THRESHOLD-GUIDED COMPOSITION OF GFLOWNETS FOR MULTI-OBJECTIVE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating candidates that satisfy multiple property requirements is a central challenge in multi-objective generation. Optimizing an aggregate reward can improve some properties while leaving others below their required levels. We introduce μGFN, which composes independently trained GFlowNets through threshold-guided sampling. Each objective model provides a flow ratio, μ, that estimates the expected terminal reward under its own policy. The sampler uses these estimates to penalize objective-wise threshold shortfalls, with no penalty once an estimate meets its threshold. This focuses guidance on predicted deficits while allowing the same models to serve different objective subsets and thresholds without retraining. Hypergrid diagnostics illustrate how objective-specific thresholds redirect sampling. Molecular experiments on three tasks with two to four objectives show competitive performance on the two-objective task and higher mean yields of distinct feasible molecules than preference-conditioned baselines on the three-objective task. On the four-objective task, μGFN achieves the highest mean objective-space coverage and top-candidate quality in the main comparison, although jointly feasible molecules remain rare. Ablations show that guidance improves mean unique success over unguided flow composition.

1 INTRODUCTION

Generative design often involves several acceptance criteria for the same candidate. A molecule, for example, may need adequate binding affinity, drug-likeness (Bickerton et al., 2012), and synthesizability (Ertl & Schuffenhauer, 2009). Each property can have a minimum acceptable level, and acceptance requires all the specified criteria to hold. Improving an already satisfactory property does not compensate for missing another requirement. The resulting task is to generate multiple distinct candidates within the region where these conditions hold simultaneously.

GFlowNets (Bengio et al., 2021; 2023) offer a way to generate diverse candidates by sampling proportionally to a terminal reward. Multi-objective variants such as MOGFN-PC (Jain et al., 2023b) and HN-GFN (Zhu et al., 2023) condition generation on preferences over objectives, enabling exploration of different trade-offs. Goal-conditioned GFlowNets also learn to target specified regions in objective space (Roy et al., 2023). These approaches establish control over trade-offs and target regions. We consider a complementary setting in which objective models are trained separately and requested lower cutoffs enter only the sampling rule.

Changing objective subsets creates a further reason to reuse trained models. Compositional Sculpting (Garipov et al., 2023) combines pretrained generative distributions through operators such as harmonic means and contrast, using classifier guidance to coordinate generation. This establishes model composition as a route to reuse. We study how objective-wise reward estimates can guide composed GFlowNets toward requested lower cutoffs, with no additional classifier training when the selected objectives or thresholds change.

We introduce μGFN, which guides the composition of independently trained objective models toward requested requirements. Each model supplies a flow ratio $\widehat{\mu}_k$ estimating its expected terminal reward. The sampler combines normalized flows with penalties for predicted threshold shortfalls, with no penalty once an estimate meets its threshold. The same models serve different objective subsets, thresholds, and guidance strengths without retraining. These estimates concern separate objective policies and support soft guidance without certifying joint feasibility.

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

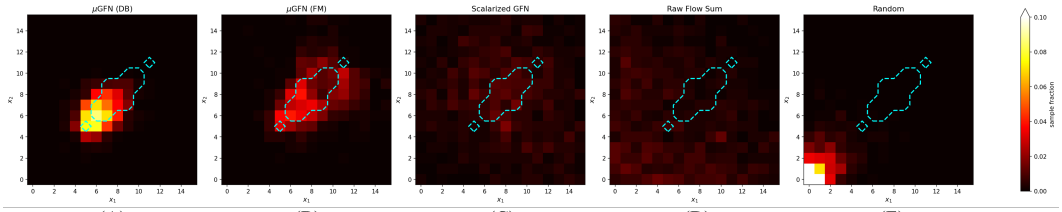


Figure 1: **Per-objective guidance focuses sampling near the jointly feasible region.** Sampling distributions on a 16×16 hypergrid for (A) μGFN (DB), (B) μGFN (FM), (C) Scalarized GFN, (D) Raw Flow Sum, and (E) Random. Panels (A) and (B) apply the linear grid gate (Appendix E.8) to estimates from DB- and FM-trained models. Scalarized GFN learns the product reward $R_1 R_2$, while Raw Flow Sum adds separately trained objective flows at sampling time (Section 3.1). All five panels share the same color scale, indicating the fraction of generated samples at each grid state. Cyan dashed outlines mark states satisfying both objective thresholds. Appendix Figure 5 provides a complementary view with separate color scales to make each distribution's spatial structure easier to inspect.

Our central contribution is a sampling rule that uses objective-wise reward estimates to respond to requested requirements while reusing independently trained models. We relate the composition rule and flow-based reward estimates to existing formulations (Appendix B and Section 2.3). We characterize how normalization and threshold saturation shape the sampler, and evaluate their effects on multi-objective generation.

Contributions.

1. **Threshold-guided composition.** We use objective-wise log-shortfall penalties to guide a normalized flow composition toward requested lower cutoffs. Above-threshold estimates receive no further bonus within the penalty, focusing guidance on predicted deficits. This enables flexible control over objective subsets and cutoffs at inference time without retraining the underlying objective models (Section 2.1).
2. **Interpretable objective-wise guidance.** We construct the signal μ from coupled reward heads and specify the shared-backward-policy conditions under which their ratio equals the expected terminal reward under each objective's own policy. This supplies reward-scaled estimates for threshold comparisons and identifies the scope of their interpretation (Sections 2.2–2.3).
3. **Design analysis and empirical evaluation.** We analyze the normalized base's invariance to global flow scaling, threshold cancellation in uncapped guidance, and the capped rule's local threshold response. Hypergrid diagnostics (Figure 1) and molecular experiments evaluate how guidance affects generation. Molecular ablations separate normalization, flow-greedy selection, and saturation, showing that the benefit of capping depends on the task (Section 3).

2 METHOD

Problem setup and notation. A GFlowNet (Bengio et al., 2021; 2023) constructs objects on a finite directed acyclic graph (DAG), starting from s_0. At state s, a legal action $a \in \mathcal{A}(s)$ gives $s' = T(s, a)$; a stop action yields a completed object $x \in \mathcal{X}$. For positive terminal reward $R(x)$, its target distribution is $P_T(x) = R(x)/Z$, where $Z = \sum_{x' \in \mathcal{X}} R(x')$. Positive edge flows $F(s, a)$ define the state flow $F(s) = \sum_{a \in \mathcal{A}(s)} F(s, a)$ and forward policy $P_F(a \mid s) = F(s, a)/F(s)$. A backward policy P_B assigns probabilities to legal incoming transitions. Appendix F reviews the standard flow matching (FM), detailed balance (DB), and trajectory balance (TB) objectives.

We consider K evaluation scores $c_k(x)$ and seek objects satisfying $c_k(x) \geq \tau_k$ for every selected objective. Each objective has a positive training reward $R_k(x) = g_k(c_k(x))$, where g_k is its fixed score-to-reward transformation. Evaluation uses cutoffs τ_k, while sampling is guided by requested

score cutoffs u_k, converted to reward units as $\theta_k = g_k(u_k)$. These roles are distinct even when $u_k = \tau_k$, as in our main molecular comparison.

μGFN first trains one flow model per objective, then holds the models fixed while composing them during sampling. Each step queries the selected models and penalizes their predicted shortfalls from requested thresholds. We present this sampler first, followed by its training construction and expected-reward interpretation.

2.1 Threshold-Guided Composition

Inputs. For each selected objective k, a trained model supplies positive edge flows $F_k^{(1)}(s,a)$ and $F_k^{(2)}(s,a)$, with ideal terminal targets R_k and R_k^2, respectively, on a common construction space. Write $F_k^{(n)}(s) = \sum_{a\in\mathcal{A}(s)} F_k^{(n)}(s,a)$ and use the ratio $\widehat{\mu}_k(s,a) = F_k^{(2)}(s,a)/F_k^{(1)}(s,a)$ as the guidance signal in training-reward units. Sections 2.2–2.3 specify how the models are trained and when this ratio represents an expected terminal reward. The sampling rule below operates on the learned signals.

Base composition. For each selected objective, compute $Z_k = F_k^{(1)}(s_0)$ once per trajectory and keep it fixed. The molecular implementation sums all root outputs, including stop, before applying the minimum-block stop mask. Root normalization removes global flow-scale differences. The unguided score and action distribution are

$$B(s,a) = \sum_{k=1}^{K} \frac{F_k^{(1)}(s,a)}{Z_k}, \qquad P_0(a \mid s) = \frac{B(s,a)}{\sum_{c\in\mathcal{A}(s)} B(s,c)}. \tag{1}$$

Under consistent flows, $F_k^{(1)}(s,a)/Z_k = d_k(s)P_{F,k}^{(1)}(a \mid s)$ is a globally normalized edge mass, where $d_k(s)$ is the reaching probability under the unmasked model policy. This identity need not hold for the policy renormalized after minimum-block stop masking (Appendix E.3). Appendix B discusses the connection to existing composition methods.

The base allocates action probability using the objective flows; $\widehat{\mu}_k$ additionally estimates the continuation reward under objective k's own policy. We use these models as separate local evaluators, penalizing actions whose estimates fall below the requested levels. Whether these penalties improve joint satisfaction under the composed sampler is assessed empirically in the molecular experiments.

Reward-based guidance. With the estimate $\widehat{\mu}_k$ and threshold $\theta_k = g_k(u_k) > 0$ in the same training-reward units, define the log-scale shortfall penalty

$$h_k(s,a) = \left[\log\theta_k - \log\widehat{\mu}_k(s,a)\right]_+, \qquad [z]_+ = \max(0,z). \tag{2}$$

With guidance strength $\lambda \geq 0$, the sampling rule is

$$\ell(s,a) = \log B(s,a) - \lambda \sum_{k=1}^{K} h_k(s,a),$$
$$P_\mu(a \mid s) = \frac{\exp \ell(s,a)}{\sum_{c\in\mathcal{A}(s)} \exp \ell(s,c)}. \tag{3}$$

The log shortfall measures a relative deficit. Multiplying an objective's signal and threshold by the same positive constant leaves its penalty unchanged. An objective above its threshold contributes zero penalty; it contributes no bonus to cancel another objective's deficit. The base score can still favor an action with a deficit, so this is soft guidance. We compute $\log\widehat{\mu}_k = \log F_k^{(2)} - \log F_k^{(1)}$ and normalize logits over legal actions; Appendix E.3 specifies numerical clipping and molecular stop masks. The backward head is required during training but need not be evaluated for sampling.

Setting $\lambda = 0$ yields **Root-Normalized Flow Sum** (μGFN w/o guidance), with identical checkpoints and root-flow normalization. Changing θ_k, λ, or the selected subset of compatible objective models requires no retraining. An unseen objective requires training its own model. Algorithm 1 summarizes inference.

Algorithm 1 μGFN guided composition

Require: Selected dual-head models; reward-space cutoffs $\theta_k = g_k(u_k) > 0$; $\lambda \geq 0$

Compute unmasked root totals $Z_k \leftarrow \sum_a F_k^{(1)}(s_0, a)$ for each k
$s \leftarrow s_0$
while s is nonterminal **do**
  for $k = 1, \ldots, K$ **do**
    Evaluate $F_k^{(1)}(s, a), F_k^{(2)}(s, a)$ for all legal a
    $m_k(s, a) \leftarrow \log F_k^{(2)}(s, a) - \log F_k^{(1)}(s, a)$
    $h_k(s, a) \leftarrow [\log \theta_k - m_k(s, a)]_+$
  end for
  $\ell(s, a) \leftarrow \log\big(\sum_k F_k^{(1)}(s, a)/Z_k\big) - \lambda \sum_k h_k(s, a)$
  Sample $a \sim \mathrm{softmax}_{a \in \mathcal{A}(s)}\, \ell(s, a)$
  $s \leftarrow T(s, a)$
end while
return Completed object s

Local optimization interpretation. For fixed estimates and thresholds at a state, Equation (3) minimizes λ times expected total predicted shortfall plus a KL penalty relative to P_0. Increasing λ weakly reduces that expected shortfall, without guaranteeing higher terminal success (Appendix A.4).

At a fixed state, varying a cutoff within a range that keeps all action estimates on the same side leaves the policy unchanged (Appendix A.3).

2.2 Coupled Flow Models with Detailed Balance

All objective models share the construction space defined above, allowing any selected subset to evaluate the same partial object and legal actions. In molecular generation, an action attaches a fragment to an available stem or terminates construction (Bengio et al., 2021).

Each objective model has a shared backbone for its two flow heads and one backward-policy head. The terminal targets introduced in Section 2.1 use the same transformed reward $R_k = g_k(c_k)$, with the second head targeting its square. The map g_k includes normalization, a reward exponent, and a score floor. The molecular implementation adds a small log-stabilization constant to each target; Appendix E.1 gives the correction to the ideal expectation identity. The backward policy is shared within each objective, while models for different objectives are trained independently.

Consistent path allocation. Flow conservation alone does not fix how reward mass is allocated across multiple construction paths. Two heads can therefore satisfy their individual reward constraints while allocating mass differently, even if they share a backbone. Their ratio then need not represent an expected reward. We use detailed balance (DB) (Bengio et al., 2023) to tie both heads to the same backward transition probabilities, imposing a common path allocation.

Let $\mathcal{I}(s') = \{(s, a) : T(s, a) = s'\}$ denote the legal incoming transitions of s'. Backward logits $b_k(s, a; s')$ define

$$P_{B,k}(s, a \mid s') = \frac{\exp b_k(s, a; s')}{\sum_{(u,c) \in \mathcal{I}(s')} \exp b_k(u, c; s')}. \tag{4}$$

The same normalized values are used by both reward heads. Normalization is over incoming transitions, retaining distinct actions when multiple transitions share a parent state. For each sampled child s', we enumerate all its legal incoming transitions and apply the DB constraints to both heads. We denote the resulting shared-backward-policy training variant μGFN DBPB.

Define the boundary-extended child flow $\overline{F}_k^{(n)}$, using prescribed rewards at terminal states and model state flows otherwise, as

$$\overline{F}_k^{(n)}(s') = \begin{cases} R_k(x)^n, & s' = x \text{ is terminal,} \\ F_k^{(n)}(s'), & s' \text{ is nonterminal.} \end{cases} \tag{5}$$

For $n \in \{1, 2\}$, DB requires $F_k^{(n)}(s, a) = \overline{F}_k^{(n)}(s')P_{B,k}(s, a \mid s')$. We minimize the following squared log-residuals over sampled children and their incoming transitions

$$d_{k,n}(s, a, s') = \log F_k^{(n)}(s, a) - \log \overline{F}_k^{(n)}(s') - \log P_{B,k}(s, a \mid s'),$$
$$\mathcal{L}_k^{\mathrm{DB}} = \mathbb{E}_{s' \sim \mathcal{D}_k}\left[\sum_{(s,a)\in\mathcal{I}(s')} \sum_{n=1}^{2} d_{k,n}(s, a, s')^2\right], \tag{6}$$

where $\mathcal{D}_k$ is the training distribution of sampled child states. The flow heads and the shared backward head are optimized jointly.

Sharing $P_{B,k}$ is the structural condition used below; learning it allows the common allocation to adapt during training. A fixed uniform backward policy also satisfies the sharing condition when DB is enforced exactly. Thus, learning $P_{B,k}$ is a modeling choice, rather than a prerequisite for the identity, and DB is our mechanism for enforcing consistency rather than the only possible training objective.

2.3 Expected-Reward Interpretation of the Guidance Signal

Flow-based expectation estimation also appears in conditional self-flow constructions with terminal rewards R and R^2 (**?**, Appendix D). The following statement supplies the policy-specific interpretation of the edge-flow ratio used by our sampler.

Proposition 1 (Expected reward from coupled flows)**.** *Suppose both objective-k heads satisfy exact DB on all legal transitions with the same backward policy $P_{B,k}$ and terminal targets R_k and R_k^2. Let $P_{F,k}^{(1)}(a \mid s) = F_k^{(1)}(s, a)/F_k^{(1)}(s)$ be the first head's forward policy. For an action leading to $s' = T(s, a)$,*

$$\mu_k(s, a) := \frac{F_k^{(2)}(s, a)}{F_k^{(1)}(s, a)} = \frac{\overline{F}_k^{(2)}(s')}{\overline{F}_k^{(1)}(s')} = \mathbb{E}_{P_{F,k}^{(1)}}[R_k(X) \mid s, a]. \tag{7}$$

In particular, $\mu_k(s, a) = R_k(x)$ for a terminal transition.

The second head carries an extra factor of terminal reward under the same path allocation. Dividing by the first head averages that reward under its continuation policy. Appendix A.1 gives the derivation. For these ideal targets ($c = 0$), the ratio lies between the minimum and maximum reachable terminal rewards. The ratio estimates $\mathbb{E}[g_k(c_k(X)) \mid s, a]$; a nonlinear reward transformation generally prevents interpreting it as the expected evaluation score.

Finite training leaves nonzero DB residuals, so the sampler uses the estimate $\widehat{\mu}_k$. Each estimate concerns continuation under its own objective model; it is not the expected reward under the composed sampler. These signals therefore guide action selection without certifying terminal feasibility. Appendix A.2 discusses their interpretation and limits.

Relation to action values. With terminal-only rewards and unit discount, Equation (7) is the action value $Q_k^{P_{F,k}^{(1)}}(s, a)$ for reward R_k. It averages terminal rewards under the first-head continuation policy and does not estimate the maximum reachable reward. The coupled heads provide objective-wise signals for the threshold penalties in Section 2.1.

3 Experiments

The experiments examine threshold response and joint requirement satisfaction in two settings. The enumerable hypergrid makes terminal probability allocation directly inspectable through a linear guidance diagnostic. Its role is to characterize responses to objective-wise cutoffs. The molecular comparisons and ablations evaluate the full log-shortfall sampler in Equation (3) and the contribution of its components.

3.1 HYPERGRID GUIDANCE DIAGNOSTICS

Setup. Starting at $(0, 0)$ on a 16×16 hypergrid, actions increment either coordinate within the grid or stop at the current state. Two conflicting Gaussian rewards peak at $(11, 4)$ and $(4, 12)$ (Appendix C). Scores equal rewards here, and a terminal state is feasible when $R_k(x) \geq \tau_k$ for both objectives. The grid gate uses linear scores $\sum_k \widehat{\mu}_k(s, a) - \lambda \sum_k [\theta_k - \widehat{\mu}_k(s, a)]_+$ with $\lambda = 10$ and softmax temperature $T = 0.5$. It uses DB- or FM-derived estimates; DB fixes a uniform backward policy shared by both heads. This diagnostic omits the molecular base-flow term and uses linear rather than log shortfalls. Appendix E.8 specifies its rules, and Appendix D compares the flow-ratio estimates.

Baselines. *Scalarized GFN* trains a single GFlowNet on the product reward $R_{\text{sc}}(x) = R_1(x)R_2(x)$ and samples its learned forward policy. *Raw Flow Sum* instead trains one model per objective and combines their first-head edge flows at inference as $P_{\text{raw}}(a \mid s) \propto \sum_k F_k^{(1)}(s, a)$, normalized over legal actions. It uses neither root normalization by Z_k nor threshold guidance, so objectives with larger flow scales can dominate the sum. The former combines terminal rewards during training; the latter combines learned flows during sampling. Random selects uniformly among legal actions. None of these baselines uses the feasibility cutoffs to guide sampling.

Sampled success rates. SR is the fraction of samples satisfying both cutoffs. Table 1 reports it when guidance and evaluation thresholds coincide. Both guided variants have higher mean SR than the baselines at all three thresholds. FM has the higher mean at $\theta = 0.3$, whereas DB has the higher mean at 0.4 and 0.5. At $\theta = 0.5$, their means of 0.286 and 0.263 exceed Scalarized GFN's 0.032.

Table 1: Sampled Success Rate on the 16×16 hypergrid (mean $\pm$ SD over 3 seeds; $1{,}000$ samples per method and seed at each threshold). The DB and FM rows evaluate the linear guidance diagnostic using μGFN estimates. Bold indicates the highest mean at each threshold.

Method	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.5$
μGFN (DB)	0.636 ± 0.007	$\mathbf{0.536 \pm 0.009}$	$\mathbf{0.286 \pm 0.026}$
μGFN (FM)	$\mathbf{0.662 \pm 0.010}$	0.415 ± 0.038	0.263 ± 0.052
Scalarized GFN	0.317 ± 0.024	0.140 ± 0.009	0.032 ± 0.005
Raw Flow Sum	0.146 ± 0.009	0.050 ± 0.003	0.008 ± 0.002
Random	0.017 ± 0.004	0.002 ± 0.001	0.000 ± 0.000

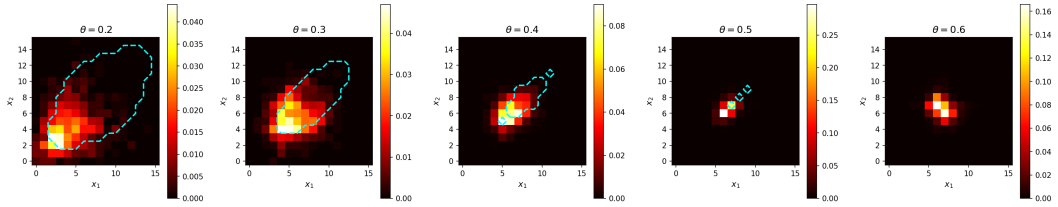


Figure 2: Sampling distributions of the linear grid gate with DB-derived $\widehat{\mu}$ at thresholds $\theta \in \{0.2, 0.3, 0.4, 0.5, 0.6\}$. Cyan contours outline states satisfying both thresholds; this feasible region shrinks with θ and is empty at $\theta = 0.6$. Colors indicate the fraction of samples at each grid state, with a separate color scale for each panel.

Shared and objective-specific thresholds. Figure 2 shows samples becoming more localized as both thresholds increase from 0.2 to 0.5. At 0.6, no terminal state is feasible. A separate forward dynamic programming (DP) experiment holds each DB checkpoint fixed and varies the two guidance cutoffs independently (Appendix E.8). At evaluation cutoffs $(0.5, 0.3)$, matched guidance gives SR 0.6083, compared with 0.0740 under the opposite request $(0.3, 0.5)$; reversing the evaluation cutoffs gives 0.6076 versus 0.0745. These comparisons support objective-specific control in the linear diagnostic. They do not test the molecular log-shortfall rule. The FM shared-threshold sweep is retained in Appendix E.6.

Table 2: Main molecular results (mean ± standard deviation over 5 seeds; 5,000 samples per seed). μGFN uses DBPB training with $\lambda = 8$ for SEH + QED + SA and $\lambda = 1$ for the other two tasks. Guidance and evaluation use the same objective-specific score cutoffs in every task, with 0.5 for binding objectives, 0.6 for QED, and 0.7 for SA. Reward-space guidance thresholds are $\theta_k = g_k(\tau_k)$. $\mathrm{SR}_{\mathrm{unique}}$ counts unique feasible molecules divided by all generated samples. HV uses the origin as reference; Geom measures top-100 unique candidate quality. Bold indicates the highest mean for each task and metric, including ties at the displayed precision.

Method	$\mathrm{SR}_{\mathrm{unique}}\uparrow$	HV↑	Geom↑
JNK3 + GSK3β (2 objectives)			
μGFN DBPB	$\mathbf{0.0055 \pm 0.0009}$	0.6707 ± 0.0057	$\mathbf{0.5274 \pm 0.0057}$
MOGFN-PC	0.0050 ± 0.0019	$\mathbf{0.6747 \pm 0.0394}$	0.5182 ± 0.0175
HN-GFN	0.0008 ± 0.0008	0.5141 ± 0.1031	0.4759 ± 0.0131
Raw Flow Sum	0.0006 ± 0.0003	0.5053 ± 0.1190	0.4266 ± 0.0170
Random	0	0.2075 ± 0.0684	0.2315 ± 0.0045
SEH + QED + SA (3 objectives)			
μGFN DBPB	$\mathbf{0.0655 \pm 0.0150}$	0.6534 ± 0.0113	$\mathbf{0.7535 \pm 0.0143}$
MOGFN-PC	0.0594 ± 0.0080	$\mathbf{0.6613 \pm 0.0145}$	0.7514 ± 0.0106
HN-GFN	0.0222 ± 0.0036	0.6267 ± 0.0219	0.7007 ± 0.0043
Raw Flow Sum	0.0176 ± 0.0011	0.6314 ± 0.0064	0.6938 ± 0.0024
Random	0.0035 ± 0.0005	0.5814 ± 0.0251	0.6452 ± 0.0034
JNK3 + GSK3β + QED + SA (4 objectives)			
μGFN DBPB	$\mathbf{0.0006 \pm 0.0001}$	$\mathbf{0.4252 \pm 0.0247}$	$\mathbf{0.5878 \pm 0.0025}$
MOGFN-PC	0.0001 ± 0.0001	0.3077 ± 0.0918	0.4902 ± 0.0342
HN-GFN	0.0001 ± 0.0001	0.3254 ± 0.0345	0.5232 ± 0.0085
Raw Flow Sum	0.0000 ± 0.0001	0.2534 ± 0.0404	0.4582 ± 0.0069
Random	0	0.1393 ± 0.0428	0.3560 ± 0.0067

3.2 Molecular Optimization

Setup. We evaluate μGFN using the fragment-based molecular generation MDP of Bengio et al. (2021), with pre-trained predictors providing objective scores. Table 2 compares our shared-backward-policy detailed-balance model (μGFN DBPB; Section 2) against the baselines on the following three objective combinations.

- **JNK3 + GSK3β** uses the two-target inhibition benchmark (Zhu et al., 2023).
- **SEH + QED + SA** combines a binding-related objective with drug-likeness (QED; Bickerton et al. 2012) and synthesizability (SA; Ertl & Schuffenhauer 2009).
- **JNK3 + GSK3β + QED + SA** extends the two-target benchmark with QED and SA, giving four objectives.

Baselines. We compare against MOGFN-PC (Jain et al., 2023b), a preference-conditioned GFlowNet with weighted-sum scalarization; HN-GFN (Zhu et al., 2023), a hypernetwork-based preference-conditioned GFlowNet; Raw Flow Sum, using the unnormalized first-head flow composition defined in Section 3.1; and Random, which selects actions uniformly among legal actions.

Training budget. One objective evaluation scores one sampled training item for one property. All trained methods in Table 2 receive 160,000 such evaluations per objective and seed, totaling $160{,}000K$ for K objectives, including all separately trained models. This matches oracle access; Appendix Table 4 details the accounting.

Evaluation protocol. The main comparison uses 5 seeds and 5,000 generated molecules per method and seed, reporting means and standard deviations. A valid molecule satisfies the evaluation constraints if $c_k(x) \geq \tau_k$ for every selected objective. Requested guidance score cutoffs u_k match evaluation cutoffs τ_k for every task, with $u_k = \tau_k = 0.5$ for JNK3, GSK3β, and SEH, 0.6 for QED, and 0.7 for SA. μGFN DBPB uses task-specific strengths fixed before the experiments, with $\lambda = 8$ for SEH + QED + SA and $\lambda = 1$ for the other two tasks. Each guidance cutoff is converted separately to reward units as $\theta_k = g_k(\tau_k)$ (Appendix E.1).

Metrics. Our main measure of requirement satisfaction is unique success rate ($\mathrm{SR}_{\mathrm{unique}}$), the number of distinct valid molecules satisfying every cutoff divided by the total number of generated samples. This measures feasible yield; structural diversity within the feasible subset requires separate assessment. Hypervolume (HV) and Top-100 geometric mean score (Geom) provide complementary measures of objective-space coverage and top-candidate quality. Both are computed after deduplication and include candidates that do not meet every cutoff. HV uses raw oracle scores in $[0, 1]$ and reference point $(0, \ldots, 0)$. Geom is the arithmetic mean of the highest 100 per-molecule geometric mean scores after SMILES deduplication, using all unique molecules if fewer than 100 are available (Appendix E.1).

Main results. On JNK3 + GSK3β, DBPB has higher mean $\mathrm{SR}_{\mathrm{unique}}$ (0.0055 versus 0.0050) and Geom (0.5274 versus 0.5182) than MOGFN-PC, which has the higher mean HV (0.6747 versus 0.6707). Thus, the two-objective comparison does not show a uniform advantage across metrics. On SEH + QED + SA, DBPB has the highest mean $\mathrm{SR}_{\mathrm{unique}}$ (0.0655) and Geom (0.7535) among the methods in Table 2, compared with 0.0594 and 0.7514 for MOGFN-PC. MOGFN-PC again has the higher mean HV (0.6613 versus 0.6534). These are comparisons of reported means, not claims of statistical significance. Appendix E.2 compares DBPB with the FM variant.

On JNK3 + GSK3β + QED + SA, DBPB's mean $\mathrm{SR}_{\mathrm{unique}}$ is 0.0006, corresponding to approximately three unique feasible molecules per $5{,}000$ samples. It also achieves the highest mean HV (0.4252) and Geom (0.5878) in Table 2. These auxiliary metrics show improved objective-space coverage and top-candidate quality despite the low feasible yield. Joint satisfaction remains difficult on this four-objective task.

3.3 Design Properties and Sampler Ablations

We examine how the sampler handles unequal flow scales, objective-wise reward estimates, and requested thresholds. The properties below follow from the sampling rules; Table 3 evaluates their effect on generated molecules using the same DBPB checkpoints within each task.

Removing global flow-scale dependence. Raw Flow Sum weights objectives through their unnormalized first-head flows. Multiplying one objective's flows by a positive constant can change this mixture. Root normalization cancels that constant through Z_k, leaving its contribution to B unchanged. This invariance concerns the base composition and does not equalize objective influence at every state. Raw Flow Sum and Root-Normalized Flow Sum isolate this normalization step.

Using objective-wise reward estimates. Neither P_0 nor greedy selection by B checks predicted rewards against requirements. The flow-greedy arm selects $\arg\max_a B(s, a)$ with probability 0.25 and samples from P_0 otherwise. Comparing these arms with μ-guided sampling assesses the benefit of using continuation-reward estimates beyond flow allocation.

Making thresholds affect the policy. The uncapped arm uses the same $\widehat{\mu}_k$, B, and λ as the full method. With $z_k = \log \widehat{\mu}_k - \log \theta_k$, their logits are

$$\ell_{\mathrm{uncap}} = \log B + \lambda \sum_k z_k, \qquad \ell_{\mathrm{cap}} = \log B + \lambda \sum_k \min(0, z_k).$$

Within the capped penalty, surplus on one objective cannot cancel another's deficit. The base still affects action selection. As shown in Section 2.1, the capped policy minimizes expected predicted shortfall with KL regularization toward P_0 at each state, without guaranteeing terminal feasibility. Appendix E.9 details these properties; Appendix E.3 specifies the controls.

Observed effects on feasible yield. The capped sampler achieves higher mean unique success than normalization alone and the tested flow-greedy sampler on all three tasks. Compared with uncapped guidance, it achieves higher mean unique success on the two- and three-objective tasks, including 0.0655 versus 0.0381 on SEH + QED + SA. On the four-objective task, uncapped guidance achieves higher mean unique success (0.0011 versus 0.0006), HV, and Geom. By construction, capping enables cutoff-dependent action selection when action estimates straddle a requested cutoff, while assigning zero penalty to each objective whose estimate meets or exceeds its cutoff. In uncapped guidance, the cutoff terms cancel when normalizing action probabilities. The observed benefit of saturation is therefore task-dependent at the reported settings; threshold responsiveness alone does not ensure higher feasible yield.

Table 3: DBPB sampler ablation (mean ± SD over 5 seeds; 5,000 molecules per seed and arm). Metrics and task-specific guidance/evaluation cutoffs follow Table 2. All arms within a task reuse the same checkpoints with $T = 1$. The displayed λ applies to the capped and uncapped arms. Bold marks the highest mean within each task and metric, including displayed ties. Rounded zero means do not imply zero successes.

Configuration	$\mathrm{SR}_{\mathrm{unique}}$ ↑	HV↑	Geom↑
JNK3 + GSK3β ($\lambda = 1$)			
Raw Flow Sum	0.0006 ± 0.0003	0.5053 ± 0.1190	0.4266 ± 0.0170
Root-Normalized Flow Sum	0.0012 ± 0.0004	0.6018 ± 0.0850	0.4522 ± 0.0113
Greedy flow ($p = 0.25$)	0.0026 ± 0.0010	0.6290 ± 0.0482	0.4697 ± 0.0177
Uncapped μ bonus	0.0050 ± 0.0012	0.6667 ± 0.0222	0.5246 ± 0.0087
Capped penalty (μGFN)	$\mathbf{0.0055 \pm 0.0009}$	$\mathbf{0.6707 \pm 0.0057}$	$\mathbf{0.5274 \pm 0.0057}$
SEH + QED + SA ($\lambda = 8$)			
Raw Flow Sum	0.0176 ± 0.0011	0.6314 ± 0.0064	0.6938 ± 0.0024
Root-Normalized Flow Sum	0.0245 ± 0.0014	0.6508 ± 0.0063	0.7128 ± 0.0042
Greedy flow ($p = 0.25$)	0.0252 ± 0.0034	0.6324 ± 0.0104	0.7029 ± 0.0091
Uncapped μ bonus	0.0381 ± 0.0077	0.6463 ± 0.0272	0.7194 ± 0.0063
Capped penalty (μGFN)	$\mathbf{0.0655 \pm 0.0150}$	$\mathbf{0.6534 \pm 0.0113}$	$\mathbf{0.7535 \pm 0.0143}$
JNK3 + GSK3β + QED + SA ($\lambda = 1$)			
Raw Flow Sum	0.0000 ± 0.0001	0.2534 ± 0.0404	0.4582 ± 0.0069
Root-Normalized Flow Sum	0.0002 ± 0.0002	0.3507 ± 0.0603	0.5297 ± 0.0059
Greedy flow ($p = 0.25$)	0.0000 ± 0.0000	0.4371 ± 0.0294	0.5383 ± 0.0136
Uncapped μ bonus	$\mathbf{0.0011 \pm 0.0003}$	$\mathbf{0.4400 \pm 0.0258}$	$\mathbf{0.5932 \pm 0.0027}$
Capped penalty (μGFN)	0.0006 ± 0.0001	0.4252 ± 0.0247	0.5878 ± 0.0025

4 Conclusion

We presented μGFN, a framework for multi-objective generation that composes independently trained GFlowNets and guides construction toward objective-specific requirements. Its sampler combines root-normalized flows and penalizes each objective's estimated shortfall from a requested threshold. Each penalty is zero when its estimate meets the threshold, focusing guidance on predicted deficits. Coupled flow heads supply these expected-reward estimates under exact balance. For a library of trained objectives, the same models can serve different objective subsets, cutoffs, and guidance strengths without retraining. This combines model reuse with inference-time control over the requirements used to guide generation.

We evaluated the approach through hypergrid diagnostics and fragment-based molecular generation. On the grid, the linear guidance rule achieved higher mean sampled success than the tested baselines and responded to asymmetric objective cutoffs in exact forward-DP evaluations. Molecular experiments covered three tasks with two to four objectives under matched training oracle budgets. μGFN was competitive on the two-objective task, improved mean unique success over preference-conditioned baselines on the three-objective task, and attained the highest mean hypervolume and Top-100 quality in the main four-objective comparison. The sampler analysis separates invariance of the normalized base to flow scaling from the threshold response enabled by capping. Molecular ablations support the practical value of guidance beyond normalization and the tested flow-greedy rule. Capping increased mean unique success on the two- and three-objective tasks, while uncapped guidance had higher means on all three metrics in the four-objective task.

These findings support threshold-guided composition while leaving joint feasibility challenging. The signals describe expected rewards under separate objective policies, so guidance does not guarantee simultaneous satisfaction; the four-objective task still yielded only about three unique feasible molecules per 5,000 samples. The task comparisons do not isolate scaling with objective count. New objectives require training; storage grows with the model library and inference cost with the selected objectives. Future work should improve calibration along composed trajectories, extend independent-cutoff tests to the molecular sampler, and evaluate feasible-subset diversity and computational cost on broader benchmarks.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, and assist with translation. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, assist in the writing of proofs, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, or interpret results, and generating synthetic data sets and cleaning and reformatting datasets are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, source and search for information, and to edit the research paper to improve readability.

We have reviewed all AI-assisted work. All statements, assumptions, and proof strategies were devised by the authors; generative AI was used only to help write up proofs whose steps the authors had already established, and every resulting step was checked line by line by the authors. All AI-generated or AI-edited code was reviewed line by line and tested for correctness by the authors, and every reported number was reproduced from the authors' own experimental runs. AI-assisted translations and readability edits were compared against the authors' original drafts to confirm that no technical claim was altered. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our anonymous code repository is available at `https://anonymous.4open.science/r/muGFN-23D2/`.

REFERENCES

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

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

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

Miruna Cretu, Charles Harris, Ilia Igashov, Arne Schneuing, Marwin Segler, Bruno Correia, Julien Roy, Emmanuel Bengio, and Pietro Liò. Synflownet: Design of diverse and novel molecules with synthesis constraints. In *International Conference on Learning Representations*, volume 2025, pp. 47101–47126, 2025.

Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence*, pp. 518–528. PMLR, 2022.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

Timur Garipov, Sebastiaan De Peuter, Ge Yang, Vikas Garg, Samuel Kaski, and Tommi Jaakkola. Compositional sculpting of iterative generative processes. *Advances in neural information processing systems*, 36:12665–12702, 2023.

Moksh Jain, Tristan Deleu, Jason Hartford, Cheng-Hao Liu, Alex Hernandez-Garcia, and Yoshua Bengio. Gflownets for ai-driven scientific discovery. *Digital Discovery*, 2(3):557–577, 2023a.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-Garcıa, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective gflownets. In *International conference on machine learning*, pp. 14631–14653. PMLR, 2023b.

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.

Elaine Lau, Stephen Z Lu, Ling Pan, Doina Precup, and Emmanuel Bengio. Qgfn: Controllable greediness with action values. *Advances in neural information processing systems*, 37:81645–81676, 2024.

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.

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.

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.

Julien Roy, Pierre-Luc Bacon, Christopher Pal, and Emmanuel Bengio. Goal-conditioned gflownets for controllable multi-objective molecular design. *arXiv preprint arXiv:2306.04620*, 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.

Seokwon Yoon, Youngbin Choi, Seunghyuk Cho, Seungbeom Lee, MoonJeong Park, and Dongwoo Kim. Routing by reaching: Composition of pre-trained gflownets for multi-objective generation. *arXiv preprint arXiv:2602.21565*, 2026.

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

A Supporting Analysis for Threshold-Guided Composition

A.1 Proof of Proposition 1

Fix an objective k and the common backward policy $P_{B,k}$. Let $w_k(s, x)$ be the probability that a backward trajectory starting at terminal x visits state s. On the finite DAG, recursively expanding the DB equations from the terminal boundary gives, at any nonterminal state s,

$$F_k^{(n)}(s) = \sum_{x \in \mathcal{X}} R_k(x)^n w_k(s, x), \qquad n \in \{1, 2\}. \tag{8}$$

Both heads have the same weights because they share $P_{B,k}$. A path contributes the product of its normalized backward transition probabilities; summing these products gives $w_k(s, x)$. Raw trajectory counts do not in general give this allocation.

For the first head, the probability of a forward continuation path from s to x telescopes to $R_k(x)/F_k^{(1)}(s)$ times the corresponding product of backward probabilities. Summing over continuation paths yields

$$P_{F,k}^{(1)}(X = x \mid s) = \frac{R_k(x) w_k(s, x)}{F_k^{(1)}(s)}.$$

Consequently,

$$\frac{F_k^{(2)}(s)}{F_k^{(1)}(s)} = \frac{\sum_x R_k(x)^2 w_k(s, x)}{\sum_x R_k(x) w_k(s, x)} = \mathbb{E}_{P_{F,k}^{(1)}}[R_k(X) \mid s]. \tag{9}$$

For an action (s, a) leading to s', DB gives $F_k^{(n)}(s, a) = \overline{F}_k^{(n)}(s')P_{B,k}(s, a \mid s')$. The common backward factor cancels in the ratio of the two heads. If s' is nonterminal, the remaining ratio is the expected reward from s' by Equation (9); if it is terminal, the ratio is $R_k(x)^2/R_k(x) = R_k(x)$. The deterministic transition therefore gives Equation (7).

This derivation concerns the ideal terminal targets R_k and R_k^2. Appendix E.1 states the correction for the molecular implementation's additive stabilizer. Flow conservation without common path allocation does not suffice to identify this expectation. Two independently allocated flows can satisfy their individual terminal-reward constraints while producing a ratio with different semantics.

A.2 Interpretation and Limits of the Guidance Signals

Reward bounds and ordering. Under Proposition 1 with ideal targets ($c = 0$), $\mu_k(s, a)$ averages reachable terminal rewards under the first-head continuation policy and lies between their minimum and maximum. If every reward reachable through a is at least every reward reachable through a', then $\mu_k(s, a) \geq \mu_k(s, a')$. These statements assume exact flows. With stabilized targets $R_k + c$ and $R_k^2 + c$, the exact ratio instead averages $\rho_c(R_k)$ from Equation (20), so its bounds are the minimum and maximum reachable transformed values, which may lie outside the original reward interval. A learned ratio outside the applicable exact interval indicates inconsistency; a value within it does not establish calibration. Appendix E.1 bounds the stabilizer's effect under the reported reward settings.

Terminal agreement. For ideal targets ($c = 0$), the exact action value equals $R_k(x)$ on a terminal transition; stabilized targets instead give $\rho_c(R_k(x))$. At earlier states it can rise or fall along a sampled trajectory; increasing construction depth alone does not guarantee progressively more accurate learned estimates. The relevant continuation policy is objective-specific and generally differs from the composed sampler.

Thresholds and feasibility. An expectation above θ_k can assign positive probability to terminal rewards below θ_k. For ideal targets ($c = 0$), an exact ratio above θ_k on the final terminal action implies a terminal reward above θ_k, but our soft sampler does not impose this test as a hard constraint. With stabilization, the terminal ratio is $\rho_c(R_k(x))$ and this implication need not hold for the original reward cutoff. Neither an intermediate threshold check nor the use of an estimated ratio establishes joint terminal feasibility. Success rates therefore measure the effect of the guidance empirically.

Objective-specific expectations and joint feasibility. The limitation can persist even with exact flows. Consider an action leading to a state with two terminal children whose reward vectors are $(1, \varepsilon)$ and $(\varepsilon, 1)$, with $\varepsilon = 0.01$. Each objective's first-head policy samples these children in proportion to its own reward. Consequently, both exact action values are $(1 + \varepsilon^2)/(1 + \varepsilon) \approx 0.9902$. Thresholds $\theta_1 = \theta_2 = 0.5$ produce zero shortfall on this action, although neither terminal is jointly feasible. Each estimate averages a different continuation distribution; the estimates do not encode their joint success event. This example shows why objective-wise expected rewards alone cannot certify joint feasibility.

A.3 Guided Sampling Details

Limiting behavior. When all $\widehat{\mu}_k(s, a) \geq \theta_k$, the penalty vanishes and the rule reduces to the root-normalized composition P_0 in Equation (1). The same reduction holds for $\lambda = 0$.

Threshold response. For a fixed objective, changing θ_k affects relative action probabilities only when it separates some action estimates from others. If all actions lie above it, that objective contributes no penalty; if all lie below it, its threshold is an action-independent logit offset. In particular, when all actions fall below every threshold, the policy is proportional to $B(s, a) \prod_k \widehat{\mu}_k(s, a)^\lambda$ and is unchanged by further threshold increases within that regime.

Policy ratios and reward scale. Normalizing each flow head into an action policy removes its state-flow scale. Taking the ratio of these policies yields

$$\frac{P_F^{(2)}(a|s)}{P_F^{(1)}(a|s)} = \frac{F^{(2)}(s, a)/\sum_{a'} F^{(2)}(s, a')}{F^{(1)}(s, a)/\sum_{a'} F^{(1)}(s, a')} = \frac{\mu_k(s, a)}{\bar{\mu}_k(s)}, \tag{10}$$

where $\bar{\mu}_k(s) = \sum_{a'} F^{(2)}(s, a') / \sum_{a'} F^{(1)}(s, a')$ is a positive state-level constant. The policy ratio preserves the ordering of μ_k across actions at a given state, but recovering its absolute reward scale also requires $\bar{\mu}_k(s)$. This is a limitation of using policy ratios alone, not a requirement to use FM loss. Our DB construction retains reward-scaled flows and couples their path allocations through the shared backward policy (Section 2.2).

Linear-scale interpretation. In linear scale, the combined sampling rule reads

$$P_\mu(a \mid s) \;\propto\; \underbrace{B(s,a)}_{\text{root-normalized flow mixture}} \prod_{k=1}^{K} \min\left\{1, \left(\frac{\widehat{\mu}_k(s,a)}{\theta_k}\right)^{\lambda}\right\}, \tag{11}$$

with normalization over the allowed actions. The base mixture uses globally normalized edge flows. Each objective's damping factor is 1 above its threshold and, for $\lambda > 0$, below 1 when its expected-reward estimate falls short.

A.4 Local Variational Interpretation

Fix a state s, its legal action set, the learned estimates, and the thresholds. Let $H_s(a) = \sum_k [\log \theta_k - \log \widehat{\mu}_k(s,a)]_+$ and assume $P_0(a \mid s) > 0$ on this set. Writing P_λ for Equation (3) at strength λ, define

$$C_\lambda(s) = \sum_a P_0(a \mid s) e^{-\lambda H_s(a)}, \qquad P_\lambda(a \mid s) = \frac{P_0(a \mid s) e^{-\lambda H_s(a)}}{C_\lambda(s)}. \tag{12}$$

For any action distribution π on this set,

$$D_{\mathrm{KL}}(\pi \| P_0) + \lambda \mathbb{E}_{a\sim\pi}[H_s(a)] = D_{\mathrm{KL}}(\pi \| P_\lambda) - \log C_\lambda(s). \tag{13}$$

Nonnegativity of KL divergence establishes that P_λ uniquely minimizes the left-hand side. Thus the sampling rule trades off predicted shortfall against departure from the base distribution. Its dependence on λ also satisfies

$$\frac{\partial}{\partial \lambda} \mathbb{E}_{a\sim P_\lambda}[H_s(a)] = -\operatorname{Var}_{a\sim P_\lambda}[H_s(a)] \le 0. \tag{14}$$

To obtain this identity, differentiate the normalized weights to get $\partial_\lambda P_\lambda(a \mid s) = P_\lambda(a \mid s)(\mathbb{E}_{P_\lambda}[H_s] - H_s(a))$, then sum against $H_s(a)$.

This is a local variational characterization of the existing rule. It applies to the fixed predicted shortfall at one state; changing the sampler also changes visited states and subsequent actions. Consequently, it does not establish monotonic improvement in expected terminal reward, joint success rate, or diversity as λ increases.

B Related Work

GFlowNets. GFlowNets (Bengio et al., 2021; 2023) learn to sample compositional objects proportionally to a reward function. Extensions include trajectory balance (Malkin et al., 2022), subtrajectory balance (Madan et al., 2023), detailed balance (Deleu et al., 2022), local search (Kim et al., 2024), and improved training techniques (Shen et al., 2023; Pan et al., 2023). GFlowNets have been applied broadly to scientific discovery (Jain et al., 2023a), including synthesis-aware molecular generation (Cretu et al., 2025). Our primary focus is inference-time guidance. Its signal construction couples two reward heads through a shared backward policy and DB constraints. The construction supplies the consistency condition for the flow-ratio interpretation.

Multi-objective GFlowNets. MOGFN-PC (Jain et al., 2023b) conditions a GFlowNet on preference vectors ω through weighted-sum scalarization $R_\omega(x) = \sum_k \omega_k R_k(x)$. HN-GFN (Zhu et al., 2023) generates GFlowNet parameters from ω through a hypernetwork and integrates preference search with Bayesian optimization. These methods use preference-conditioned models to explore trade-offs across the Pareto front. For a given preference vector, a scalarized flow $F_\omega(s,a)$ supplies an aggregate quantity rather than the vector of objective-wise expected rewards used by our sampler. μGFN composes independently trained objective models and applies separate threshold-shortfall penalties during construction. Appendix E.5 discusses fixed scalarization in more detail.

Action-value guidance. QGFN (Lau et al., 2024) learns action values to adjust GFlowNet sampling. Its Appendix B also evaluates applying a trained Q-function to independently trained GFlowNets and a `p-thresh` variant that masks actions with $Q(s, a) < p$. The paper also discusses property-constraint guidance as a future application. Thus inference-time guidance and absolute value thresholds have prior precedents. μGFN combines objective-wise flow-ratio estimates through soft relative-shortfall penalties on a normalized composition. Our flow-greedy ablation tests a sampling rule with fixed flow scores; it is not a reproduction or evaluation of QGFN.

Goal-conditioned generation. Roy et al. (2023) train GFlowNets conditioned on a focus region defined by cosine similarity to a requested direction in objective space. Their goal-dependent reward encourages generation within that region. Our requested conditions are per-objective lower cutoffs, applied during sampling to separately trained objective models. This distinguishes the conditioning geometry and training arrangement while recognizing prior work on direct control of objective-space regions.

Composition of GFlowNets. Garipov et al. (2023) introduce Compositional Sculpting, defining operations such as harmonic means and contrast over pretrained generative distributions and realizing these compositions through classifier guidance. Their framework establishes that guidance can coordinate intermediate decisions to compose GFlowNet distributions. Our setting emphasizes requested objective-wise lower cutoffs. Coupled flow heads provide continuation-reward estimates, and the sampler penalizes their shortfalls from those cutoffs. The same objective models can be recombined without training an additional composition classifier. These local penalties provide soft threshold guidance and do not establish exact sampling from the distribution restricted to jointly feasible objects.

A related formulation, Routing by Reaching (Yoon et al., 2026), combines pretrained objective policies using reaching probabilities. Under consistent flows and before additional action masking, our unguided base P_0 is algebraically equivalent to its linear composition with reward weights proportional to $1/Z_k$. This correspondence concerns the normalized base. Our contribution centers on the objective-wise threshold-shortfall rule used to guide sampling from that base.

C Hypergrid Reward Landscape

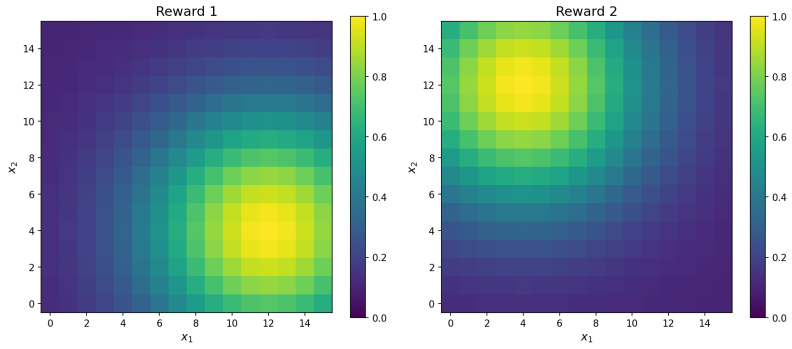


Figure 3: Hypergrid reward landscape. R_1 (left) peaks at the lower-right; R_2 (right) peaks at the upper-left. The two objectives conflict spatially, with high-R_1 regions having low R_2 and vice versa. A narrow strip of simultaneously-elevated reward lies along the antidiagonal.

D Guidance Signal Diagnostics for FM and DB

Exact reference under a fixed backward policy. We compare FM- and DB-trained flow ratios on the 16×16 hypergrid in Figure 4. In this diagnostic, the DB model uses a fixed uniform backward policy, shared by its two reward heads. The exact reference uses the same backward policy. These results describe a single run (seed 1) and are separate from the three-seed SR comparison in Table 1.

For objective k, let $w^{\text{unif}}(s, x)$ be the probability that a backward trajectory from terminal x visits s, with each backward transition chosen uniformly among its legal incoming transitions. The exact

flows and the induced first-head conditional terminal distribution are

$$F^{(n)}_{k,\star}(s) = \sum_{x \in \mathcal{X}} R_k(x)^n w^{\mathrm{unif}}(s,x), \qquad n \in \{1,2\}, \tag{15}$$

$$P^{(1)}_{F,k,\star}(x \mid s) = \frac{R_k(x) w^{\mathrm{unif}}(s,x)}{F^{(1)}_{k,\star}(s)}. \tag{16}$$

Consequently, the reference is

$$\mu_{k,\star}(s) = \frac{\sum_x R_k(x)^2 w^{\mathrm{unif}}(s,x)}{\sum_x R_k(x) w^{\mathrm{unif}}(s,x)} = \mathbb{E}_{P^{(1)}_{F,k,\star}}[R_k(X) \mid s]. \tag{17}$$

Here the terminal weights in the reward average are proportional to $R_k(x)w^{\mathrm{unif}}(s,x)$, while w^{unif} itself is a backward visitation probability. Forward dynamic programming from each state under the policy induced by the exact first-head flows reproduces this reference, with terminal probability mass summing to 1.0000 at the reported precision. For Objective 1, both calculations give 0.3378 at $(0,1)$ and 0.8807 at $(12,2)$.

Why a reachable-set reference differs. A reachable-set alternative is

$$\mu_{k,\mathrm{set}}(s) = \frac{\sum_{x \in \mathcal{X}(s)} R_k(x)^2}{\sum_{x \in \mathcal{X}(s)} R_k(x)}, \tag{18}$$

where $\mathcal{X}(s)$ contains the terminals reachable from s. This is a reward-weighted average, not a uniform average of rewards. It replaces the backward visitation weights by one on every reachable terminal, discarding how mass is routed through the construction graph. On the right/up grid, multiple paths connect states to terminals, but their contribution is the sum of products of normalized backward transition probabilities, rather than the raw number of paths. At s_0, every backward trajectory visits the root, so the two references agree (0.5991 for Objective 1). Away from the root they can differ substantially. At $(0,1)$, the set-based value is 0.6012 whereas the conditional expectation is 0.3378; at $(12,2)$ the values are 0.7349 and 0.8807. Figure 4 uses the exact conditional reference in Equation (17).

Calibration results and interpretation. Against this exact reference, DB has MAEs of 0.0011 and 0.0002 for Objectives 1 and 2, respectively, with both Pearson correlations rounding to 1.0000. FM has MAEs of 0.1180 and 0.1377, with correlations of 0.8645 and 0.7826. The DB estimate closely reproduces the spatial structure of its exact target, whereas the FM estimate assigns elevated values over broader regions. Thus, the localized high-value regions in the DB plot reflect the conditional expectation under the specified path allocation; their shape alone is not evidence of risk aversion or a conservative lower bound.

This experiment supports accurate recovery by DB of its target under a fixed uniform backward policy in the displayed run. FM conservation alone does not impose this backward allocation or require its two heads to share an allocation. Its discrepancy from this reference can therefore reflect both estimation error and differences in flow allocation, rather than solely inaccurate value estimation under its own learned forward policy. The forward-DP check above verifies the reference using the *exact-flow* policy; it does not separately measure either model's error against continuation values recomputed under its *learned* policy. This state-level diagnostic also does not establish a terminal-feasibility guarantee for the composed multi-objective sampler.

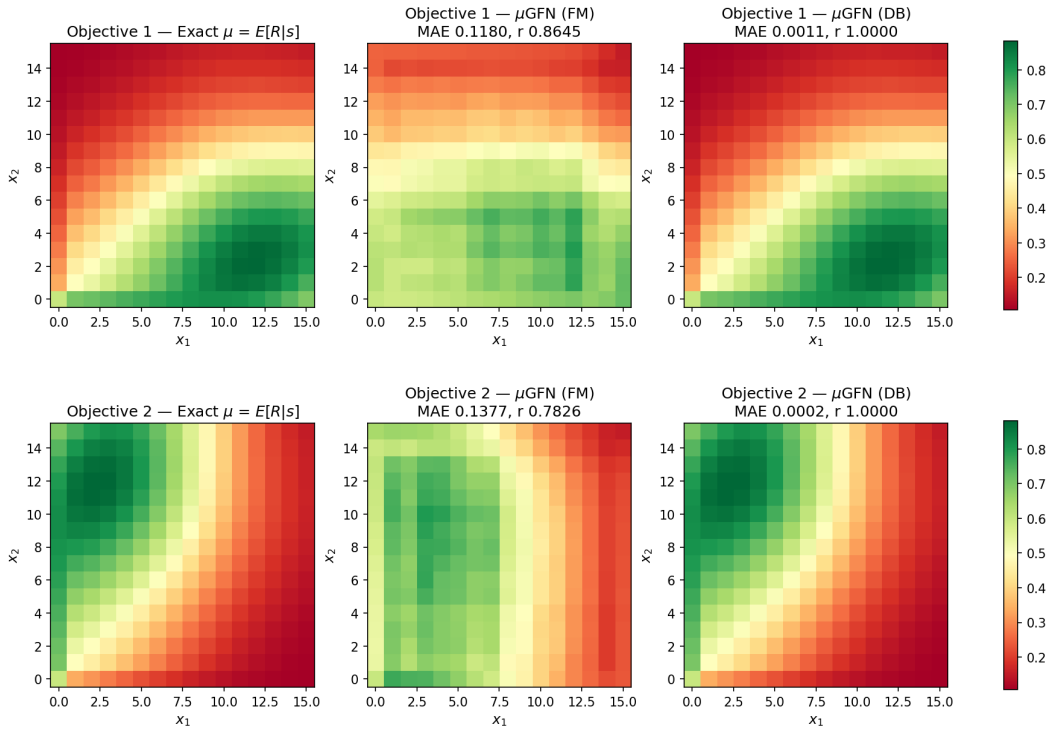


Figure 4: **Flow-ratio comparison with an exact conditional-reward reference.** Rows correspond to Objectives 1 and 2; columns show the exact expectation under the first-head policy induced by uniform-backward exact flows, the FM estimate, and the DB estimate. The DB model is trained with this same fixed uniform backward policy shared across its two heads. Each objective row uses a common color scale across all three columns. Panel annotations report mean absolute error (MAE) and Pearson correlation r against the exact reference for this seed-1 run. Displayed correlations of 1.0000 are rounded and do not imply zero error.

E Experimental Details

This appendix documents molecular training budgets, training and sampler comparisons, threshold sensitivity, and hypergrid sampling diagnostics. Appendix E.8 specifies the linear hypergrid diagnostic and its exact cross-evaluation.

E.1 Training Budgets and Hyperparameters

Top-100 geometric mean score. For each run, duplicate entries are removed by SMILES before computing Geom. Let n be the number of remaining molecules and $c_k(x_i)$ the score of molecule x_i on objective k. The per-molecule geometric mean is

$$q_i = \exp\left(\frac{1}{K}\sum_{k=1}^{K}\log\max\{c_k(x_i), 10^{-8}\}\right).$$

The floor prevents taking the logarithm of zero and is separate from the training-reward floor. Sort these values as $q_{(1)} \geq \cdots \geq q_{(n)}$. For $n > 0$, the reported run-level metric is

$$\text{Geom} = \frac{1}{m}\sum_{i=1}^{m} q_{(i)}, \qquad m = \min(100, n).$$

Thus, the outer aggregation is an arithmetic mean of per-molecule geometric means. When fewer than 100 unique molecules are available, every remaining molecule contributes. Deduplication prevents repeated copies of a high-scoring molecule from occupying multiple top-ranked positions. The metric is calculated separately for each seed before reporting its mean and standard deviation across seeds.

The geometric mean is sensitive to low individual objective scores, but increases in other scores can still compensate for a decrease in one objective. Geom applies no feasibility cutoff and can include molecules that violate the requirements. It therefore complements $\mathrm{SR}_{\mathrm{unique}}$ without certifying simultaneous satisfaction.

Score cutoffs and reward-space guidance. For the molecular sampler, let $\eta_k > 0$ and $\beta_k > 0$ denote the `reward_norm` and `reward_exp` for objective k, respectively. A requested guidance cutoff u_k in score units is converted before sampling as

$$\theta_k = g_k(u_k) = \left(\frac{\max(u_k, 10^{-2})}{\eta_k}\right)^{\beta_k}. \tag{19}$$

The reported molecular DBPB experiments use `reward_norm`$= 1.0$ and `reward_exp`$= 1.0$ for every objective, so $g_k(u) = \max(u, 10^{-2})$. Thus, for scores in $[0, 1]$, the training rewards lie in $[0.01, 1]$, and all reported guidance cutoffs are unchanged by the conversion. This is the same map used for the training reward. The general form above supports objective-specific normalization and exponent settings. Each cutoff is converted separately; a common transformation would change the intended score-space constraints when objectives use different normalization or exponent settings. Evaluation compares scores $c_k(x)$ directly with cutoffs τ_k. The main comparison uses $u_k = \tau_k$ as specified in Section 3.2, while Appendix E.4 varies guidance with evaluation held fixed. Guidance and evaluation cutoffs can differ in general, and aligning them does not guarantee terminal feasibility.

Stabilized terminal targets. The molecular MuDBPB implementation uses $\log(c+R_k)$ and $\log(c+R_k^2)$ for the two terminal DB targets, with `log_reg_c` set to $c = (0.1/8)^4 = 2.44140625 \times 10^{-8}$. The second target is $c + R_k^2$, not $(c + R_k)^2$. Under exact balance with a shared backward policy, the stabilized flow ratio is the conditional expectation of

$$\rho_c(r) = \frac{r^2 + c}{r + c} = r + \frac{c(1 - r)}{r + c}, \tag{20}$$

under the first head's continuation policy for terminal reward $R_k + c$. The ideal expected-R_k identity in Section 2.3 holds exactly for $c = 0$. This additive log stabilizer is distinct from the score floor 10^{-2} in Equation (19).

To bound the stabilizer's effect, fix the backward allocation at a state or edge and normalize its nonnegative terminal weights to $\sum_x w_x = 1$. Write $m = \sum_x w_x R_k(x)$ and let the ideal and stabilized ratios under exact balance be

$$\mu_0 = \frac{\sum_x w_x R_k(x)^2}{m}, \qquad \mu_c = \frac{\sum_x w_x R_k(x)^2 + c}{m + c}.$$

For $R_k(x) \in [r_{\min}, 1]$ with $r_{\min} = 0.01$, both m and μ_0 lie in this interval, giving

$$0 \le \mu_c - \mu_0 = \frac{c(1 - \mu_0)}{m + c} \le \frac{c(1 - r_{\min})}{r_{\min} + c} < 2.42 \times 10^{-6}.$$

Since $\mu_0 \ge r_{\min}$, the relative change is below 2.42×10^{-4}, or 0.0242%. These are uniform bounds over the stated reward range, not empirical estimates from sampled molecules. They isolate the additive stabilizer under a fixed shared backward allocation and exact balance. They do not bound errors in learned flows, changes in allocation during training, or the effect on terminal feasibility.

Oracle budget for the main comparison. For the trained methods in Table 2, we use a batch size of 16 and 10,000 training iterations per model and seed. Each independently trained objective model evaluates only its own objective, whereas each preference-conditioned model evaluates all K objectives on every sampled training item. Table 4 accounts for all K objective models in the composed methods. Both training strategies allocate 160,000 evaluations to each objective, for a total of $160{,}000K$ objective evaluations per task and seed.

For the two-, three-, and four-objective tasks, the matched training budgets are therefore 320,000, 480,000, and 640,000 objective evaluations per trained method and seed, respectively. The model count treats a preference-conditioned system as one jointly trained model. The second reward head of μGFN uses the square of the same observed reward, so constructing its target incurs no additional oracle evaluation. Random has no training phase. Matching objective-evaluation budgets controls oracle access; it does not imply equal training time or inference cost.

Table 4: Training oracle-evaluation budgets for the main molecular comparison, per task and seed with K objectives. Each trained model uses batch size 16 for 10,000 iterations. An objective evaluation scores one sampled training item for one objective. Budgets are computed from the training schedule and count sampled items, rather than distinct molecules. The final 5,000-molecule evaluation per seed is separate from this training budget.

Method	Models	Training samples (all models)	Objectives per sample	Total objective evaluations
μGFN DBPB	K	$160{,}000K$	1	$160{,}000K$
Raw Flow Sum	K	$160{,}000K$	1	$160{,}000K$
MOGFN-PC	1	160,000	K	$160{,}000K$
HN-GFN	1	160,000	K	$160{,}000K$
Random	0	0	—	0

Table 5: Model, training, and sampling settings for μGFN (FM and DB). FM-specific reward and loss settings are listed separately.

Hyperparameter	Value
Architecture	
Fragment vocabulary size	105
Min / max blocks	2 / 8
Message-passing steps	10
Embedding dimension	256
Training	
Training iterations	10,000
On-policy batch size	16
Learning rate	5×10^{-4}
Exploration rate (ϵ)	0.05
Score floor	0.01
Log-regularization c	2.4×10^{-8}
FM-specific reward and loss settings	
Reward exponent	1 (no sharpening)
Balanced loss	Yes
Leaf coefficient	10
Pruned coefficient	0.1
Guided Sampling	
Samples per setting	5,000

Model and training settings. The FM and DB variants share the model architecture and training settings summarized in Table 5. Their per-objective dual-head GFlowNets use the fragment-based MDP of Bengio et al. (2021) with the 105-fragment vocabulary (`blocks_105.json`). Molecules are constructed by sequentially attaching 2–8 fragments. The shared backbone is a graph neural network (`v4` architecture) with 10 message-passing steps and embedding dimension 256. Each model is trained for 10,000 iterations with a batch size of 16 on-policy trajectories, using Adam with learning rate 5×10^{-4}. An ϵ-greedy exploration rate of 0.05 is used during training.

The FM variant uses the loss in Equation (23), with balanced loss enabled, leaf coefficient 10, and pruned coefficient 0.1. The DB variant uses the coupled detailed-balance loss in Section 2.2. Both use the log-regularization constant $c = (0.1/8)^4 \approx 2.4 \times 10^{-8}$. The FM reference experiments use oracle scores as rewards without exponentiation, with a score floor of 0.01. The objective-specific score-to-reward map for the DB sampler is given in Equation (19).

At inference time, guided sampling generates 5,000 molecules per threshold setting. All experiments use a single NVIDIA RTX 3090 GPU per objective.

Table 5 summarizes the key hyperparameters.

E.2 FM AND DBPB TRAINING COMPARISON

Table 6 compares the FM and DBPB variants on all three molecular objective combinations. Both variants use the settings in Table 2, with $\lambda = 8$ for SEH + QED + SA and $\lambda = 1$ for the other two tasks. Guidance and evaluation cutoffs coincide in score space for all tasks, with $u_k = \tau_k = 0.6$ for QED, 0.7 for SA, and 0.5 for binding objectives. Each guidance cutoff is mapped to reward units as $\theta_k = g_k(\tau_k)$. Results are means and standard deviations over 5 seeds, with $5{,}000$ samples per seed. The comparison concerns the training variants with guidance enabled; it is separate from the guidance-component ablation in Table 3.

Table 6: FM–DBPB comparison with matching settings within each task. We use $\lambda = 8$ for SEH + QED + SA and $\lambda = 1$ for the other two tasks. Guidance and evaluation score cutoffs coincide for every objective, with 0.5 for binding objectives, 0.6 for QED, and 0.7 for SA. Values are mean $\pm$ standard deviation over 5 seeds, with $5{,}000$ samples per seed. Reward-space conversion and metrics match Table 2. Bold indicates the higher mean for each task and metric, including ties at the displayed precision.

Training variant	$\mathrm{SR}_{\mathrm{unique}}\uparrow$	HV↑	Geom↑
JNK3 + GSK3β (2 objectives)			
μGFN DBPB	$\mathbf{0.0055 \pm 0.0009}$	$\mathbf{0.6707 \pm 0.0057}$	$\mathbf{0.5274 \pm 0.0057}$
μGFN FM	0.004 ± 0.001	0.617 ± 0.063	0.495 ± 0.012
SEH + QED + SA (3 objectives)			
μGFN DBPB	$\mathbf{0.0655 \pm 0.0150}$	$\mathbf{0.6534 \pm 0.0113}$	$\mathbf{0.7535 \pm 0.0143}$
μGFN FM	0.0540 ± 0.0161	0.6489 ± 0.0186	0.7397 ± 0.0141
JNK3 + GSK3β + QED + SA (4 objectives)			
μGFN DBPB	0.0006 ± 0.0001	$\mathbf{0.4252 \pm 0.0247}$	$\mathbf{0.5878 \pm 0.0025}$
μGFN FM	$\mathbf{0.0007 \pm 0.0006}$	0.4035 ± 0.0512	0.5599 ± 0.0206

Task-dependent differences. DBPB has higher mean HV and Geom on all three tasks. On SEH + QED + SA, it also has higher mean $\mathrm{SR}_{\mathrm{unique}}$ (0.0655 versus 0.0540). On the four-objective task, FM has a slightly higher reported mean $\mathrm{SR}_{\mathrm{unique}}$ (0.0007 versus 0.0006), while DBPB has higher mean HV (0.4252 versus 0.4035) and Geom (0.5878 versus 0.5599). These comparisons are descriptive; rounded means and standard deviations alone do not establish statistical significance or equivalence.

Choice of DBPB as the primary model. The shared-backward-policy DB constraints explicitly couple the path allocations of the two reward heads within each objective, providing the structural condition used for the expected-reward interpretation in Section 2.3. Sharing a backbone or enforcing flow conservation separately for each head does not by itself impose this condition. DBPB is selected to make this coupling explicit, rather than on the premise that it must outperform FM on every task. The reported HV and Geom means favor DBPB on these tasks, while FM also produces useful guided samples and has a slightly higher mean unique success rate on the four-objective task.

E.3 DBPB SAMPLER ABLATIONS

For each task in Table 3, all reported arms reuse the same per-objective DBPB checkpoints, with 5 seeds and $5{,}000$ generated molecules per seed. Metrics follow Table 2, including $\mathrm{SR}_{\mathrm{unique}}$. Guidance and evaluation use identical score cutoffs of $(0.5, 0.5)$ for JNK3 + GSK3β, $(0.5, 0.6, 0.7)$ for SEH + QED + SA, and $(0.5, 0.5, 0.6, 0.7)$ for JNK3 + GSK3β + QED + SA. Capped and uncapped guidance use $\lambda = 1, 8, 1$, respectively, with objective-specific reward thresholds $\theta_k = g_k(\tau_k)$. Objective weights and the flow exponent are both 1, and temperature is $T = 1$. No arm adds training or changes the checkpoints. Table 3 reports all three metrics for every arm and task, and Appendix E.9 provides additional interpretation.

Raw and root-normalized flow sums. Raw Flow Sum uses logits $\ell_{\mathrm{raw}}(s,a) = \log \sum_k F_k^{(1)}(s,a)$. Root-Normalized Flow Sum, denoted `norm` or μGFN w/o guidance, uses

$$\ell_{\mathrm{norm}}(s,a) = \log \sum_k \frac{F_k^{(1)}(s,a)}{Z_k}, \qquad Z_k = \sum_a F_k^{(1)}(s_0,a). \tag{21}$$

For each trajectory, Z_k is computed from the root outputs, including stop, before applying the minimum-block stop mask, and is held fixed throughout construction. It is not recomputed at intermediate states. This removes each objective's global flow scale while retaining its state-dependent reaching mass; it does not normalize each model into a local action policy. The final categorical sampler normalizes the combined action scores. Computing the mixture as $\mathrm{logsumexp}_k(\log F_k^{(1)} - \log Z_k)$ is mathematically equivalent to the linear-space sum in Equation (21). Since all arms use $T = 1$, Raw Flow Sum versus `norm` changes only the root normalization.

Threshold penalty. The full method follows the same code path as `norm`, with logits $\ell_{\mathrm{norm}}(s,a) - \lambda \sum_k h_k(s,a)$ from Equation (2). The task-specific λ and transformed thresholds are specified above. Setting $\lambda = 0$ recovers `norm`; the computed shortfalls then have no effect on action selection. This comparison isolates the penalty's contribution while retaining root normalization. The molecular implementation computes the guidance readout as $\widehat{\mu}_k = \exp(\mathrm{clip}(\log F_k^{(2)} - \log F_k^{(1)}, -20, 20))$ before applying the log-shortfall penalty.

Uncapped reward bonus. Let $z_k(s,a) = \log \widehat{\mu}_k(s,a) - \log \theta_k$. The uncapped arm retains the same normalized base, clipped flow-ratio readout, masks, and λ, but replaces $\min(0, z_k)$ by z_k, giving

$$\ell_{\mathrm{uncap}}(s,a) = \ell_{\mathrm{norm}}(s,a) + \lambda \sum_k z_k(s,a), \qquad \ell_{\mathrm{cap}}(s,a) = \ell_{\mathrm{norm}}(s,a) + \lambda \sum_k \min(0, z_k(s,a)).$$

Their logit difference is $\lambda \sum_k [z_k(s,a)]_+$, so the comparison isolates the additional bonus for estimates above their requested thresholds at a fixed strength. Numerical clipping of the flow ratio is retained in both arms. Since $-\lambda \sum_k \log \theta_k$ is constant across actions, the uncapped policy is independent of the requested thresholds. Unlike the linear grid rule in Appendix E.8, this molecular control retains $\log B$ and uses log-reward bonuses. Changing λ is therefore not generally a temperature adjustment of the normalized base.

Greedy mixing. Let $P_0 = \mathrm{softmax}\, \ell_{\mathrm{norm}}$. The greedy arm samples from

$$P_{\mathrm{greedy}}(a \mid s) = (1-p) P_0(a \mid s) + p\, \mathbf{1}[a = a_g(s)], \qquad a_g(s) = \arg\max_a q_g(s,a). \tag{22}$$

For the reported flow-greedy arm, $q_{\mathrm{fl}} = \ell_{\mathrm{norm}}$ and $p = 0.25$. The implementation also supports $q_\mu(s,a) = \min_k \widehat{\mu}_k(s,a)$ in training-reward units, but Table 3 reports only the flow-greedy setting. Both choices use P_0 for the non-greedy branch; no guidance threshold or shortfall penalty is used. The raw minimum across objectives is distinct from threshold-relative shortfall aggregation. The p-greedy mixture is adapted from Lau et al. (2024), using the fixed flow-derived scores above with no additional training. The flow-greedy comparison changes both the signal and the sampling rule; the separate uncapped arm above isolates saturation while retaining the same $\widehat{\mu}_k$. The reported $p = 0.25$ is one operating point, not an evaluation of the full family of greedy samplers. Algorithm 2 specifies the normalized and greedy samplers.

Algorithm 2 Normalized and greedy sampler ablations (one trajectory)

Require: Fixed objective models; mode $m \in \{\text{norm}, \text{greedy}\}$; $p \in [0, 1]$; selector $g \in \{\text{f1}, \mu\}$; block limits $b_{\min}, b_{\max}$

$s \leftarrow s_0$
$Z_k \leftarrow \sum_a F_k^{(1)}(s_0, a)$ for each k, before the stop mask
for $t = 0, \ldots, b_{\max} - 1$ **do**
  Evaluate $F_k^{(1)}(s, a)$ and $\widehat{\mu}_k(s, a) = F_k^{(2)}(s, a)/F_k^{(1)}(s, a)$
  $\ell(a) \leftarrow \text{logsumexp}_k(\log F_k^{(1)}(s, a) - \log Z_k)$
  if $m = \text{greedy}$ **then**
    $q(a) \leftarrow \ell(a)$
    if $g = \mu$ **then**
      $q(a) \leftarrow \min_k \widehat{\mu}_k(s, a)$
    end if
  end if
  if $t < b_{\min}$ **then**
    $\ell(\text{stop}) \leftarrow -\infty$; also $q(\text{stop}) \leftarrow -\infty$ if greedy
  end if
  if $m = \text{greedy}$ and a Bernoulli(p) draw is 1 **then**
    $a \leftarrow \arg\max_a q(a)$
  else
    Sample $a \sim \text{Categorical}(\text{logits} = \ell)$
  end if
  $(s, \text{done}) \leftarrow \textsc{AddBlock}(s, a)$
  if done **then**
    break
  end if
end for
return Constructed object s

E.4 DBPB Threshold Sweep on JNK3 + GSK3β

Table 7 varies a common guidance score cutoff u from 0.1 to 0.9 for the DBPB models, converting it separately to $\theta_k = g_k(u)$. Evaluation remains fixed at score cutoffs $(0.5, 0.5)$, and $\lambda = 1$ throughout. The experiment uses 5 seeds and 5,000 samples per seed and setting. This varies the sampler's guidance cutoff independently of the evaluation criterion. At $u = 0.5$, its full-method results match Tables 2 and 3. This experiment is separate from the older FM sweep below.

Mean $\text{SR}_{\text{unique}}$ rises from 0.0028 at $u = 0.1$ to 0.0055 at $u = 0.5$, then varies between 0.0045 and 0.0052. Mean HV peaks at $u = 0.4$ (0.6768), whereas mean Geom peaks at $u = 0.5$ (0.5274). Thus, increasing the guidance cutoff does not monotonically improve performance under the fixed evaluation cutoffs. This shared-cutoff sweep measures sensitivity along one direction in threshold space. It does not test independent control of the two objectives or identify which visited states have active threshold boundaries.

Table 7: DBPB guidance-score sweep on JNK3 + GSK3β with $\lambda = 1$ and fixed evaluation cutoffs $(0.5, 0.5)$. Values are mean $\pm$ SD over 5 seeds, with 5,000 samples per seed and setting. u is converted to each objective's reward-space threshold. Metrics follow Table 2; bold marks the highest mean within the sweep and within the baseline block separately.

Method	u	$\mathrm{SR}_{\mathrm{unique}} \uparrow$	HV↑	Geom↑
μGFN DBPB	0.1	0.0028 ± 0.0008	0.6217 ± 0.0260	0.4867 ± 0.0099
μGFN DBPB	0.2	0.0042 ± 0.0006	0.6492 ± 0.0441	0.5112 ± 0.0074
μGFN DBPB	0.3	0.0051 ± 0.0007	0.6765 ± 0.0112	0.5237 ± 0.0057
μGFN DBPB	0.4	0.0052 ± 0.0007	$\mathbf{0.6768 \pm 0.0070}$	0.5253 ± 0.0047
μGFN DBPB	0.5	$\mathbf{0.0055 \pm 0.0009}$	0.6707 ± 0.0057	$\mathbf{0.5274 \pm 0.0057}$
μGFN DBPB	0.6	0.0045 ± 0.0005	0.6744 ± 0.0113	0.5231 ± 0.0043
μGFN DBPB	0.7	0.0046 ± 0.0008	0.6733 ± 0.0100	0.5221 ± 0.0081
μGFN DBPB	0.8	0.0050 ± 0.0009	0.6672 ± 0.0211	0.5257 ± 0.0064
μGFN DBPB	0.9	0.0052 ± 0.0013	0.6667 ± 0.0222	0.5251 ± 0.0080
MOGFN-PC	—	$\mathbf{0.0050 \pm 0.0019}$	$\mathbf{0.6747 \pm 0.0394}$	$\mathbf{0.5182 \pm 0.0175}$
HN-GFN	—	0.0008 ± 0.0008	0.5141 ± 0.1031	0.4759 ± 0.0131
Raw Flow Sum	—	0.0006 ± 0.0003	0.5053 ± 0.1190	0.4266 ± 0.0170
Random	—	0	0.2075 ± 0.0684	0.2315 ± 0.0045

E.5 Objective-Wise Threshold Information

Remark 1 (Objective-wise shortfalls)**.** *A scalarized flow or reward estimate generally does not identify which objective falls below a specified threshold. Our sampler retains a separate signal $\widehat{\mu}_k(s, a)$ for each objective and computes its shortfall relative to θ_k. Within the penalty, exceeding one threshold supplies no bonus that cancels another objective's deficit. The base mixture still affects action probabilities, so this construction does not establish dominance over scalarization or guarantee a feasible terminal outcome. Table 3 compares the full guidance rule with normalization alone and flow-greedy sampling across three tasks. Its same-signal uncapped control tests saturation on all three tasks. Capping raises mean unique success on the two- and three-objective tasks, whereas uncapped guidance has higher means on all three metrics in the four-objective task.*

E.6 Hypergrid Sampling Distributions

Figure 5 shows the same five hypergrid sampling distributions as Figure 1, with a separate color scale for each panel. This display makes the location, spread, and shape of each distribution easier to inspect, including low-density structure that is less visible on the common scale. The panel-specific color limits preserve the sample-fraction units shown by the colorbars. Color intensity should therefore be compared within each panel, while Figure 1 provides a common scale for comparisons across methods. Table 1 reports the aggregate SR comparison.

Figure 6 retains the FM-trained μGFN threshold sweep for comparison with the DB results in Figure 2. In both visualizations, increasing θ from 0.2 to 0.5 shifts samples toward the central overlap and produces a more localized distribution. At $\theta = 0.6$, the feasible region is empty. Each panel uses its own color scale, so colors alone do not support comparisons of absolute sample fractions across panels or training variants.

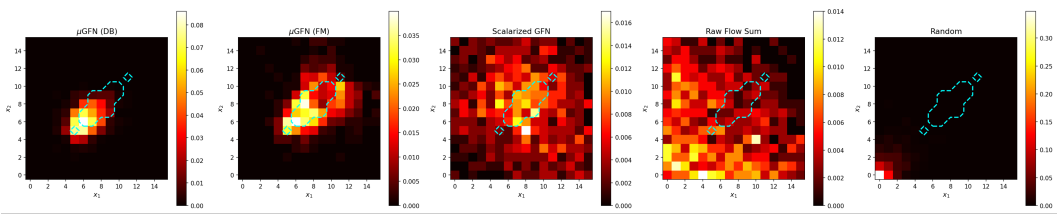


Figure 5: The same sampling distributions as in Figure 1, shown with separate color scales for (A) μGFN (DB), (B) μGFN (FM), (C) Scalarized GFN, (D) Raw Flow Sum, and (E) Random. Each panel uses its own color range to reveal the distribution's spatial structure. Colorbars retain sample-fraction units, but equal colors do not imply equal sample fractions across methods. Cyan outlines mark the jointly feasible region. Figure 1 provides the common-scale comparison.

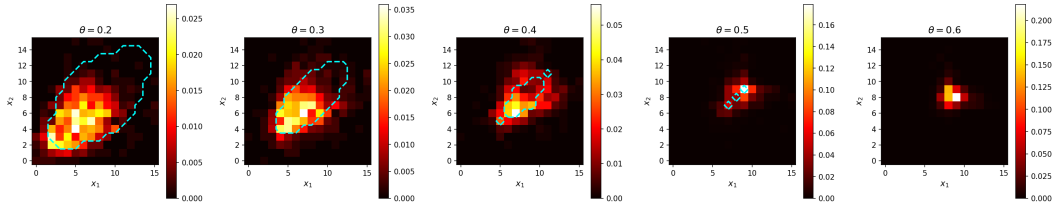
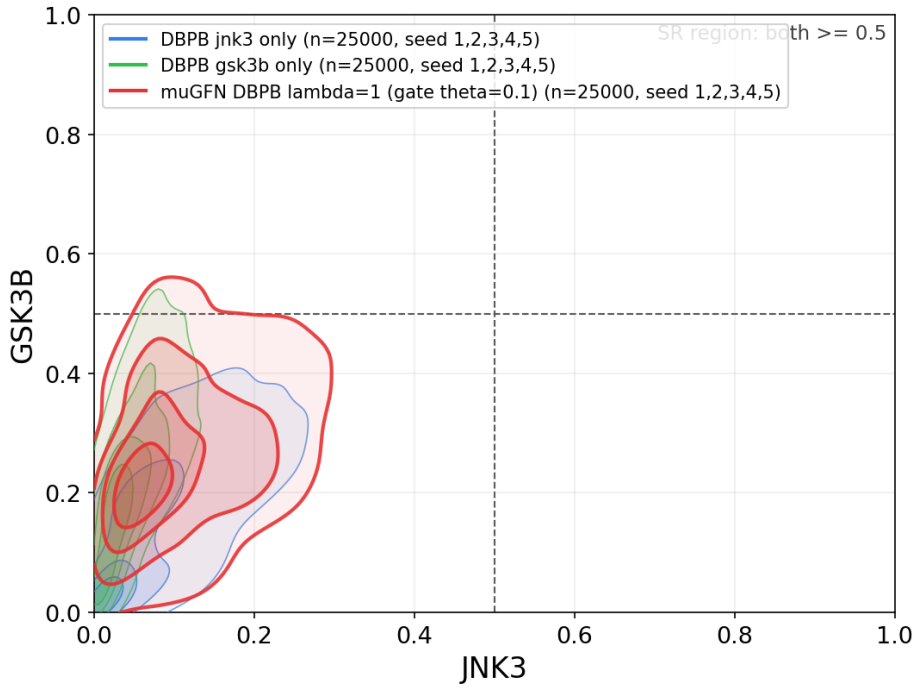


Figure 6: μGFN (FM) sampling distributions from the earlier reference experiment at thresholds $\theta \in \{0.2, 0.3, 0.4, 0.5, 0.6\}$; the DB sweep appears in Figure 2. Cyan contours outline the feasible region, which is empty at $\theta = 0.6$. Colors indicate the sample fraction at each grid state, with a separate color scale for each panel.

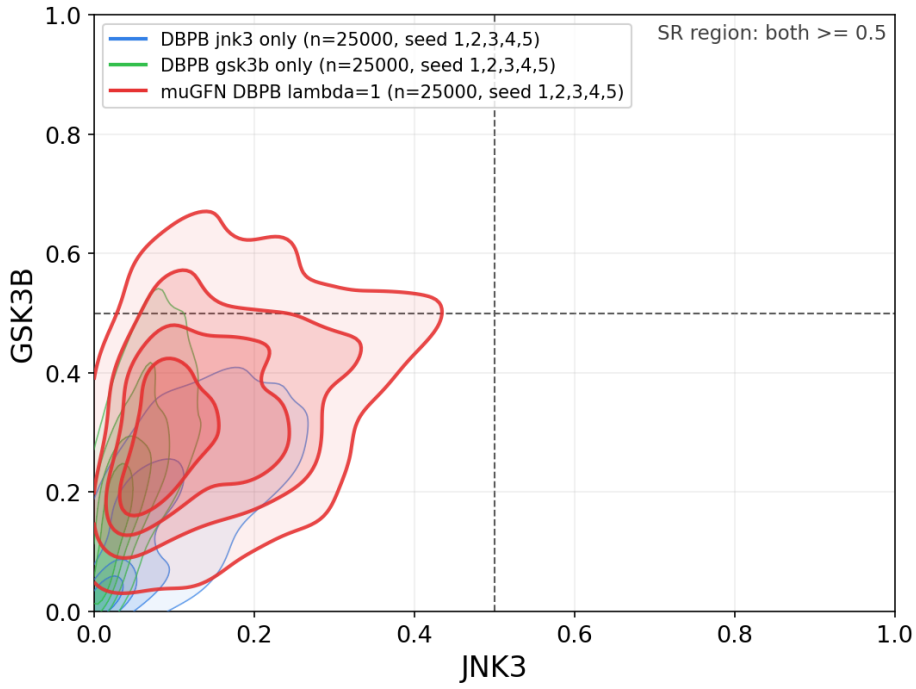
E.7 KDE Contour Analysis

Figure 7 compares the joint JNK3 and GSK3β score distributions of μGFN DBPB with dedicated single-objective DBPB models. Each distribution pools 25,000 generated molecules from seeds 1–5. The guided sampler uses $\lambda = 1$ at the two guidance settings shown in the panels. Evaluation remains fixed at JNK3 ≥ 0.5 and GSK3$\beta \geq 0.5$ in both panels, as marked by the dashed lines.

At the guidance setting $\theta = 0.1$ (Figure 7a), the red contours overlap the single-objective distributions and extend across a broader range of joint score combinations. At $\theta = 0.5$ (Figure 7b), they extend farther toward higher JNK3 and GSK3β scores, while the blue and green reference distributions are unchanged. The displayed contours therefore show how the guidance setting changes the joint score distribution. Much of the displayed density remains below the evaluation cutoffs, so the contours do not show concentration inside the jointly feasible region. Contour extent alone does not quantify the success rate.



(a) Guidance setting $\theta = 0.1$.



(b) Guidance setting $\theta = 0.5$.

Figure 7: Joint JNK3 and GSK3β score distributions as KDE contours for μGFN DBPB with $\lambda = 1$ (red), JNK3-only DBPB (blue), and GSK3β-only DBPB (green). Each distribution pools 25,000 generated molecules from seeds 1–5. The panels vary the guidance setting; dashed lines mark fixed evaluation cutoffs of 0.5 for both objectives. The joint success region is above and to the right of these lines. The contours visualize score distributions rather than success-rate estimates.

E.8 Exact Cross-Evaluation of Linear Hypergrid Guidance

A diagnostic sampler distinct from molecular composition. The hypergrid experiment reads $\widehat{\mu}_k(s,a)$ from the DB checkpoints using the flow-ratio readout. For each checkpoint seed, the same estimate table supplies all three rules and all four guidance requests. Write $S_\mu(s,a) = \sum_k \widehat{\mu}_k(s,a)$. The action scores are

$$q_{\mathrm{sum}}(s,a) = S_\mu(s,a),$$
$$q_{\mathrm{uncap}}(s,a;\boldsymbol{\theta}) = S_\mu(s,a) - \lambda \sum_k (\theta_k - \widehat{\mu}_k(s,a)),$$
$$q_{\mathrm{cap}}(s,a;\boldsymbol{\theta}) = S_\mu(s,a) - \lambda \sum_k [\theta_k - \widehat{\mu}_k(s,a)]_+,$$

and each rule uses $P(a \mid s) = \mathrm{softmax}_a(q(s,a)/T)$ over legal actions. We use $\lambda = 10$ and $T = 0.5$. The implementation labels these rules `norm`, `Uncapped`, and `Ours`, respectively. Here `norm` means ungated μ-sum scores, rather than the root-normalized flow composition in the molecular ablation. The linear grid gate contains neither $B(s,a)$ nor log-scale shortfalls. It tests the behavior of threshold gating, while the molecular experiments evaluate Equation (3).

Threshold cancellation and temperature. The uncapped score simplifies to

$$q_{\mathrm{uncap}}(s,a;\boldsymbol{\theta}) = (1+\lambda) S_\mu(s,a) - \lambda \sum_k \theta_k.$$

The threshold-dependent term is constant across actions and cancels upon normalization. Thus, at fixed λ and T, this rule is independent of the guidance request and equals the ungated μ-sum rule at temperature $T/(1+\lambda)$, here $0.5/11 \approx 0.04545$. This temperature identity is specific to the linear grid construction. For the molecular uncapped variant, changing λ modifies the μ term while leaving $\log B$ fixed, so it is not generally a temperature change of the normalized base. The capped grid rule can respond to objective-specific requests because above-threshold values receive no further guidance bonus.

Exact terminal probabilities. For DB checkpoint seeds $0, 1, 2$, we evaluate every guidance request in $\{(0.3, 0.3), (0.5, 0.3), (0.3, 0.5), (0.5, 0.5)\}$. The action probabilities depend only on the current grid state. We propagate root mass 1 through the DAG in increasing $i + j$ order, accumulating visits and terminal mass using the legal transitions and stop probabilities. This forward dynamic program computes the gate sampler's terminal distribution P_{term} up to floating-point precision, without sampled trajectories. For evaluation cutoffs $\boldsymbol{\tau}$,

$$\mathrm{SR}(\boldsymbol{\theta};\boldsymbol{\tau}) = \sum_{x:\, R_k(x) \geq \tau_k \ \forall k} P_{\mathrm{term}}(x;\boldsymbol{\theta}).$$

The reported spread is the sample standard deviation across the three checkpoint seeds ($\mathrm{ddof} = 1$), not Monte Carlo sampling error. It describes variability between learned checkpoints. This diagnostic is separate from the 1,000-sample estimates in Table 1 and does not evaluate the unguided DB model's own forward policy.

Interpretation. Table 8 reports the capped rule's cross-evaluation matrix. Holding evaluation at $(0.5, 0.3)$, increasing only the first guidance cutoff from 0.3 to 0.5 raises mean SR from 0.2286 to 0.6083. The corresponding second-objective change at evaluation $(0.3, 0.5)$ raises it from 0.2279 to 0.6076. Opposite asymmetric requests give much lower SR under those same evaluation conditions. These results support selective responses to objective-specific thresholds in the linear gate. Matching guidance is not always optimal. The looser evaluation $(0.3, 0.3)$ has its highest mean under guidance $(0.5, 0.5)$. The matrix compares requests within the capped sampler; it does not establish an advantage over uncapped guidance, temperature tuning, or the molecular sampling rule.

E.9 Design Properties and Additional Ablation Interpretation

The following properties distinguish the sampling rules analytically. Table 3 measures feasible yield and complementary quality metrics under these rules. It does not contain a flow-rescaling experiment or a molecular cross-evaluation across independently varied thresholds. The sampler definitions and shared controls appear in Appendix E.3.

Table 8: Exact SR of the capped linear hypergrid gate, with $\lambda = 10$ and $T = 0.5$ (mean $\pm$ sample SD over DB checkpoint seeds $0, 1, 2$). Rows vary guidance cutoffs; columns specify evaluation cutoffs. Each seed reuses the same checkpoint and $\widehat{\mu}$ table for all requests. Forward DP uses no sampled trajectories. Bold marks the highest mean within each evaluation column; displayed zero SDs are rounded.

Guidance $\boldsymbol{\theta}$	Evaluation cutoffs $\boldsymbol{\tau}$			
	$(0.3, 0.3)$	$(0.5, 0.3)$	$(0.3, 0.5)$	$(0.5, 0.5)$
$(0.3, 0.3)$	0.6376 ± 0.0000	0.2286 ± 0.0009	0.2279 ± 0.0007	0.0220 ± 0.0001
$(0.5, 0.3)$	0.7462 ± 0.0017	$\mathbf{0.6083 \pm 0.0002}$	0.0745 ± 0.0003	0.0466 ± 0.0004
$(0.3, 0.5)$	0.7460 ± 0.0035	0.0740 ± 0.0031	$\mathbf{0.6076 \pm 0.0012}$	0.0461 ± 0.0020
$(0.5, 0.5)$	$\mathbf{0.9954 \pm 0.0000}$	0.4672 ± 0.0101	0.4688 ± 0.0019	$\mathbf{0.2691 \pm 0.0071}$

Invariance of the normalized base. For any positive constants c_k, rescale each objective's first-head flows as $\widetilde{F}_k^{(1)}(s, a) = c_k F_k^{(1)}(s, a)$. Its root flow becomes $\widetilde{Z}_k = c_k Z_k$, giving

$$\widetilde{B}(s, a) = \sum_k \frac{c_k F_k^{(1)}(s, a)}{c_k Z_k} = B(s, a).$$

The raw sum instead becomes $\sum_k c_k F_k^{(1)}(s, a)$ and generally changes its normalized action probabilities. Root normalization therefore removes dependence of the base on global objective-flow scales. It preserves each objective's state-dependent reaching mass and does not assign equal influence at every state. This statement concerns B; rescaling only the first head also changes its flow ratio. If both heads of each objective receive the same multiplier, their ratios and the full guided policy remain unchanged as well.

Threshold cancellation without saturation. With fixed B, estimates, and λ, the uncapped distribution is

$$P_{\text{uncap}}(a \mid s) \propto B(s, a) \prod_k \left(\frac{\widehat{\mu}_k(s, a)}{\theta_k} \right)^{\lambda} \propto B(s, a) \prod_k \widehat{\mu}_k(s, a)^{\lambda}.$$

Thus, changing the requested thresholds cannot change this policy. For the capped rule, holding the estimates fixed and away from threshold crossings gives

$$\frac{\partial}{\partial \log \theta_k} \log \frac{P_\mu(a \mid s)}{P_\mu(b \mid s)} = -\lambda \left(\mathbf{1}\{\widehat{\mu}_k(s, a) < \theta_k\} - \mathbf{1}\{\widehat{\mu}_k(s, b) < \theta_k\} \right).$$

Increasing a threshold can therefore reduce the relative probability of an action below it compared with one above it. If all legal actions are on the same side of that objective's threshold, its variation within that interval leaves the policy unchanged. This is a local response property, rather than a guarantee that terminal satisfaction improves when a threshold increases.

Shortfalls without surplus compensation. Every contribution h_k is nonnegative and vanishes when its estimate meets the threshold. Increasing an already sufficient estimate supplies no bonus in the penalty to offset another objective's deficit. This property is restricted to the penalty; B can still favor an action with a larger deficit. Writing $H(s, a) = \sum_k h_k(s, a)$, the resulting policy solves

$$P_\mu(\cdot \mid s) = \arg\min_q \left\{ \lambda \, \mathbb{E}_{a \sim q}[H(s, a)] + \mathrm{KL}\big(q \,\|\, P_0(\cdot \mid s)\big) \right\}.$$

Appendix A.4 proves this identity and the weak decrease in expected predicted shortfall as λ increases for fixed state, estimates, and thresholds. Since the estimates concern separate objective policies, these local properties do not certify joint terminal feasibility under their composition. The molecular ablations assess their empirical utility.

Task-dependent effect of saturation. On JNK3 + GSK3β, capped guidance has mean $\mathrm{SR}_{\text{unique}}$ of 0.0055 versus 0.0050 for uncapped guidance, with higher mean HV (0.6707 versus 0.6667) and Geom (0.5274 versus 0.5246). On SEH + QED + SA, the capped mean $\mathrm{SR}_{\text{unique}}$ is 0.0655 versus

0.0381 for uncapped guidance (about $1.72\times$), with higher mean HV and Geom as well. On the four-objective task, uncapped guidance instead has higher means for unique success (0.0011 versus 0.0006), HV (0.4400 versus 0.4252), and Geom (0.5932 versus 0.5878). These are descriptive comparisons at the specified strengths and do not establish statistical significance. Capping enables threshold-dependent action selection, but that structural property does not ensure higher terminal success than uncapped reward guidance.

Contribution beyond normalization with four objectives. On JNK3 + GSK3β + QED + SA, root normalization raises mean HV from 0.2534 to 0.3507 and Geom from 0.4582 to 0.5297. Adding the capped penalty further increases these means to 0.4252 and 0.5878, and raises mean $\mathrm{SR}_{\mathrm{unique}}$ from 0.0002 to 0.0006. Greedy flow has higher mean HV (0.4371) than the capped rule, but lower unique success (displayed as 0.0000) and Geom (0.5383). Uncapped guidance exceeds both rules on all three mean metrics. The capped rule's mean yield still corresponds to about three unique feasible molecules per $5{,}000$ samples, compared with about five to six for uncapped guidance. The task differences and distinct λ values prevent attributing these cross-task patterns solely to objective count.

F GFlowNet Background

A GFlowNet (Bengio et al., 2021) defines a generative process over a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{S}, \mathcal{A})$, constructing objects by sequentially applying actions from an initial state s_0 to a terminal state $x \in \mathcal{X}$. A GFlowNet learns a forward policy $P_F(s'|s)$ such that the probability of generating terminal state x is proportional to its reward, giving $P_T(x) = R(x)/Z$, where $Z = \sum_{x'} R(x')$.

Flow Matching (FM) loss (Bengio et al., 2021) trains edge flows $F(s \to s')$ to satisfy flow conservation at every state. With $R(s)$ nonzero only at terminal states, its loss is

$$\mathcal{L}_{\text{FM}}(s) = \left(\log \frac{\sum_{s' \in \text{Pa}(s)} F(s' \to s)}{R(s) + \sum_{s'' \in \text{Ch}(s)} F(s \to s'')} \right)^2 \tag{23}$$

Detailed Balance (DB) loss (Bengio et al., 2023) enforces consistency on individual transitions. Let $F(s) > 0$ denote the state flow and $P_B(s \mid s')$ a backward policy normalized over the parents of s'. For each legal transition $s \to s'$, DB requires

$$F(s)P_F(s' \mid s) = F(s')P_B(s \mid s'), \tag{24}$$

with terminal boundary condition $F(x) = R(x)$. A corresponding squared log-residual loss is

$$\mathcal{L}_{\text{DB}}(s, s') = \left(\log \frac{F(s)P_F(s' \mid s)}{F(s')P_B(s \mid s')} \right)^2. \tag{25}$$

Training minimizes this loss over sampled transitions. With an edge-flow parameterization $F(s \to s') = F(s)P_F(s' \mid s)$, the balance condition becomes $F(s \to s') = F(s')P_B(s \mid s')$, the form used by our method.

Trajectory Balance (TB) loss (Malkin et al., 2022) operates on complete trajectories and is defined as

$$\mathcal{L}_{\text{TB}}(\tau) = \left(\log \frac{Z \prod_t P_F(s_{t+1}|s_t)}{R(x) \prod_t P_B(s_t|s_{t+1})} \right)^2 \tag{26}$$

FM imposes state-level flow conservation, DB constrains individual transitions, and TB constrains complete trajectories. Our primary model trains two reward heads with DB and shares a backward policy between them within each objective, providing the common path allocation used for the flow-ratio interpretation in Section 2.