $[w_m, w_{\max}]$); hence g is continuous and weakly decreasing on $[0, w_{\max}]$, with

$$g(0) = |\{m : w_m > 0\}| \geq 1, \qquad g(w_{\max}) = 0. \tag{26}$$

Moreover g is strictly decreasing on the interval where at least one coordinate is active. By the intermediate value theorem there exists a unique $\alpha^* \in [0, w_{\max})$ with $g(\alpha^*) = 1$, giving the unique simplex-feasible threshold solution equation 24.

Uniqueness of q^*. On the active set $\mathcal{A}^* := \{m : w_m > \alpha^*\}$, the strict concavity of F_N on the corresponding face yields a unique maximizer. If $w_m = \alpha^*$, equation 24 assigns that coordinate $q^*_m = 0$.

Degenerate case $N = 1$. The expression equation 23 contains $1/(N-1)$ and is therefore undefined for $N = 1$. In that case $F_1(q) = \sum_m w_m q_m$ is linear, and any maximizer is supported on $\arg\max_m w_m$; the threshold form is recovered only as the limit $N \downarrow 1$. □

Remark 2 (Unweighted terminal coverage)**.** *The unweighted formula in equation 2 follows by taking $I_x = \mathbf{1}[\exists i \le N : X_i = x]$ and summing $\mathbb{E}[I_x] = 1 - (1 - q(x))^N$ over x, which is the special case $w_m \equiv 1$ of F_N.*

A.2 Minimum-KL lifting of a terminal target

Proof of Lemma 1. Let Q be any trajectory distribution whose terminal-mode marginal equals q, i.e. $Q(M = m) = q_m$ for all m. We assume $\mathrm{supp}(q) \subseteq \mathrm{supp}(p_0^{\mathcal{M}})$; otherwise $D_{\mathrm{KL}}(Q\|p_0) = +\infty$ and the minimization is vacuous.

Chain rule for KL. Decomposing the trajectory KL by the mode marginal,

$$\begin{aligned} D_{\mathrm{KL}}(Q\|p_0) &= \sum_\tau Q(\tau) \log \tfrac{Q(\tau)}{p_0(\tau)} \\ &= \sum_{m\in\mathcal{M}} Q(M=m) \log \tfrac{Q(M=m)}{p_0^{\mathcal{M}}(m)} \\ &\quad + \sum_m Q(M=m) \sum_\tau Q(\tau \mid M=m) \log \tfrac{Q(\tau|M=m)}{p_0(\tau|M=m)} \\ &= D_{\mathrm{KL}}\big(q \,\|\, p_0^{\mathcal{M}}\big) + \sum_{m\in\mathcal{M}} q_m \, D_{\mathrm{KL}}\big(Q(\tau \mid M=m) \,\|\, p_0(\tau \mid M=m)\big). \end{aligned} \tag{27}$$

The first term is fixed by the constraint $Q(M = m) = q_m$; the second is a sum of nonnegative terms, each minimized uniquely (when $q_m > 0$) at $Q(\tau \mid M = m) = p_0(\tau \mid M = m)$. Therefore

$$Q_q^\dagger(\tau) = q_{M(\tau)}\, p_0(\tau \mid M(\tau)) = p_0(\tau)\, \frac{q_{M(\tau)}}{p_0^{\mathcal{M}}(M(\tau))} = p_0(\tau)\, r_q(M(\tau)), \tag{28}$$

where $r_q(m) := q_m / p_0^{\mathcal{M}}(m)$.

Local conditional under $Q_q^\dagger$. Fix a non-sink state s visited with positive probability under p_0, and an action $a \in \mathcal{A}(s)$. Let $\mathcal{T}_s = \{\tau : s \in \tau\}$ and $\mathcal{T}_{s,a} = \{\tau : \tau \text{ uses } (s,a)\}$. Then

$$Q_q^\dagger(a \mid s) = \frac{\sum_{\tau\in\mathcal{T}_{s,a}} Q_q^\dagger(\tau)}{\sum_{\tau\in\mathcal{T}_s} Q_q^\dagger(\tau)} = \frac{\sum_{\tau\in\mathcal{T}_{s,a}} p_0(\tau)\, r_q(M(\tau))}{\sum_{\tau\in\mathcal{T}_s} p_0(\tau)\, r_q(M(\tau))}. \tag{29}$$

Define the prefix mass $p_0(s) := \sum_{\sigma: s_0 \to \cdots \to s} \prod_t P_F$. By the definition of $V_q(s,a)$, the numerator of equation 29 equals $p_0(s) P_F(a \mid s) V_q(s,a)$. Summing over actions, the denominator equals $p_0(s) \sum_{b\in\mathcal{A}(s)} P_F(b \mid s) V_q(s,b)$. The shared factor $p_0(s)$ cancels, giving

$$Q_q^\dagger(a \mid s) = \frac{P_F(a \mid s) V_q(s,a)}{\sum_{b\in\mathcal{A}(s)} P_F(b \mid s) V_q(s,b)}, \tag{30}$$

which is equation 8. If $\sum_{b\in\mathcal{A}(s)} P_F(b \mid s) V_q(s,b) = 0$, then $Q_q^\dagger$ assigns zero mass to trajectories visiting s, so the conditional is irrelevant for sampling.

Local variational form. For any $\pi \in \Delta(\mathcal{A}(s))$, set $\kappa_q^\dagger(s,a) = -\log V_q(s,a)$ (well-defined whenever $V_q(s,a) > 0$; set to $+\infty$ otherwise so that the corresponding action receives zero probability). Then

$$D_{\mathrm{KL}}\big(\pi \,\|\, P_F(\cdot \mid s)\big) + \sum_a \kappa_q^\dagger(s,a)\, \pi(a) = \sum_a \pi(a) \log \frac{\pi(a)}{P_F(a \mid s)\, V_q(s,a)}, \tag{31}$$

and applying Proposition 2 (proved in Section A) with $\kappa(a) = \kappa_q^\dagger(s,a)$ gives a unique minimizer matching equation 30, up to an additive constant in $\kappa_q^\dagger(s,\cdot)$ (since softmax is invariant under uniform shifts of its logits). □

A.3 RESOURCE-REUSE IDENTITY

Proof of Proposition 1. Fix a resource $j \in \mathcal{J}$. Expanding the binomial coefficient,

$$\binom{C_N(j)}{2} = \tfrac{1}{2} C_N(j)(C_N(j)-1) = \sum_{1 \le i < \ell \le N} B_{j,\tau_i} B_{j,\tau_\ell}, \tag{32}$$

where we used $B_{j,\tau_i}^2 = B_{j,\tau_i}$ since $B_{j,\tau_i} \in \{0,1\}$. Trajectories are i.i.d., so for $i \neq \ell$,

$$\mathbb{E}\big[B_{j,\tau_i} B_{j,\tau_\ell}\big] = \Pr(B_{j,\tau_i}=1)\,\Pr(B_{j,\tau_\ell}=1) = z_p(j)^2. \tag{33}$$

Taking expectations of equation 32 gives $\mathbb{E}\big[\binom{C_N(j)}{2}\big] = \binom{N}{2} z_p(j)^2$. Summing with weights $w_j \ge 0$ yields equation 5. □

Remark 3 (Variance and concentration)**.** *$C_N(j) \sim \mathrm{Bin}(N, z_p(j))$, so*

$$\mathrm{Var}\Big[\binom{C_N(j)}{2}\Big] = \binom{N}{2} z_p(j)^2\big(1-z_p(j)\big)\big(1+(2N-3)z_p(j)\big). \tag{34}$$

In particular, when $N\, z_p(j) \to \infty$,

$$\frac{\binom{C_N(j)}{2}}{\binom{N}{2} z_p(j)^2} \xrightarrow{\Pr} 1, \tag{35}$$

so resources with the largest $z_p(j)$ dominate L_N not only in expectation but also with high probability for sufficiently large N. This justifies the claim in the main text that pairwise reuse concentrates on the heavy-mass resources of p.

A.4 STATIC THRESHOLDING AS A ONE-STEP FINITE-POOL UTILITY

A fixed thresholding rule such as nucleus sampling keeps only the smallest set $H_p(s)$ whose frozen-policy mass is at least p (Holtzman et al., 2019). Let $T_p(s) = \mathcal{A}(s) \setminus H_p(s)$, $\delta_p(s) = P_F(T_p(s) \mid s)$, and let $V(s,a) \in [0,1]$ be a bounded one-step downstream utility (e.g., the indicator that the trajectory continuation from $T(s,a)$ discovers a previously unseen mode within the remaining budget). Define the per-state one-step utilities

$$U_{\mathrm{std}}(s) = \mathbb{E}_{a \sim P_F(\cdot|s)}[V(s,a)], \qquad U_{\mathrm{nuc}}(s) = \mathbb{E}_{a \sim P_F(\cdot|s,\, a \in H_p(s))}[V(s,a)]. \tag{36}$$

With $\mu_H(s) = \mathbb{E}[V(s,a) \mid a \in H_p(s)]$ and $\mu_T(s) = \mathbb{E}[V(s,a) \mid a \in T_p(s)]$, decomposing the standard expectation by membership in $H_p(s)$ gives

$$U_{\mathrm{std}}(s) = (1-\delta_p(s))\,\mu_H(s) + \delta_p(s)\,\mu_T(s), \qquad U_{\mathrm{nuc}}(s) = \mu_H(s), \tag{37}$$

hence

$$U_{\mathrm{nuc}}(s) - U_{\mathrm{std}}(s) = \delta_p(s)\big(\mu_H(s) - \mu_T(s)\big). \tag{38}$$

Static thresholding helps when the kept set has larger one-step finite-pool utility than the removed set, and can hurt when useful modes lie outside the threshold.

A.5 PROOFS FOR CAS COMPONENTS

Proof of Proposition 2. We minimize

$$J(\pi) = \sum_{a \in \mathcal{A}(s)} \pi(a) \log \frac{\pi(a)}{P_F(a \mid s)} + \sum_{a \in \mathcal{A}(s)} \kappa(a)\,\pi(a) \tag{39}$$

over $\pi \in \Delta(\mathcal{A}(s))$, with the convention $0 \log 0 = 0$ and $x \log(x/0) = +\infty$ for $x > 0$. The latter convention forces $\pi(a) = 0$ whenever $P_F(a \mid s) = 0$; otherwise $J(\pi) = +\infty$. If $P_F(a \mid s) = 0$ or $\kappa(a) = +\infty$, any distribution with $\pi(a) > 0$ has infinite objective. Hence, under the assumed positive normalizer, we may restrict optimization to the nonempty set

$$\mathcal{S} = \{a : P_F(a \mid s) > 0,\ \kappa(a) < +\infty\}.$$

Strict convexity. The map $\pi \mapsto \sum_a \pi(a) \log \pi(a)$ is strictly convex on the relative interior of $\Delta(\mathcal{S})$; the linear terms $-\sum_a \pi(a) \log P_F(a \mid s)$ and $\sum_a \kappa(a)\pi(a)$ preserve strict convexity. Therefore J admits a unique minimizer on $\Delta(\mathcal{S})$.

First-order condition. The Lagrangian for the simplex constraint is

$$\mathcal{L}(\pi, \eta) = \sum_{a \in \mathcal{S}} \pi(a) \log \frac{\pi(a)}{P_F(a \mid s)} + \sum_{a \in \mathcal{S}} \kappa(a)\pi(a) + \eta \left(\sum_{a \in \mathcal{S}} \pi(a) - 1 \right), \tag{40}$$

with stationarity

$$\log \frac{\pi(a)}{P_F(a \mid s)} + 1 + \kappa(a) + \eta = 0 \quad \iff \quad \pi(a) = P_F(a \mid s)\, e^{-\kappa(a)}\, e^{-(1+\eta)}. \tag{41}$$

The simplex constraint determines the multiplier:

$$\sum_{a \in \mathcal{S}} \pi(a) = 1 \iff e^{1+\eta} = \sum_{b \in \mathcal{S}} P_F(b \mid s)\, e^{-\kappa(b)}. \tag{42}$$

Substituting back into equation 41 yields

$$\pi^*(a) = \frac{P_F(a \mid s)\, e^{-\kappa(a)}}{\sum_{b \in \mathcal{A}(s)} P_F(b \mid s)\, e^{-\kappa(b)}}, \tag{43}$$

which is equation 11. Setting $P_F(a \mid s) = 0$ gives $\pi^*(a) = 0$; a finite constant κ cancels in numerator and denominator, recovering $\pi^* = P_F$. □

A.6 Interior reallocation can improve over i.i.d. and hard truncation

Proof of Proposition 3. Recall the two-region finite-pool model

$$F_N(\alpha) = K_H \left[1 - \left(1 - \tfrac{\alpha \theta_H}{K_H} \right)^N \right] + K_T \left[1 - \left(1 - \tfrac{(1-\alpha)\theta_T}{K_T} \right)^N \right], \quad \alpha \in [0, 1]. \tag{44}$$

Strict concavity of F_N. Differentiating,

$$F_N'(\alpha) = N\theta_H \left(1 - \tfrac{\alpha \theta_H}{K_H} \right)^{N-1} - N\theta_T \left(1 - \tfrac{(1-\alpha)\theta_T}{K_T} \right)^{N-1}, \tag{45}$$

$$F_N''(\alpha) = -\tfrac{N(N-1)\theta_H^2}{K_H} \left(1 - \tfrac{\alpha \theta_H}{K_H} \right)^{N-2} - \tfrac{N(N-1)\theta_T^2}{K_T} \left(1 - \tfrac{(1-\alpha)\theta_T}{K_T} \right)^{N-2}. \tag{46}$$

For $N \geq 2$ and $(\alpha\theta_H/K_H, (1-\alpha)\theta_T/K_T) \in [0, 1)^2$, $F_N''(\alpha) < 0$, so F_N is strictly concave on $[0, 1]$.

Existence of an interior maximizer. The endpoint derivatives are

$$F_N'(p_H) = N\theta_H \left(1 - \tfrac{p_H \theta_H}{K_H} \right)^{N-1} - N\theta_T \left(1 - \tfrac{(1-p_H)\theta_T}{K_T} \right)^{N-1}, \tag{47}$$

$$F_N'(1) = N\theta_H \left(1 - \tfrac{\theta_H}{K_H} \right)^{N-1} - N\theta_T. \tag{48}$$

Strict concavity together with $F_N'(p_H) > 0$ and $F_N'(1) < 0$ implies that F_N' has a unique zero $\alpha^* \in (p_H, 1)$, and $F_N(\alpha^*) > \max\{F_N(p_H), F_N(1)\}$. So both i.i.d. sampling ($\alpha = p_H$) and hard truncation ($\alpha = 1$) are strictly suboptimal.

Sufficient conditions for the endpoint inequalities. Equation 48 gives $F_N'(1) < 0$ iff

$$\theta_T \; > \; \theta_H \left(1 - \tfrac{\theta_H}{K_H} \right)^{N-1}, \tag{49}$$

which holds whenever the head is partially saturated, i.e. when N is large enough that the marginal coverage value of an additional head draw falls below θ_T. Equation 47 gives $F_N'(p_H) > 0$ iff

$$\theta_H \left(1 - \tfrac{p_H \theta_H}{K_H} \right)^{N-1} \; > \; \theta_T \left(1 - \tfrac{(1-p_H)\theta_T}{K_T} \right)^{N-1}, \tag{50}$$

which holds whenever the head is more useful per draw than the tail at the i.i.d. allocation. Both equation 49 and equation 50 hold simultaneously in the typical regime $\theta_H > \theta_T > 0$ with N large enough to partially saturate the head but not exhaust the tail. □

B DETAILED EXPERIMENTAL PROTOCOL AND RESULTS

We evaluate frozen GFlowNet checkpoints on BAG, TFBIND8, TFBIND10, QM9STR, and SEHSTR. Task names follow Shen's benchmark convention: TFBIND8 corresponds to SIX6, TFBIND10 corresponds to PHO4, and QM9STR and SEHSTR are the string-MDP versions of QM9 and sEH. All five checkpoints follow the released implementation of Shen et al. (2023) and its task-specific reward transformations, construction MDPs, and mode predicates. All samplers use the same checkpoint, valid-action mask, reward function, sample budget, and matched seed set within each task. No method updates model parameters or calls the reward model during generation. We report mean $\pm$ standard error (SE) over five matched seeds, following Shen's convention. The main matched-budget comparison uses $N = 2000$. All experiments are run on a single NVIDIA RTX 6000 Ada Generation GPU (48 GB) on a server with 2× AMD EPYC 7643 CPUs and 1 TB RAM.

B.1 TASK-SPECIFIC SCORING PROTOCOL

Table 4 specifies the per-task reward exponent, construction MDP, and mode definition used to compute absolute mode counts.

Table 4: Per-task protocol used to score the frozen-deployment pools. "Scale" is the post-exponent reward maximum. Mode discovery counts in Table 1 are raw counts under the listed predicate, not percentages.

Task	Reward transform	MDP type	Mode predicate / denominator
BAG	Categorical reward; no exponent; values 0.01, 10, 30	Multiset bag MDP; add-one-symbol actions; forced length 13	Mode iff the code-generated bag state has reward 30; substructure threshold is 7 repeated symbols.
TFBIND8	Reward exponent 3; scale 10; floor 10^{-3}	Sequence insertion MDP under the released setting `mdp_style=insert`; length 8 over 4 symbols	Fixed top-mode set loaded from `modes_tfbind8.pkl`; counts are against that code predicate over $4^8 = 65{,}536$ sequences.
TFBIND10	Sigmoid normalization, reward exponent 3; scale 10	Sequence prepend–append MDP; length 10 over 4 symbols	Top 0.5% by scaled reward, i.e. about $5{,}243$ modes out of $4^{10} = 1{,}048{,}576$.
QM9STR	Reward exponent 5; scale 100; floor 10^{-3}	Molecular string MDP, implemented as sequence prepend–append over two-stem blocks; length 5	Top 0.5% by scaled oracle reward, i.e. 293 modes out of $58{,}765$ enumerated molecules.
SEHSTR	Reward exponent 6; scale 10; floor 10^{-3}	Molecular string MDP, implemented as sequence prepend–append over 18 two-stem blocks; length 6	Released-code predicate uses top 0.1% by scaled reward, i.e. about $34{,}012$ modes out of $34{,}012{,}224$. The ICML text describes a top-0.5% denominator ($\approx 170{,}061$); our reported counts follow the released code and should be compared under the same predicate.

The absolute counts in Table 1 should be read against the mode set in Table 4, not as normalized discovery rates. In particular, the headline SEHSTR count $4.4 \rightarrow 131.8$ is a matched-seed comparison under the same frozen checkpoint and the same code-defined mode predicate. If one wants a direct numerical comparison to the Shen paper's stated top-0.5% sEH convention, the mode predicate must be recomputed before comparing absolute counts.

B.2 COMPLEMENTARITY WITH EXPLORATION-ORIENTED GFLOWNET TRAINING

The main experiments evaluate CAS as an inference-time modification of a fixed frozen GFlowNet. A natural question is whether the observed gains are specific to a standard Trajectory Balance (TB) model, or whether CAS remains useful when the frozen policy is obtained using training methods that explicitly encourage stronger exploration or diversity.

Table 5: Mode discovery at $N = 2000$ when CAS is applied to frozen GFlowNets obtained from different training methods. For each method, three checkpoints are independently trained and frozen. Each entry first averages ten matched inference seeds per checkpoint and then averages across the three checkpoints. Numbers in parentheses report the absolute gain over i.i.d. deployment under the same training method. CAS improves finite-budget discovery for every training method on both tasks.

	TFBIND8			QM9STR		
Frozen training method	IID	State-CAS	Trans.-CAS	IID	State-CAS	Trans.-CAS
Standard TB	44.23	77.03 (+32.80)	77.20 (+32.97)	60.37	180.50 (+120.13)	180.33 (+119.96)
Adaptive Teachers	40.40	73.73 (+33.33)	74.70 (+34.30)	60.93	181.07 (+120.14)	180.67 (+119.74)
SA-GFN	44.00	75.63 (+31.63)	76.67 (+32.67)	60.20	183.90 (+123.70)	184.37 (+124.17)
Divergent TB	43.97	77.73 (+33.76)	79.17 (+35.20)	61.87	185.50 (+123.63)	185.30 (+123.43)

To test this, we train GFlowNets using four training procedures: standard TB, Adaptive Teachers (Kim et al., 2025), Sibling-Augmented GFlowNet (SA-GFN) (Madan et al., 2025), and Divergent TB (DTB) (Dall'Antonia et al., 2026). Each method is trained under the same task-level training budget with three independent training seeds. We freeze the final checkpoint of each trained model and evaluate three deployment rules: standard i.i.d. sampling, State-CAS, and Transition-CAS. For each frozen checkpoint, we use ten matched inference seeds and a sampling budget of $N = 2000$. The CAS hyperparameters are kept fixed at the default values

$$(\lambda, \beta, t, p_{\min}) = (1, 1, 1, 0.2)$$

for all training methods and both tasks. We first average over the ten inference seeds for each training seed and then average across the three independently trained models. Adaptive Teachers, SA-GFN, and Divergent TB are implemented under our common training and evaluation protocol to isolate their interaction with frozen-policy deployment.

CAS remains beneficial across exploration-oriented training methods. Table 5 shows that the improvement from CAS is not specific to a standard TB checkpoint. On TFBIND8, the best capacity-aware variant improves mode discovery by $+32.97$, $+34.30$, $+32.67$, and $+35.20$ modes for standard TB, Adaptive Teachers, SA-GFN, and Divergent TB, respectively. On QM9STR, the corresponding gains are $+120.13$, $+120.14$, $+124.17$, and $+123.63$.

Importantly, each comparison is made within the same frozen trained model: the training procedure, checkpoint, reward function, valid-action set, and sample budget are unchanged, and only the deployment rule is replaced. The resulting gains therefore isolate the effect of inference-time finite-pool reallocation rather than differences in training quality.

These results support a complementary interpretation of training-time exploration and CAS. Exploration-oriented GFlowNet training changes the frozen policy P_F, whereas CAS addresses how a finite deployment budget is allocated after that policy has been trained. Improving the former does not remove the finite-pool diminishing-return effect characterized in Theorem 1; correspondingly, CAS continues to provide substantial additional mode discovery when applied to exploration-enhanced frozen policies.

B.3 DISCOVERY BEYOND THE TRAINING ARCHIVE

A finite deployment budget is useful only if it can add candidates beyond those already discovered during training. We therefore repeat the evaluation after removing every qualifying mode present in the full training archive and count only *archive-new* modes discovered after freezing the policy.

Each training run contains 128,000 training trajectories, which already cover most of the 328 qualifying TFBIND8 modes. Despite this limited remaining headroom, CAS consistently discovers additional modes absent from the training archive.

Even after most qualifying modes have already been absorbed by the training archive, post-training allocation still matters. Relative to i.i.d. deployment, Transition-CAS discovers 4.23, 4.77, and 5.83 additional archive-new modes for Adaptive Teachers, SA-GFN, and Divergent TB, respectively. Thus

Table 6: Archive-new mode discovery on TFBIND8. Training-archive modes are removed before evaluation; the remaining columns count only qualifying modes discovered after the policy is frozen. Archive coverage is the mean number of the 328 qualifying modes already found during training.

Training method	Archive coverage	IID	State-CAS	Trans.-CAS
Standard TB	293.33/328	2.30	4.03	3.63
Adaptive Teachers	290.00/328	2.47	6.57	6.70
SA-GFN	280.00/328	4.03	8.37	8.80
Divergent TB	279.67/328	3.47	8.83	9.30

the deployment gain is not obtained by rediscovering candidates already seen during training: CAS extracts additional coverage from the finite budget available after the policy is frozen.

B.4 SENSITIVITY OF HISTORY-BLIND BASELINES

We also sweep the main hyperparameters of the history-blind inference baselines. For each baseline, we vary one hyperparameter while keeping the other baseline defaults fixed: temperature sampling uses $T = 0.5$, Top-k uses $k = 3$, nucleus sampling uses $p = 0.8$, and ϵ-greedy uses $\epsilon = 0.3$. The swept grids are

$$T \in \{0.5, 1, 2, 3, 5, 7, 10\}, \quad k \in \{1, 2, 3, 5, 8, 12\},$$
$$p \in \{0.5, 0.7, 0.8, 0.9, 0.95, 0.99\}, \quad \epsilon \in \{0, 0.1, 0.2, 0.3, 0.5, 0.7, 1.0\}.$$

All runs use the same frozen checkpoints, sampling budget $N = 2000$, and five random seeds as in the main experiments.

Figures 3 and 4 report mode discovery and prefix overlap across the sweep. The results show that history-blind baselines are sensitive to their tuning and that no single setting dominates across tasks. Lower temperature can improve discovery on several benchmarks by concentrating probability mass on stronger local actions, but it remains fixed across the whole sample pool. Top-k and nucleus sampling can be strong when the useful modes lie in the local head of the frozen policy, but they are static truncation rules: the same local actions are kept or removed for every draw. This helps on some tasks and hurts on others. ϵ-greedy exploration is generally weak in this setting, since uniform random actions often spend budget on low-quality continuations rather than targeted new modes.

These sweeps motivate the tuned history-blind settings used in the main table: temperature $T = 0.5$, Top-k with $k = 3$, nucleus sampling with $p = 0.8$, and ϵ-greedy with $\epsilon = 0.3$. We use one global setting for each baseline across all tasks, rather than choosing a separate best value per task. This yields a matched comparison using globally tuned history-blind baselines. The resulting performance gap reflects the difference between static local transformations and history-aware finite-pool correction.

B.5 SENSITIVITY TO CAS HYPERPARAMETERS

We evaluate the sensitivity of CAS to the three penalty parameters λ, t, and β, using the same frozen checkpoints and evaluation protocol as in the main experiments. The parameter λ sets the overall strength of the correction, t controls when the capacity penalty becomes active, and β controls the growth rate after activation.

Activation threshold t. Figure 5 shows the effect of varying t. Smaller values activate the penalty earlier and therefore apply a stronger correction. Larger values delay activation, so the sampler stays closer to the frozen policy. The mode-discovery curves show that neither extreme is uniformly best. On some tasks, aggressive correction improves discovery; on others, weakening the penalty avoids suppressing useful repeated structure. The prefix-overlap curves also show that lower overlap is not the objective by itself. In several cases, mode discovery drops even when overlap decreases. These results support capacity-normalized reuse as the local feedback signal and show that raw overlap need not vary monotonically with mode discovery.

Penalty growth β. Figure 6 reports the sweep over β. Increasing β makes the penalty more selective: actions slightly above the threshold are only mildly affected, while actions with large

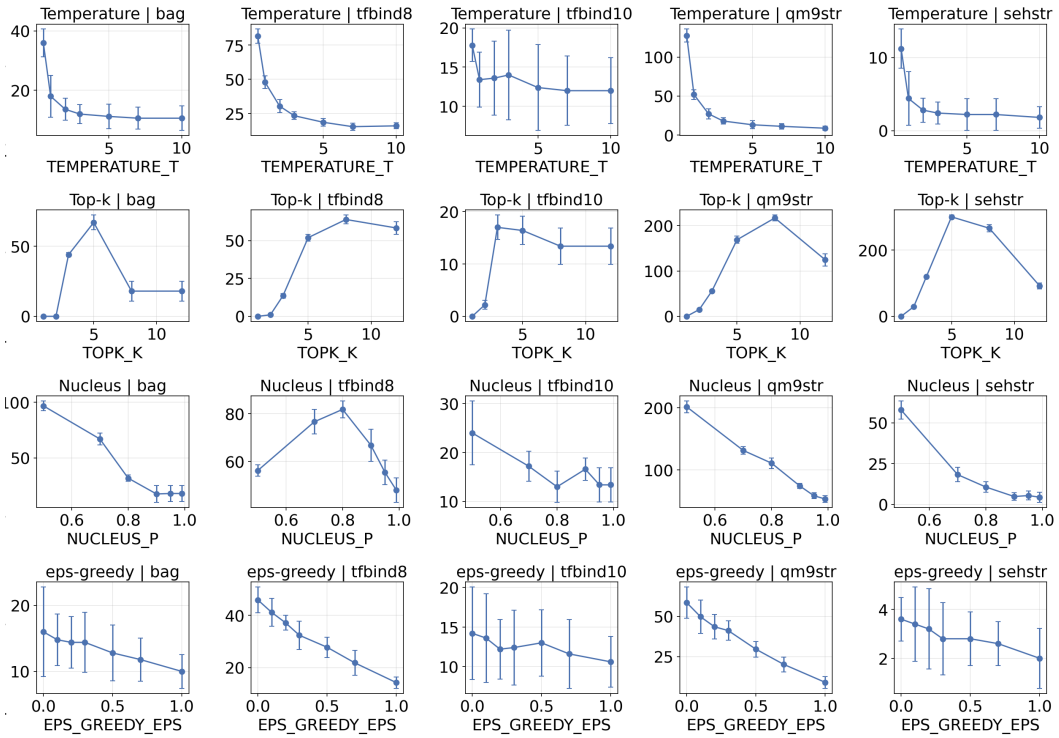


Figure 3: Sensitivity of history-blind baselines to their main hyperparameters, measured by mode discovery at $N = 2000$. Each row corresponds to one baseline and each column to one task.

capacity ratios are penalized more sharply. The best value is task-dependent. Bag and sEHStr benefit from a moderately larger β, where stronger suppression of overused local choices remains useful. On TFBind8 and QM9Str, however, very large β reduces mode discovery, which suggests that some repeated local structure is productive and should not be penalized too sharply. The prefix-overlap curves increase with β on several tasks, again showing that high overlap can coexist with strong discovery when the overlap occurs in productive regions of the DAG.

Penalty strength λ. Figure 7 shows the sweep over λ for State-CAS. Increasing λ from zero improves discovery on most tasks, which indicates that the capacity correction is doing more than passively renormalizing the frozen policy. The gains are not monotone, however. Very large values can reduce discovery when the correction moves too far from the frozen policy. This behavior reflects the role of CAS as a finite-pool correction anchored to the learned GFlowNet policy.

B.6 Matched-seed statistical support

For each headline comparison, we compute the seed-level difference between the best capacity-aware variant and standard i.i.d. sampling under the same checkpoint and random seed. Paired bootstrap intervals over these five differences are strictly positive on all tasks. The largest gain, sEHStr, increases from 4.4 to 131.8 discovered modes under the released-code top-0.1% mode predicate in Table 4; the paired mean difference is $+127.4$ with a 95% paired bootstrap interval $[119.8, 135.0]$. We also report exact paired Wilcoxon values in Table 7. With only five seeds, the smallest exact two-sided Wilcoxon value is 0.0625.

B.7 Why CAS underperforms soft threshold on Bag

Table 2 reports that on Bag, soft threshold alone (50.8 modes) outperforms both State-CAS (45.0) and Trans-CAS (42.4). This subsection works out the mechanism: on Bag, the state-level capacity ratio degenerates from a trajectory-level overuse signal into a terminal-level popularity signal, and

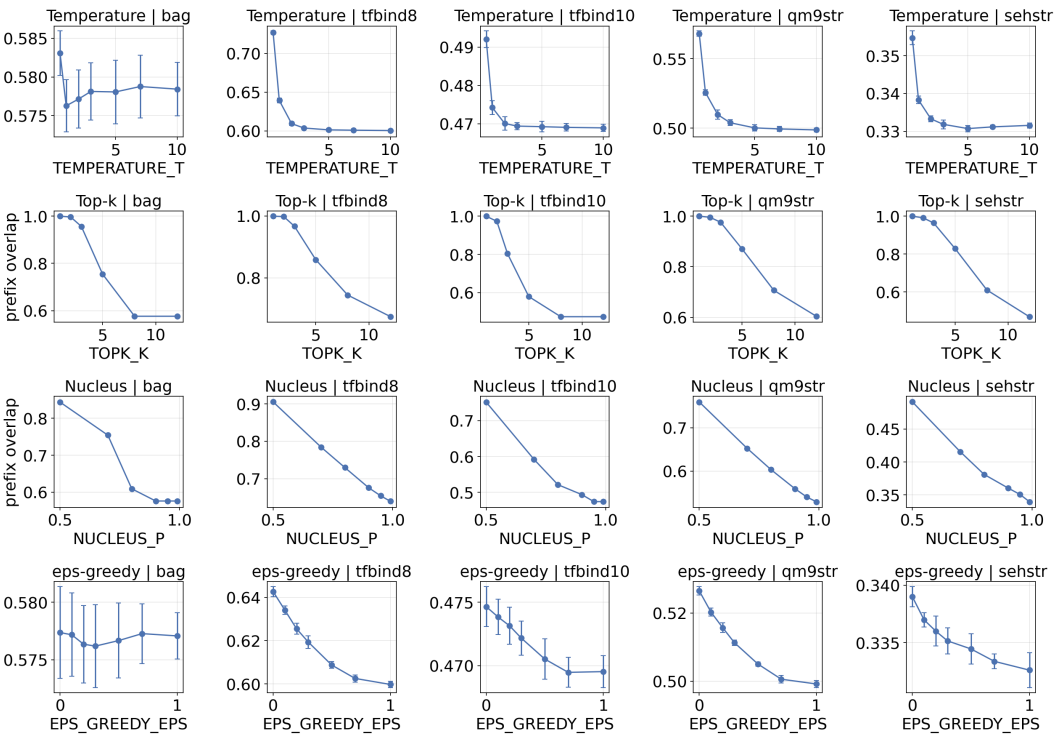
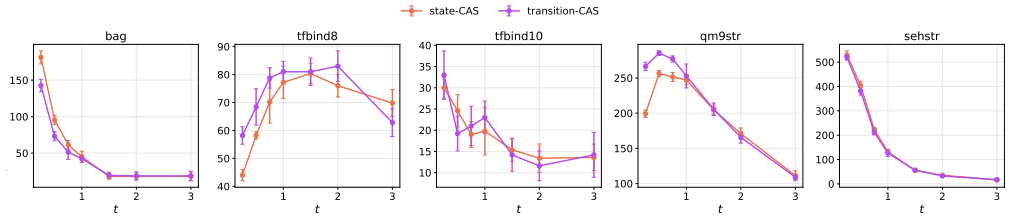
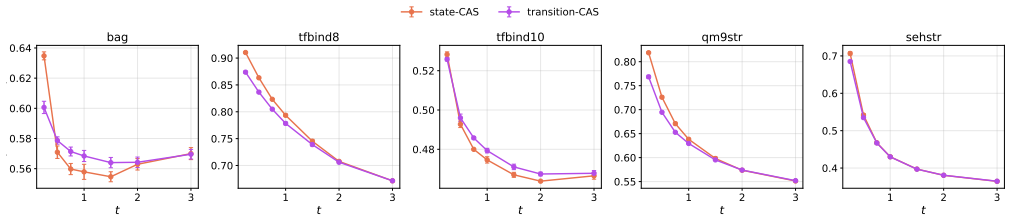


Figure 4: Sensitivity of history-blind baselines to their main hyperparameters, measured by prefix overlap at $N = 2000$. Static truncation or smoothing can change raw overlap, but lower overlap does not consistently imply better mode discovery.



(a) Mode discovery.

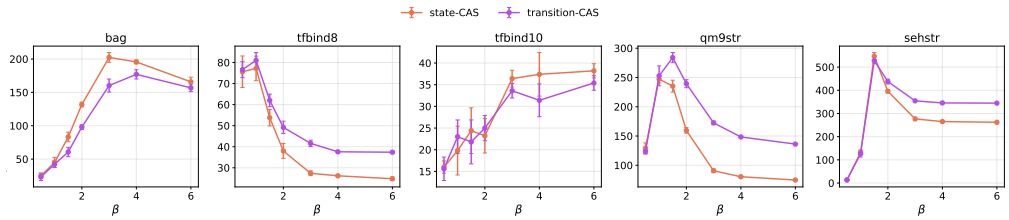


(b) Prefix overlap.

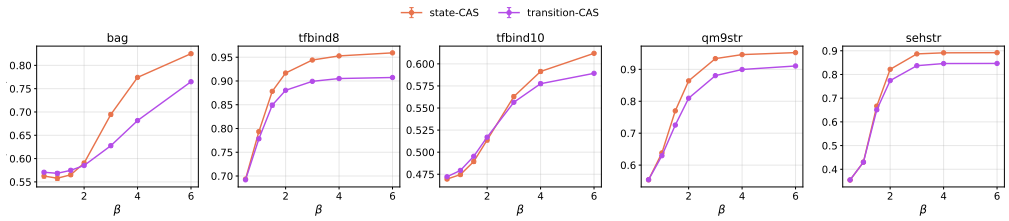
Figure 5: Sensitivity to the activation threshold t. Smaller t activates the capacity penalty earlier, while larger t keeps the sampler closer to the frozen policy.

once high-reward bags have been sampled, the penalty actively suppresses the intermediate multisets that lead to other unsampled high-reward bags.

BAG (Shen et al., 2023) is a synthetic GFlowNet benchmark in which a state is an unordered multiset of tokens drawn from a fixed alphabet, and an action either appends a token or terminates. Two



(a) Mode discovery.



(b) Prefix overlap.

Figure 6: Sensitivity to the penalty growth parameter β. Larger β makes the correction more selective above the activation threshold.

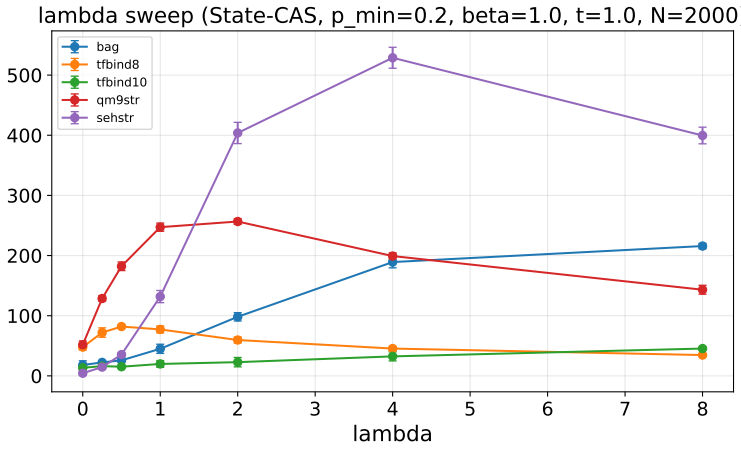


Figure 7: Sensitivity to the penalty strength λ for State-CAS. Moderate values usually improve mode discovery, while overly strong correction can hurt some tasks.

trajectories that add the same tokens in different orders reach the same state. Concretely, a terminal bag x of size n with token multiplicities $(k_1, k_2, \ldots)$ is reachable through $\binom{n}{k_1,k_2,\ldots}$ distinct action sequences, all passing through the same family of intermediate sub-multisets of x.

Visit count on a multiset state. Let $\mathcal{P}_{k-1} = \{x_1, \ldots, x_{k-1}\}$ denote the multiset of terminal bags sampled by trajectories $\tau_1, \ldots, \tau_{k-1}$. For an intermediate multiset u, let $u \subseteq x$ denote that u is a sub-multiset of x. Because every trajectory ending at x passes through some sub-multiset chain of x, and because the multiset structure of u does not encode the order in which its tokens were added,

$$C_{k-1}(u) \;=\; \sum_{i=1}^{k-1} \mathbb{1}\big[\tau_i \text{ visits } u\big] \;\approx\; \sum_{i=1}^{k-1} \mathbb{1}[u \subseteq x_i] \cdot \pi(u \mid x_i), \tag{51}$$

where $\pi(u \mid x_i)$ is the probability under P_F that the construction of x_i passes through u as one of its intermediate snapshots. The dominant factor in (51) is the indicator $\mathbb{1}[u \subseteq x_i]$: u contributes to $C_{k-1}(u)$ once for every previously sampled terminal bag that contains u as a sub-multiset.

Table 7: Matched-seed support for headline mode-discovery gains over standard i.i.d. sampling. Confidence intervals are paired bootstrap percentile intervals over the five seed-level differences. Wilcoxon p-values are exact, with the directional alternative reported first and the two-sided value in parentheses.

Task	Compared variant	Standard	CAS	Gain	Multiplicative gain	95% paired bootstrap CI; Wilcoxon p
BAG	State-CAS	18.0	45.0	+27.0	2.50×	[22.0, 30.8]; 0.031 (0.063)
TFBIND8	Trans.-CAS	47.8	81.0	+33.2	1.69×	[30.0, 36.8]; 0.031 (0.063)
TFBIND10	Trans.-CAS	13.4	23.0	+9.6	1.72×	[4.0, 13.4]; 0.063 (0.125)
QM9STR	Trans.-CAS	51.8	252.8	+201.0	4.88×	[183.4, 217.4]; 0.031 (0.063)
SEHSTR	State-CAS	4.4	131.8	+127.4	29.95×	[119.8, 135.0]; 0.031 (0.063)

State-level capacity ratio degenerates to a weighted terminal-popularity ratio. The frozen-policy capacity at u is

$$\bar{C}^{\text{state}}_{k-1}(u) \;=\; \sum_{(v,b):\, T(v,b)=u} C_{k-1}(v)\, P_F(b \mid v) \;\approx\; (k-1)\, p_0(u), \tag{52}$$

where $p_0(u) = \sum_x p_0(x)\, \pi(u \mid x)$ is the frozen-policy probability of visiting u, and $\pi(u \mid x) = 0$ unless $u \subseteq x$. Substituting into (51),

$$\rho^{\text{state}}_{k-1}(u) \;=\; \frac{C_{k-1}(u)}{\bar{C}^{\text{state}}_{k-1}(u)} \;\approx\; \frac{\sum_{x:\, u\subseteq x} \hat{p}_{k-1}(x)\, \pi(u \mid x)}{\sum_{x:\, u\subseteq x} p_0(x)\, \pi(u \mid x)}, \tag{53}$$

where $\hat{p}_{k-1}$ is the empirical terminal distribution after $k-1$ samples. On BAG, the state-level capacity ratio is therefore not a measure of trajectory-level overuse on the current trajectory; it is a $\pi(u \mid \cdot)$-weighted empirical-to-frozen terminal-popularity ratio over bags containing u as a sub-multiset.

Why the degenerate ratio is harmful. The intended design of CAS is to suppress intermediate resources that the pool has overused relative to the frozen policy, freeing budget for unexplored regions. Equation (53) shows that on BAG this rule fires whenever the pool has over-sampled *any* terminal containing u. Two consequences follow.

First, productive sub-multisets become over-penalized as soon as a few high-reward bags are sampled. A sub-multiset like $u = \{a, a\}$ typically appears in many high-reward bags (e.g. $\{a, a, b, c, d\}$, $\{a, a, b, d, e\}$, $\{a, a, c, d, e\}$, …). After even a handful of these have been sampled, $\hat{p}_{k-1}(\{x : \{a, a\} \subseteq x\}) > p_0(\{x : \{a, a\} \subseteq x\})$, so $\rho^{\text{state}}_{k-1}(\{a, a\}) > 1$ and the BPR penalty in (15) suppresses every action leading into $\{a, a\}$.

Second, this suppression blocks discovery of the *other* high-reward bags that share u but have not yet been sampled. The intermediate multisets that are most useful to revisit (because they lie on the trajectory to many productive modes) are precisely the ones that the degenerate $\rho^{\text{state}}_{k-1}$ over-penalizes. The history term thus inverts its intended effect: it discourages the productive flow that the frozen policy has learned, rather than discouraging trajectory-level redundancy.

Trans-CAS is subject to the same degeneracy because $C_{k-1}(s, a)$ on BAG aggregates over all order-permutations of any prefix that contains the relevant tokens, again collapsing trajectory-level information.

Why soft threshold is unaffected. Soft threshold is a function of $P_F(a \mid s)$ alone:

$$\tilde{\rho}_{k-1}(s, a) \;=\; \max\!\left(\rho_{k-1}(s, a),\; \frac{p_{\min}}{P_F(a \mid s)}\right). \tag{54}$$

The first argument inherits the degeneracy above, but the second argument depends only on the frozen forward policy's local logits at s. On BAG, actions that destroy a productive multiset pattern typically have small $P_F(a|s)$ under a well-trained frozen policy (since the policy concentrates on high-reward continuations), so the soft threshold downweights them; productive actions have $P_F(a|s) \geq p_{min}$ and are left untouched. Because soft threshold never reasons about the pool history, it cannot be misled by terminal popularity, and on BAG this is precisely the property that matters.

When CAS regains its intended behavior. On the other four benchmarks (TFBIND8, TFBIND10, QM9STR, SEHSTR), the state space is sequence-structured or graph-structured rather than multiset-structured, so different action histories leading to the same terminal pass through *distinct* intermediate states. The visit count $C_{k-1}(u)$ on these tasks reflects how often the pool has actually traversed the specific trajectory through u, not how popular the terminals containing u are. The capacity ratio therefore retains its intended trajectory-level interpretation, and the history term provides additional gain over soft threshold (Table 2: QM9STR from 184.0 to 252.8, SEHSTR from 107.4 to 131.8).

Implication for the scope of CAS. This analysis localizes the failure mode to a structural property of the state representation: CAS requires that visit counts on intermediate states reflect trajectory-level traversal frequency, not terminal-marginal mass. Sequential and graph-structured tasks satisfy this; permutation-invariant constructions like BAG do not.

C NOTATION AND TERMINOLOGY

Table 8 lists the symbols and terms used in the paper. We follow the GFlowNet framework of Bengio et al. (2023) where applicable.

Table 8: Notation table.

Symbol	Meaning
Graph and states	
$\mathcal{G} = (\mathcal{S}, \mathcal{A})$	Pointed DAG with state set $\mathcal{S}$ and edge set $\mathcal{A}$.
s_0	Source (initial) state.
s_f	Sink state; absorbs all completed trajectories.
$\mathcal{S}^f = \mathrm{Par}(s_f)$	Set of terminating states (parents of s_f).
$x(\tau)$	Terminating state visited by trajectory τ, $x(\tau) \in \mathcal{S}^f$.
$\mathrm{Child}(s)$, $\mathrm{Par}(s)$	Children and parents of s in $\mathcal{G}$.
Actions and transitions	
$\mathcal{A}(s)$	Valid actions at state $s \neq s_f$.
$T(s, a)$	Deterministic transition function; $T(s, a) \in \mathrm{Child}(s)$.
$a \in \mathcal{A}(s) \;\leftrightarrow\; s \to T(s, a) \in \mathcal{A}$	Bijection between actions and outgoing edges in deterministic environments.
Trajectories	
$\tau = (s_0, s_1, \ldots, s_n, s_{n+1} = s_f)$	Complete trajectory; $s_n \in \mathcal{S}^f$.
τ_k in Algorithm 1	Executed (s, a, s') trace of the k-th draw; carries action labels for count updates.
Policies and induced distributions	
$P_F(a \mid s)$	Frozen forward policy over actions; equals $P_F(T(s, a) \mid s)$ in edge notation.
$p_0(\tau) = \prod_{t=0}^{n} P_F(s_{t+1} \mid s_t)$	Frozen trajectory distribution.
$p_0(x) = \sum_{\tau:\, x \to s_f \in \tau} p_0(\tau)$	Terminal marginal over $\mathcal{S}^f$.
$R : \mathcal{S}^f \to \mathbb{R}_+$	Reward function on terminating states. Asymptotic target $p_0(x) \propto R(x)$.
Modes (paper-specific)	
$\mathcal{M}$	Set of modes (a partition or labeling of $\mathcal{S}^f$ used for evaluation).
$M(\tau) \in \mathcal{M}$	Mode label of $x(\tau)$.
$p_0^{\mathcal{M}}$	Frozen marginal over modes induced by p_0.
w_m	Weight of mode m in the coverage objective F_N.
Finite-pool deployment	
N	Sample budget.
$\tau_1, \ldots, \tau_N$	Drawn trajectories.
$\mathcal{P}_N = \{x(\tau_1), \ldots, x(\tau_N)\}$	Terminal pool (multiset).
$h_{k-1} = (\tau_1, \ldots, \tau_{k-1})$	History before draw k.
Counts and capacities	
$C_{k-1}(s)$	Number of past trajectories visiting s.
$C_{k-1}(s, a)$	Number of past trajectories using transition (s, a).
$\tilde{C}^{\mathrm{trans}}_{k-1}(s, a) = C_{k-1}(s) P_F(a \mid s)$	Frozen-policy capacity of transition (s, a).
$\tilde{C}^{\mathrm{state}}_{k-1}(u) = \sum_{(v,b):T(v,b)=u} C_{k-1}(v) P_F(b \mid v)$	Frozen-policy capacity of child state u.
$\rho^{\mathrm{trans}}_{k-1}$, $\rho^{\mathrm{state}}_{k-1}$	Volume-to-capacity ratios; see equation 12, equation 13.
$\widetilde{\rho}_{k-1}(s, a)$	Effective ratio with capacity-resolution floor; equation 14.
CAS *penalty*	
$\kappa_{k-1}(s, a)$, $\kappa^{\mathrm{cap}}_{k-1}(s, a)$	Local penalty in the KL-regularized objective.
λ	Penalty strength.
t	Free-flow threshold (penalty activates above t).
β	Growth exponent above threshold.
$p_{\min}$	Capacity-resolution floor on $P_F(a \mid s)$.
Resource-reuse identity (Proposition 1)	
$j \in \mathcal{J}$	Resource (state, transition, prefix).
$B_{j,\tau} \in \{0, 1\}$	Indicator that τ uses resource j.
$z_{p_0}(j) = \mathbb{P}_{\tau \sim p_0}(B_{j,\tau} = 1)$	Frozen-policy probability of using j.
$C_N(j) = \sum_i B_{j,\tau_i}$	Pool-level usage count of j.