Stable GFlowNets with TV Monitoring and Probabilistic Guarantees

Zengxiang Lei[1], **Ananth Shreekumar**[2], **Jonathan Rosenthal**[2], **Ruoyu Song**[2]
Alvaro A. Cardenas[3], **Daniel J. Fremont**[3], **Dongyan Xu**[2]
Satish Ukkusuri[1,2*], **Z. Berkay Celik**[2*]

[1]Lyles School of Civil and Construction Engineering, Purdue University, IN, USA
[2]Department of Computer Science, Purdue University, IN, USA
[3]Computer Science and Engineering, University of California, Santa Cruz, USA

[*]Corresponding authors: sukkusur@purdue.edu, zcelik@purdue.edu

Abstract

Generative Flow Networks (GFlowNets) sample diverse structured objects in proportion to reward and have been applied to molecular discovery and biological-sequence design, where finding multiple high-quality candidates is more useful than returning a single optimum. Despite their theoretical promise, practical training is often unstable, exhibiting severe loss spikes and mode collapse. To address this, we first assess the sensitivity of GFlowNet objectives, demonstrating that a small Total Variation (TV) distance between the learned and target distributions does not preclude an unbounded training loss. Motivated by this mismatch, we establish converse guarantees by deriving loss-to-TV bounds that certify global fidelity from bounded trajectory balance losses. Lastly, we propose *Stable GFlowNets*, which leverages our theoretical results to stabilize training via adaptive reference flow and improves the trade-off among mode coverage, robustness, and certifiability. Code is available at: https://github.com/tjleizeng/StableGFlowNets.

1 Introduction

Generative Flow Networks (GFlowNets) provide a principled framework for learning generative policies to sample states according to a target, unnormalized reward function [Bengio et al., 2021]. By modeling generation as a sequential decision process and enforcing flow consistency constraints, GFlowNets enable efficient sampling from complex combinatorial spaces and have shown promise in molecular design, biological sequence discovery, combinatorial optimization, and adversarial generation [Bengio et al., 2021, Jain et al., 2022, Shen et al., 2024, Zhang et al., 2023, Lahlou et al., 2023a].

Despite their successes, training GFlowNets remains challenging in practice. Prior work reports numerical instability [Madan et al., 2025, Malkin et al., 2022a] and difficulty in reliably capturing rare but high-reward modes [Pan et al., 2023, Kim et al., 2024a]. These behaviors contrast with the comparatively stable and scalable optimization of other likelihood-based generative models (e.g., diffusion models [Rombach et al., 2022]), and raise a fundamental question: *What guarantees on sampling fidelity can we certify with GFlowNet training?*

A central difficulty lies in the ambiguous relationship between non-zero GFlowNet losses and global sampling error. Unlike in reinforcement learning (RL) and diffusion models, where small training loss often yields theoretical guarantees that bound policy performance [Singh and Yee, 1994, Sutton et al., 1998, Song et al., 2021], common GFlowNets training objectives, including flow matching (FM), detailed balance (DB), and trajectory balance (TB), guarantee correctness only at their global optima

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

(i.e., when loss is zero everywhere) [Bengio et al., 2023, Malkin et al., 2022a]. In realistic regimes with finite data, function approximation, and non-stationary policies, optimization is necessarily approximate. Furthermore, we demonstrate in Section 3.1 that high loss signals can persist even when the learned policy is near-optimal. Consequently, it is unclear whether observed loss values meaningfully reflect distributional fidelity, or whether controlling the loss is sufficient to bound the mismatch between the learned state distribution and the target reward distribution.

In this work, we close this gap by establishing a rigorous connection between GFlowNet training losses and the global Total Variation (TV) error, i.e., the total variation distance between the learned distribution and the reward-proportional target. We show, in an analytically tractable setting, that small TV error does not imply bounded training loss, so extreme optimization signals may arise even when the learned distribution is globally accurate. We then derive loss-to-TV bounds that quantify global fidelity from bounded TB losses, and we further provide finite-sample TV certificates via trajectory sampling. Since loss spikes can be inevitable, we analyze *reference flow*, a customized flow injected into the flow conservation constraints to prevent extreme ratios and thus cap GFlowNet loss, as a stabilization mechanism that reduces loss magnitudes without altering the global objective, and quantify the resulting stability-fidelity trade-off. Building on these insights, we propose *Stable GFlowNets*, which uses adaptive reference flows to stabilize incremental learning and provide global or subgraph-level TV certificates depending on the available sampling oracle. Our contributions are summarized as follows:

1. We characterize the sensitivity of GFlowNet objectives and prove that a low Total Variation (TV) error does not imply bounded training loss.
2. We derive the first series of GFlowNet training loss-to-TV bounds that connect training losses to the fidelity of the learned distribution relative to the target distribution.
3. We introduce *Stable GFlowNets*, which use adaptive reference flows to stabilize training and improve mode coverage while supporting TV monitoring and probabilistic certification.

2 Preliminaries

A GFlowNet represents generation as a finite-state DAG (Figure 1): each node (or state) is a partially constructed object, each edge (or action) is a valid construction action, and each terminal state is a complete object with reward. Training encourages the total flow reaching each terminal state to be proportional to its reward, so that high-reward objects are sampled more often without collapsing to a single solution. In molecule or biological-sequence generation, for example, nodes correspond to partial molecules or sequences, and terminal nodes correspond to completed candidates. A formalized description is provided below.

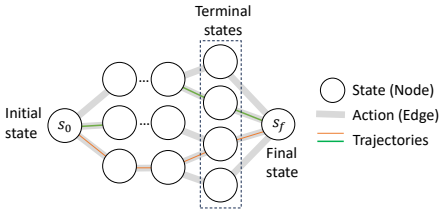

Figure 1: An illustrative GFlowNet DAG.

Notation. We consider a finite directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ where the set of nodes forms the state space $\mathcal{S}$ and the set of directed edges forms the action space $\mathcal{A}$. The unique state with no incoming edges is the initial state $s_0 \in \mathcal{S}$, and the unique state with no outgoing edges is the final state $s_f \in \mathcal{S}$. States that transition only to s_f are referred to as terminal (or end) states; their set is denoted by $\mathcal{X} \subseteq \mathcal{S}$. A (complete) trajectory is defined as a sequence of states $\tau = (s_0 \to \cdots \to s_n \to s_{n+1} = s_f)$ where the terminal state $s_n \in \mathcal{X}$ and each action (or edge) $s_i \to s_{i+1} \in \mathcal{A}$. In contrast, any prefix of such a sequence is a subtrajectory. The set of all complete trajectories is denoted by $\mathcal{T}$. Each trajectory τ is assigned a positive flow value $F : \tau \to \mathbb{R}_+$, which induces state-level flow and edge-level flow as $F(s) = \Sigma_{\tau \in \mathcal{T}, s \in \tau} F(\tau)$ and $F(s, s') = \Sigma_{\tau \in \mathcal{T}, s \to s' \in \tau} F(\tau)$, respectively. Each edge is further evaluated by a forward policy $P_F : \mathcal{S} \times \mathcal{S} \to (0, 1)$ and a backward policy $P_B : \mathcal{S} \times \mathcal{S} \to (0, 1)$; when evaluated on a specific edge $s \to s'$, they define the transition probability $P_F(s, s') = P(s' \mid s)$ and $P_B(s, s') = P(s \mid s')$. A positive scalar Z (the partition function, if treated as learnable) relates flow to policy via $F(\tau) = Z P_F(\tau) = Z \Pi_{t=0}^{n} P_F(s_t, s_{t+1})$, and we have the marginal distribution of the flow over terminal states $\mathcal{X}$ as $P_T(x) = F(x)/Z$. The reward satisfies $R(s) > 0$ for all $s \in \mathcal{X}$ and $R(s) = 0$ for $s \notin \mathcal{X}$. A table of notation is available in Appendix A.

GFlowNets. Depending on the specific parameterization, GFlowNets learn either an edge flow F or a state flow F (or a partition function Z) along with a forward policy P_F (and optionally with a backward policy P_B). The training objectives ensure that the generated flow on the DAG $\mathcal{G}$ results in a sampling probability proportional to the reward, $F(x) \propto R(x)$ for all $x \in \mathcal{X}$. The backward policy is optional because the GFlowNet framework permits P_B to be chosen arbitrarily [Bengio et al., 2023]. However, fixing P_B constrains the forward policy P_F to a unique solution, whereas learning P_B jointly with P_F can empirically accelerate convergence [Malkin et al., 2022a]. The choice between learning F versus learning Z depends on the training objective: objectives such as trajectory balance estimate Z jointly with P_F, while others learn flows F and take $Z = F(s_0)$ implicitly. Given the terminal state space $\mathcal{X}$, the true partition function Z^* can be computed explicitly as $Z^* = \sum_{x \in \mathcal{X}} R(x)$ and the ground-truth target distribution is $\pi_{target}(x) = \frac{R(x)}{Z^*}$.

Training Objectives. GFlowNets can be trained using several objectives that enforce the global flow-matching condition $F(x) \propto R(x)$ through different forms of local consistency. These objectives differ in where the consistency constraint is localized: on individual edges, on flow conservation at states, or on trajectories.

- **Flow Matching (FM).** Bengio et al. [2021] mandate flow conservation at each intermediate state:

$$\mathcal{L}_{FM}(s') = \left(\log \frac{\sum_{s \to s' \in \mathcal{A}} F(s, s')}{R(s') + \sum_{s' \to s'' \in \mathcal{A}, s'' \neq s_f} F(s', s'')} \right)^2 \tag{1}$$

  In Bengio et al. [2021], a hyperparameter δ was introduced to mitigate numerical issues. For simplicity in our theoretical analysis, we omit δ here and discuss its role in Section 4.
- **Detailed Balance (DB).** Bengio et al. [2023] enforce local consistency on individual edges by equating forward and backward flows:

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

  where $s \to s' \in \mathcal{A}$, $s' \neq s_f$, and $F(s') := R(x)$ when $s' = x \in \mathcal{X}$.
- **Trajectory Balance (TB).** Malkin et al. [2022a] impose consistency on complete trajectories $\tau = (s_0 \to s_1 \to \cdots \to s_n = x \to s_f)$:

$$\mathcal{L}_{TB}(\tau) = \left(\log \frac{Z \Pi_{t=0}^{n-1} P_F(s_t, s_{t+1})}{R(x) \Pi_{t=0}^{n-1} P_B(s_t, s_{t+1})} \right)^2 \tag{3}$$

- **Subtrajectory balance (subTB).** Madan et al. [2023] extend TB to subtrajectories $\tau^{partial} = (s_{t_1} \to \cdots \to s_{t_2})$:

$$\begin{aligned} &\mathcal{L}_{subTB}(\tau_{t_1:t_2}) \\ &= \left(\log \frac{F(s_{t_1}) \Pi_{t=t_1}^{t_2-1} P_F(s_t, s_{t+1})}{F(s_{t_2}) \Pi_{t=t_1}^{t_2-1} P_B(s_t, s_{t+1})} \right)^2 \end{aligned} \tag{4}$$

  where $F(s_{t_2}) := R(x)$ when $s_{t_2} = x \in \mathcal{X}$.

Total Variation (TV) Error. Total Variation (TV) error quantifies the distributional discrepancy and is central to recent theoretical analyses of GFlowNets [Silva et al., 2025a]. The TV error corresponds to half the total L_1 error:

$$\mathrm{TV}(P_T, \pi_{target}) = \frac{1}{2} \sum_{x \in \mathcal{X}} \left| P_T(x) - \pi_{target}(x) \right| \tag{5}$$

3 On the Sensitivity and Fidelity of GFlowNets

Our analysis is motivated by incremental mode coverage, where a GFlowNet must incorporate new high-reward modes without catastrophic forgetting. Section 3.1 shows that standard GFlowNet objectives are ill-conditioned in this setting: small target-distribution perturbations can induce unbounded

local losses through worst-case local contrast ratios. This motivates the reverse question: what distributional fidelity can be guaranteed when training losses are bounded? Section 3.2 answers this by linking training loss to TV error (Theorem 3.5) and deriving a trajectory-sampling-based probabilistic certificate (Theorem 3.6). Section 3.3 then introduces reference flow to smooth local contrast and stabilize training. Although reference flow caps loss magnitudes, it imposes an inherent fidelity trade-off: a multiplicative degradation governed by the augmentation magnitude (Theorem 3.10). Finally, Theorem 3.11 provides a probabilistic certificate for reference-flow-stabilized GFlowNets, directly motivating the training procedure in Section 3.4.

3.1 Incremental Mode Coverage Causes Loss Explosion

To lay the groundwork, we examine the simplified setting in Figure 2. The following remarks illustrate a sharp mismatch between global flow error and local training behavior. Remark 3.1 shows that adding a reward to a single leaf in a large tree (g^h leaves) results in a negligible Total Variation error of $TV(P_T, \pi_{target}) \approx 1/g^h$. However, Remark 3.2 reveals that the local loss scales as $(\log \epsilon)^2$. As $\epsilon \to 0$, the training loss diverges to $+\infty$, even though the global distribution is nearly correct.

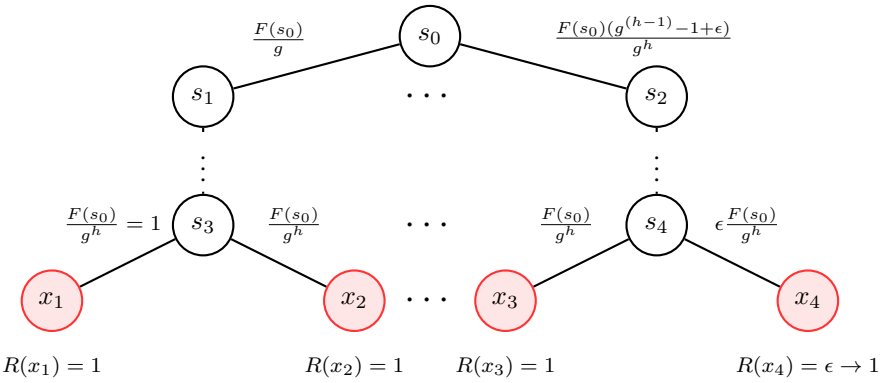

Figure 2: The "one more mode" learning setup on a g-ary tree of depth h. The underlying structure is a regular tree. We initialize the experiment with an already fitted model where this specific node leads to a negligible reward $R(x_4) = \epsilon$ ($\ll 1$). To introduce the new mode, we update the reward function by promoting x_4 to a high-reward state with $R(x_4) = 1$. The objective is to learn this new mode while preserving the previously acquired ones.

Remark 3.1. Flow error for "one more mode" learning over a regular tree.

$$\mathrm{TV}(P_T, \pi_{target}) = \frac{(g^h-1)}{2}\left(\frac{1}{g^h-1+\epsilon} - \frac{1}{g^h}\right) + \frac{1}{2}\left(\frac{1}{g^h} - \frac{\epsilon}{g^h-1+\epsilon}\right) \approx \frac{1}{g^h} \tag{6}$$

Remark 3.2. Training loss for "one more mode" learning over a regular tree.

Most losses are zero; non-zero losses are shown below and are all $(\log \epsilon)^2$.

$$\mathcal{L}_{FM}(x_4) = \mathcal{L}_{DB}(s_4, x_4) = \mathcal{L}_{TB}(\tau \ni x_4) = \mathcal{L}_{subTB}(\tau^{partial} \ni x_4) = (\log \epsilon)^2 \tag{7}$$

This implies that the GFlowNet training signal does not necessarily reflect the scale of the distributional mismatch. Instead, it can be dominated by worst-case local contrast ratios (i.e., states where the learned flow is orders of magnitude smaller than what the target implies).

For a general **incremental mode coverage** setting, we assume that the network has converged to an initial reward landscape $R_{prev}(x)$ such that $F(x) = R_{prev}(x)$, and we are introducing a new reward $R_{new}(x) = R_{prev}(x) + R'(x)$ where $R'(x) \geq 0$.

Proposition 3.3. *(TV bound for incremental mode coverage over arbitrary state graph). Let the reward added at each state in a subset $\mathcal{X}_{sub}$ be $R'(x)$. We define the local true partition function $Z^*_{\mathcal{Y}\subseteq\mathcal{X}} = \sum_{x\in\mathcal{Y}} R(x)$ and the local contrast ratio $\Lambda_{\mathcal{Y}} = \frac{Z^*_{\mathcal{Y}}}{Z^*_{\mathcal{Y}}+\sum_{x\in\mathcal{Y}} R'(x)}$. Then, we have*

$$\frac{Z^* - Z^*_{\mathcal{X}_{sub}}}{Z^*}(1-\Lambda_{\mathcal{X}}) \leq \mathrm{TV}(P_T, \pi_{target}) \leq (1-\Lambda_{\mathcal{X}}) \tag{8}$$

Here, note that $Z^* = Z^*_{\mathcal{X}}$ is the true partition function on the original reward function.

Proposition 3.4. *(Loss scale via local contrast). While the TV error depends on the aggregate $\Lambda_{\mathcal{X}}$, the supremum of the training loss is governed by the **worst-case** local contrast ratio, given by:*

$$\sup \mathcal{L}_{GFN} = \left(\log \min_{\{x\} \subseteq \mathcal{X}_{sub}} \Lambda_{\{x\}} \right)^2 \tag{9}$$

where $GFN \in \{FM, DB, TB, subTB\}$.

Proofs of Propositions 3.3 and 3.4 are given in Appendix B. Together, they show that incremental mode coverage in GFlowNets is governed by two quantities: the aggregate contrast ratio, e.g., $1 - \Lambda_{\mathcal{X}}$, which controls global TV error and may remain small when the new mode has little reward mass; and the worst-case local contrast ratio, $\min_{\{x\} \subseteq \mathcal{X}_{sub}} \Lambda_{\{x\}}$, which controls the scale of losses and gradients through terms such as $(\log \min_{\{x\}} \Lambda_{\{x\}})^2$. Thus, when a new mode requires a large relative reward increase at rarely visited states, the training dynamics degenerate into a regime where tiny changes in the target distribution cause huge optimization signals.

3.2 The General Link Between GFlowNets Training Loss and TV Error

As TV error does not effectively bound GFlowNet training losses, we pursue the reverse direction to bound the TV error of the resulting policy if the training loss can be contained, yielding the following theorems for the general setting.

Theorem 3.5. *(Training loss to TV distance bound). The relationship between the training loss bound and the resulting TV distance depends on the scope of the objective (trajectory-level vs. transition-level):*

***Trajectory-level Objective.** If the trajectory loss is bounded, i.e., $\mathcal{L}_{TB}(\tau) \leq c^2, \forall \tau \in \mathcal{T}$, the TV error is bounded by:*

$$\mathrm{TV}(P_T, \pi_{target}) \leq 1 - e^{-2c} \tag{10}$$

This bound is independent of the trajectory length, as TB optimizes the full path consistency directly.

***Transition-level Objective.** If the local transition loss is bounded, i.e., $\mathcal{L}_{DB}(s, s') \leq c^2$ or $\mathcal{L}_{FM}(s') \leq c^2$, the global consistency relies on the accumulation of local estimates. For trajectories of maximum number of transitions L, the error bound degrades linearly with depth in the log-domain:*

$$\mathrm{TV}(P_T, \pi_{target}) \leq 1 - e^{-2Lc} \tag{11}$$

Since DB and FM bounds depend on the maximum trajectory length, we focus our analysis on the TB loss. Extensions to the DB and FM cases follow directly by introducing the trajectory length into the bounds. The proof can be found in Appendix B.

As verifying the loss $\mathcal{L}_{TB} \leq c^2$ for each trajectory is intractable, we develop a probabilistic certificate using random sampling. Notably, this certificate is independent of the state-space size.

Theorem 3.6. *(Probabilistic TV bound via trajectory sampling). Given the ground-truth target distribution π_{target}, we define a target distribution over trajectories $\hat{\pi}(\tau) = \pi_{target}(x_\tau) P_B(\tau | x_\tau)$. Sample m trajectories $\tau_1, \ldots, \tau_m$ from $\hat{\pi}$ independently by sampling $x \sim \pi_{target}$ and $\tau \sim P_B(\cdot | x)$. Sample another n trajectories independently using P_F. Let $c = \max_{i \leq m+n} \sqrt{\mathcal{L}_{TB}(\tau_i)}$, with confidence $1 - 2\alpha$, the global TV error is bounded by:*

$$\mathrm{TV}(P_T, \pi_{target}) \leq e^{2c} + 1 - \alpha^{\frac{1}{m}} - \alpha^{\frac{1}{n}} \leq e^{2c} - 1 + \frac{\log(1/\alpha)}{m} + \frac{\log(1/\alpha)}{n} \tag{12}$$

The proof is provided in Appendix B. When sampling end states from the full ground-truth target distribution is replaced by sampling from a subset $\mathcal{X}_{sub} \subseteq \mathcal{X}$, the theorem yields a *subgraph certificate* over $\mathcal{X}_{sub}$.

Corollary 3.7. *(Subgraph Certification via trajectory sampling). Let $\mathcal{X}_{sub} \subseteq \mathcal{X}$ be a subset of end-states. Define the restricted target distribution over $\mathcal{X}_{sub}$ by $\pi^{sub}_{target}(x) = R(x) / \sum_{x \in \mathcal{X}_{sub}} R(x)$, the corresponding restricted target trajectory distribution as $\hat{\pi}_{sub}(\tau) = \pi^{sub}_{target}(x_\tau) P_B(\tau | x_\tau)$. Sample*

m trajectories $\tau_1, \ldots, \tau_m$ from $\hat{\pi}_{sub}$. Sample another n trajectories that end within $\mathcal{X}_{sub}$ independently using P_F. Suppose we observe $\mathcal{L}_{TB}(\tau_i) \le c^2$ for all trajectories in both sets. Let P_T^{sub} denote the terminal flow P_T renormalized to $\mathcal{X}_{\text{sub}}$. Then, with confidence $1-2\alpha$,

$$\mathrm{TV}(P_T^{sub}, \pi_{target}^{sub}) \le e^{2c} + 1 - \alpha^{\frac{1}{m}} - \alpha^{\frac{1}{n}} \le e^{2c} - 1 + \frac{\log(1/\alpha)}{m} + \frac{\log(1/\alpha)}{n} \tag{13}$$

Throughout the paper, when global backward sampling is replaced by subgraph-based sampling, the certificate applies only to the corresponding subgraph.

3.3 Reference Flow: Stability with Fidelity Trade-off

The bounds in Section 3.2 are only useful if the training loss can be kept small, a requirement that could be easily violated as seen in Section 3.1. To resolve this, we investigate **reference flow**, which artificially increases the background flow to reduce the training losses.

Definition 3.8. (Trajectory reference flow). For a trajectory τ, let $R(\tau) = R(x)P_B(\tau|x)$ be the target flow. We introduce a trajectory-specific, non-negative reference flow $\delta(\tau) \ge 0$ to augment the existing one. The augmented flow $F_{aug}(\tau)$ and augmented target $R_{aug}(\tau)$ are defined as: $F_{aug}(\tau) = ZP_F(\tau) + \delta(\tau), \quad R_{aug}(\tau) = R(\tau) + \delta(\tau)$

Remark 3.9. (Stabilization via reference flow). Let $\mathcal{L}_{TB}(\tau) = \left(\log \frac{ZP_F(\tau)}{R(\tau)}\right)^2$ be the standard TB loss. The reference flow proportionally reduces the scale of a positive TB loss:

$$\mathcal{L}_{aug}(\tau) = \left(\log \frac{F_{aug}(\tau)}{R_{aug}(\tau)}\right)^2 = \frac{1}{\gamma^2}\mathcal{L}_{TB}(\tau) \qquad \gamma := \frac{\log(ZP_F(\tau)/R(\tau))}{\log[(ZP_F(\tau)+\delta(\tau))/(R(\tau)+\delta(\tau))]} \tag{14}$$

It is easy to verify that $\gamma^2 > 1$ when $\delta(\tau) > 0$. To make $\mathcal{L}_{aug}(\tau) \le c^2$, we have the minimum reference flow:

$$\delta_c(\tau) = \begin{cases} \frac{ZP_F(\tau) - e^c R(\tau)}{e^c - 1} & \text{if } \log \frac{ZP_F(\tau)}{R(\tau)} > c \\ \frac{R(\tau) - e^c ZP_F(\tau)}{e^c - 1} & \text{if } \log \frac{ZP_F(\tau)}{R(\tau)} < -c \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Treating reference flows as target modifications enables Theorem 3.5 to expose the following trade-off:

Theorem 3.10. *(Fidelity trade-off under reference flow). The fidelity of the recovered policy depends on the ratio between the training loss and the augmentation magnitude. Let the total reference flow be $\Delta = \sum_{\tau \in \mathcal{T}} \delta(\tau)$. If the reference training loss is bounded by $\mathcal{L}_{aug}(\tau) \le c^2$, the terminal distribution P_T induced by the learned forward policy satisfies:*

$$\mathrm{TV}(P_T, \pi_{target}) \le \frac{(1-e^{-2c})(1+\Delta/Z^*)}{1+(1-e^{-c})\Delta/Z^*} \le (1-e^{-2c})(1+\frac{\Delta}{Z^*}) \tag{16}$$

Note $\frac{\Delta}{Z^*}$ can be expressed as the expectation of $\frac{\delta(\tau)}{R(\tau)}$ over the target distribution $\hat{\pi}(\tau) = \frac{R(\tau)}{Z^*}$, **we can approximate the TV bound via Monte Carlo estimation** ($\mathcal{M}_{\text{TV}}$):

$$\mathcal{M}_{\text{TV}} := \min_{c \in \left(0, \max_i \sqrt{\mathcal{L}_{TB}(\tau_i)}\right]} (1-e^{-2c}) \left(1 + \frac{1}{m}\sum_{i=1}^{m} \frac{\delta_c(\tau_i)}{R(\tau_i)}\right), \quad \tau_i \sim \hat{\pi} \tag{17}$$

This observation also yields the following probabilistic bound.

Theorem 3.11. *(Probabilistic TV bound with optimizable reference-flow threshold). Sample $m(\ge 2)$ trajectories $\tau_1, \ldots, \tau_m$ independently from the target $\hat{\pi}$, sample another $n(\ge 2)$ trajectories $\tau_{m+1}, \ldots, \tau_{m+n}$ independently using P_F. For each $c > 0$, we compute the minimum reference flow $\delta_c(\tau_i)$ according to Equation (15). Define $M_c := \max_{i \in \{1,\ldots,m+n\}} \frac{\delta_c(\tau_i)}{R(\tau_i)}$, and $\mathcal{C} = \{c > 0 \mid M_c < \frac{1}{e^c - 1}\}$. With confidence $1-2\alpha$, the following bound holds simultaneously for every $c \in \mathcal{C}$:*

$$\begin{aligned} \mathrm{TV}(P_T, \pi_{target}) &\le \left(\frac{e^c + (e^c-1)M_c}{e^{-c} - (1-e^{-c})M_c} - 1\right) + \beta_m(\alpha) + \beta_n(\alpha) \\ &\le \left(\frac{e^c + (e^c-1)M_c}{e^{-c} - (1-e^{-c})M_c} - 1\right) + \frac{2\log(2/\alpha)}{m} + \frac{2\log(2/\alpha)}{n} \end{aligned} \tag{18}$$

where $\beta_k(\alpha)$ is solved from

$$(1-\beta_k(\alpha))^{k-1}[1+(k-1)\beta_k(\alpha)] = \alpha \tag{19}$$

Since c can be selected within a range in the above theorem, we define

$$\mathcal{B}_{\mathrm{TV}} := \min_{c\in\mathcal{C}} \left(\frac{e^c + (e^c-1)M_c}{e^{-c} - (1-e^{-c})M_c} - 1 \right) + \frac{2\log(2/\alpha)}{m} + \frac{2\log(2/\alpha)}{n} \tag{20}$$

3.4 Applications to GFlowNets Training: Stable GFlowNets

Practical Implications of Our Analysis. Our theoretical analysis leads to several practical implications for GFlowNet training. First, our results indicate that training stability requires explicit attention and is likely to become increasingly important as the state space grows and rewards become sparser. Second, Theorem 3.6 shows that reliably optimizing trajectories beyond those sampled by the current forward policy, such as backward-sampled trajectories, is important for obtaining probabilistic performance guarantees. This perspective also helps explain the effectiveness of GFlowNet training variants that use guided exploration or replay to improve coverage of high-reward trajectories (see Appendix C for an extended discussion). Third, $\mathcal{M}_{\mathrm{TV}}$ can serve as an efficient training-time monitoring signal when exact TV is computationally prohibitive, whereas $\mathcal{B}_{\mathrm{TV}}$ provides a conservative, high-confidence certificate of model quality.

The Stable GFlowNets Algorithm. These implications directly motivate our Stable GFlowNets algorithm (Algorithm 1), which adaptively injects a reference flow $\delta(\tau)$ based on instantaneous mismatch and monitors the resulting probabilistic TV certificate. When exact backward sampling over $\mathcal{X}$ is intractable, certification is restricted to a subgraph $\mathcal{X}_{\text{sub}} \subset \mathcal{X}$, implemented as a top-$K$ high-reward buffer; in practice, a modest K often captures most of the reward mass. Full implementation details are deferred to Appendix D, including the exponential moving-average update for the loss threshold c and the bounded 1D optimization used to compute $\mathcal{B}_{\mathrm{TV}}$ and $\mathcal{M}_{\mathrm{TV}}$.

4 Related Work

Theoretical Analysis of GFlowNets. Theoretical understanding of GFlowNets has grown substantially since their formulation as flow-matching models on directed acyclic graphs [Bengio et al., 2023]. Trajectory Balance (TB) [Malkin et al., 2022a] investigated long-horizon credit assignment, while subsequent work connected GFlowNets to continuous generative modeling, diffusion models [Lahlou et al., 2023a], variational inference [Malkin et al., 2022b], and entropy-regularized reinforcement learning [Tiapkin et al., 2024].

Recent theoretical work has advanced performance assessment for GFlowNets. Krichel et al. [2024] relate the average TB loss to its sampled estimate, while Silva et al. [2025a] connect flow perturbations to total variation (TV) error and introduce Flow Consistency in Sub-graphs (FCS) as a scalable evaluation metric. However, FCS is not directly tied to the GFlowNet training loss, distinguishing their setting from ours; we empirically compare FCS with $\mathcal{M}_{\mathrm{TV}}$ in Section 5.3. Furthermore, Silva et al. [2025b] bounded TV using trajectory-level violations $(\log \frac{P_F(\tau)}{\pi(x)P_B(\tau|x)})^2$. This reliance on the true target distribution $\pi(x)$ contrasts with our focus on unconverged training losses based on learned quantities, where $\frac{Z}{R(x)} \neq \frac{1}{\pi(x)}$.

Theoretical Analysis of RL and Diffusion. In related domains like RL [Singh and Yee, 1994, Schulman et al., 2015] and diffusion [Song et al., 2021], bounded training objectives directly yield theoretical performance guarantees. GFlowNets combine distribution matching (as in diffusion) with active exploration for high-reward states (as in RL), yet differ from both: they train without a fixed dataset and optimize distributional fidelity rather than reward maximization. This makes it nontrivial to determine when training-time samples faithfully represent global objectives.

GFlowNets Training. We focus on methods relevant to our stable GFlowNets algorithm (see Appendix C for a broader discussion). To mitigate persistent numerical instability, prior works add a constant δ to transition flows [Bengio et al., 2021, 2023], or clip losses [Lahlou et al., 2023b]

Table 1: Overview of the research questions, evaluation purposes, evidence, and main finding.

RQ	Purpose	Evidence	Main finding
RQ1: How severe is the loss imbalance in standard GFlowNet training?	Characterize practical loss instability during training.	**Tasks:** All environments. **Methods:** DB, FM, and TB. **Metrics:** Training loss and Max-to-Rest Loss Ratio.	A few trajectories can dominate the training objective even as the aggregate loss decreases.
RQ2: Does the proposed Stable GFlowNet improve training stability, convergence, and mode coverage?	Evaluate Stable GFlowNet and isolate the contributions of its components under matched sampling budgets.	**Tasks:** Hypergrid, L14-RNA1, and sEH. **Methods:** DB, TB, SubTB, WDB, Teacher, Stable, and StableTeacher. **Metrics:** L_1 error, diversity, and variability across seeds.	Stable GFlowNet improves training robustness and sampling quality, particularly on more challenging tasks.
RQ3: How informative are the TV bounds derived in Theorems 3.10 and 3.11?	Evaluate global and subgraph-level monitoring of distributional error.	**Tasks:** Regular Tree, Hypergrid, and L14-RNA1. **Metrics:** True TV, FCS, $\mathcal{M}_{\mathrm{TV}}$, and $\mathcal{B}_{\mathrm{TV}}$.	$\mathcal{M}_{\mathrm{TV}}$ tracks distributional error closely, whereas $\mathcal{B}_{\mathrm{TV}}$ provides a conservative certificate.

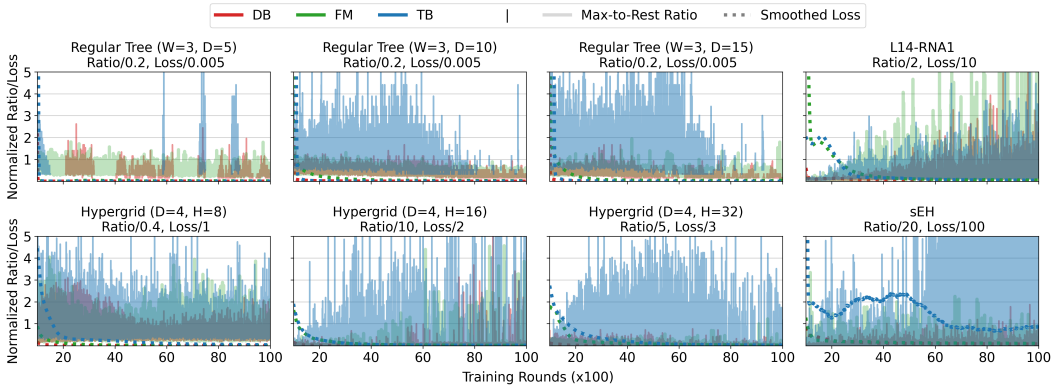

Figure 3: Loss Concentration and Training Stability. Solid lines show the Max-to-Rest Loss Ratio, defined as $\frac{\max_i \mathcal{L}_{GFN}(\tau_i)}{\sum_{j \neq i} \mathcal{L}_{GFN}(\tau_j)}$, and dotted lines show the training loss smoothed over 1,000 steps. Large spikes indicate that a single trajectory can contribute more loss than all remaining trajectories combined, revealing severe trajectory-level imbalance despite the steadily decreasing smoothed loss.

and gradients [Shen et al., 2023]. However, clipping lacks bounded loss certificates. Furthermore, Theorem 3.10 shows that a fixed δ introduces a resolution loss that can render TV guarantees vacuous without fully stabilizing training. Efficient exploration is also critical, traditionally promoted via forward-policy annealing [Lahlou et al., 2023b], transition augmentations [Pan et al., 2023], local backtracking [Kim et al., 2024a], or adaptive teachers [Kim et al., 2025a]. We reframe exploration as a tool for *certification*: backward trajectories sampled from the reward distribution and P_B bound global TV error, emphasizing the necessity of discovering and visiting high-reward trajectories.

5 Experiments

Environments. We evaluate our methods across four environments:

- **Regular Tree:** A 3-ary tree of depth D. Each leaf node corresponds to a terminal state and receives a unit reward. This environment is simple to train and allows exact computation of TV error, making it well-suited for validating our theory.
- **Hypergrid:** A grid-based environment introduced by Bengio et al. [2021]. It is parameterized by dimension D, side length H, and three reward coefficients R_0, R_1, and R_2. We use $R_0 = 10^{-2\log_2(H/8)-1}$, $R_1 = 0.5$, and $R_2 = 2.0$.
- **L14-RNA1:** The generated objects are RNA sequences of length 14. The reward function is a binding affinity to a human transcription factor, obtained via a pre-trained proxy model from Sinai et al. [2020]. Following Kim et al. [2025a], we use a reward exponent of 40 and define modes as the top 0.01% quantile of $R(x)$. Diversity filtering with a Levenshtein distance threshold of 1 is enforced, resulting in 8,967 modes out of 268,435,456 possible end states.
- **sEH:** Following Bengio et al. [2021], this environment involves generating small molecule graphs targeting the soluble epoxide hydrolase (sEH) protein. Molecules are constructed step-by-step using a vocabulary of molecular building blocks, with up to 105 actions available per state. This

Table 2: Performance comparison. All methods are trained for 10^4 rounds, and Hypergrid evaluation uses 10^5 samples from the final policy P_F. RNA mode counts and sEH scaffold counts are accumulated over training. All results are mean $\pm$ std over 5 seeds.

	Hypergrid (Empirical Total $L_1 \downarrow$)		**L14-RNA1**	**sEH**
Method	$D = 4, H = 16$	$D = 4, H = 32$	# modes $\uparrow$	# scaffolds $\uparrow$
TB	1.885 ± 0.000	1.875 ± 0.000	1669.2 ± 300.0	137.2 ± 77.9
DB	0.315 ± 0.004	0.944 ± 0.017	1285.2 ± 16.4	32.2 ± 12.5
SubTB	0.292 ± 0.001	0.749 ± 0.005	1904.0 ± 54.0	68.4 ± 41.9
WDB	0.490 ± 0.037	1.510 ± 0.024	1364.6 ± 64.5	3.2 ± 2.8
Teacher	0.407 ± 0.043	1.603 ± 0.083	1991.0 ± 189.3	5.6 ± 5.9
Stable	$\mathbf{0.290 \pm 0.002}$	$\mathbf{0.713 \pm 0.002}$	1734.0 ± 22.4	$\mathbf{14142.6 \pm 2388.2}$
StableTeacher	0.315 ± 0.004	0.815 ± 0.010	$\mathbf{2622.8 \pm 131.4}$	3722.2 ± 4187.8
TB + backward sampling[†]	0.316 ± 0.003	$\mathbf{0.713 \pm 0.003}$	1740.8 ± 37.2	8680.2 ± 2254.2

[†] Internal ablation of Stable GFlowNets that retains backward sampling but disables reference-flow stabilization.

combinatorial process results in a massive state space of approximately 10^{16} reachable terminal states. Performance is evaluated by counting the number of distinct Bemis–Murcko scaffolds among molecules whose raw reward is at least 7.5.

Baselines. For RQ1, we examine loss behavior under standard GFlowNet objectives: TB [Malkin et al., 2022a], DB [Bengio et al., 2023], and FM [Bengio et al., 2021]. For the RQ2 performance comparison, we evaluate TB, DB, SubTB [Madan et al., 2023], Adaptive Teacher (Teacher) [Kim et al., 2025a], and Weighted DB (WDB) [Silva et al., 2025a]. We exclude FM from RQ2 because its flow-conservation loss requires aggregating incoming flows over all parents of each visited state, involving additional parent-state evaluations outside the sampled trajectories. To separate the contributions of backward sampling and reference-flow stabilization, we include an internal ablation, "TB + backward sampling", which disables reference flow in Stable GFlowNets while keeping the remaining training procedure unchanged. All methods considered in RQ2 use a batch size of 32 and are trained for 10^4 rounds. We match the number of training trajectories, while allowing method-specific mixtures of forward, backward, and teacher sampling; the sampling compositions are detailed in Appendix E.2. For L14-RNA1, we employ reward-prioritized replay [Shen et al., 2023] and ϵ-greedy exploration [Jain et al., 2022]. For backward sampling, we use the ground-truth terminal state distribution on the Regular Tree and Hypergrid; on L14-RNA1 and sEH, we utilize a buffer of the top-10,000 highest-reward states discovered during training. The correspondence among environments, baselines, and RQs is summarized in Table 1.

5.1 Diagnosing Loss Instabilities (RQ1)

Figure 3 shows that batch losses can become heavily concentrated on a single trajectory, and that this concentration generally increases with state size. One exception is the medium-sized Hypergrid, which exhibits larger Max-to-Rest Loss Ratio spikes than the larger Hypergrid and we investigate this behavior in Appendix F.1.

5.2 Stability and Mode Coverage Improvements (RQ2)

Table 2 shows that Stable GFlowNets matches the best baselines on easier Hypergrid tasks and shares the lowest reported mean with TB + backward sampling on the harder $H = 32$ setting, and discovers substantially more scaffolds than all baselines on the large-scale sEH task. On L14-RNA1, StableTeacher discovers the most modes during training. Stable discovers a similar number of modes as TB with backward sampling, with lower variability across five seeds, although its average discovery count is below those of Teacher and SubTB. On sEH, Stable achieves the highest average scaffold count among the tested methods.

The comparison among standard TB, our backward sampling ablation, and Stable separates the gains from backward sampling from those provided by reference flow stabilization. Replacing half of the forward-sampled trajectories with backward-sampled trajectories is already a powerful training mechanism, consistent with our theory that capping losses on both forward- and backward-sampled trajectories leads to stronger performance guarantees. On Hypergrid and L14-RNA1, backward

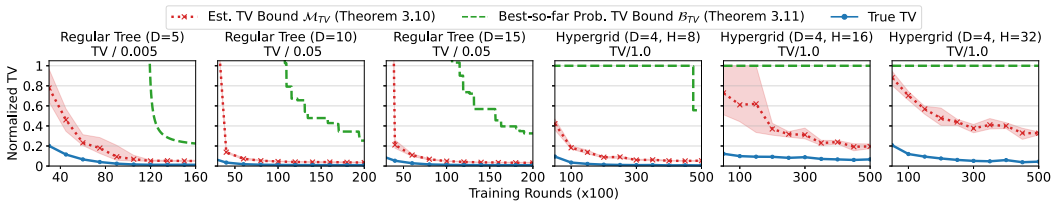

Figure 4: Derived TV bounds vs. true TV error. $\mathcal{M}_{\mathrm{TV}}$ follows the magnitude and trend of the observed TV error, providing an informative monitoring signal throughout training, whereas $\mathcal{B}_{\mathrm{TV}}$ provides a formal but more conservative probabilistic certificate. Failure would appear as an uncorrelated or oppositely varying trend, or as $\mathcal{B}_{\mathrm{TV}}$ underestimating the true TV error more frequently than permitted by its confidence level. Shaded regions show the min-max range over 5 independent trials of one trained policy, with $\mathcal{M}_{\mathrm{TV}}$ estimated using 100 backward-sampled trajectories per trial. $\mathcal{B}_{\mathrm{TV}}$ uses the accumulation procedure during training.

Table 3: Probabilistic certification and monitoring performance across environments. The best $\mathcal{B}_{\mathrm{TV}}$ is the smallest probabilistic TV certificate across 10 evenly spaced training checkpoints (10% to 100%). Monitoring performance is assessed by the correlation of $\mathcal{M}_{\mathrm{TV}}$ and FCS [Silva et al., 2025a] with global true TV across checkpoints. For calibration, a linear mapping from $\mathcal{M}_{\mathrm{TV}}$ to true TV is fitted on the first five checkpoints and evaluated by predictive RMSE on the remaining five; the RMSE of FCS is computed on the same held-out checkpoints.

Environment	Best $\mathcal{B}_{\mathrm{TV}}$	Corr.($\mathcal{M}_{\mathrm{TV}}$, true TV)	Corr.(FCS, true TV)	RMSE of calibrated $\mathcal{M}_{\mathrm{TV}}$	RMSE of FCS
RegularTree $D = 5$	0.0747	0.999	0.996	2.85×10^{-5}	1.24×10^{-5}
RegularTree $D = 10$	0.0802	0.959	0.999	0.0004	1.48×10^{-5}
RegularTree $D = 15$	0.0812	0.915	1.000	0.0009	1.89×10^{-5}
Hypergrid $D = 4,\ H = 8$	0.2542	0.997	0.996	0.0033	0.0012
Hypergrid $D = 4,\ H = 16$	1.0000	0.969	0.931	0.0088	0.0397
Hypergrid $D = 4,\ H = 32$	1.0000	0.977	0.980	0.0068	0.0044
L14-RNA1	1.0000	0.930	0.628	0.0231	0.0302

sampling accounts for most of the improvement, while Stable yields more consistent performance across seeds. On the substantially larger sEH task with much sparser rewards, however, stabilization provides its largest empirical gain: Stable discovers $14,142.6$ scaffolds, compared with $8,680.2$ for TB with backward sampling, corresponding to a 63% improvement.

5.3 Interpreting Global and Subgraph TV Certificates (RQ3)

Table 3 shows that $\mathcal{B}_{\mathrm{TV}}$ provides nontrivial probabilistic certificates on Regular Tree and the smaller Hypergrid setting. In contrast, $\mathcal{M}_{\mathrm{TV}}$ correlates strongly with the reported TV values across checkpoints. After linear calibration, it achieves lower held-out RMSE than FCS on Hypergrid with $H = 16$ and L14-RNA1, whereas FCS has lower RMSE in the remaining settings. However, FCS requires the probabilities that the model assigns to individual terminal states, whereas $\mathcal{M}_{\mathrm{TV}}$ uses only backward-sampled trajectories and does not require these probabilities. We also evaluate FCS using Monte Carlo estimates of the terminal state probabilities instead of their exact values; see Table 8 in Appendix F.3.

6 Conclusion and Discussion

We establish loss-to-TV guarantees and finite-sample TV certificates. Building on these theoretical foundations, we introduce Stable GFlowNets, which adapt reference flows to stabilize training while supporting TV-based monitoring and certification. Despite this progress, our guarantees currently rely on backward sampling, which could be difficult in continuous settings, and the probabilistic certificate can be overly conservative due to its worst-case dependence on the reference flow. Although the certificate remains theoretically valid and may become nonvacuous as training converges, developing tighter and more practically informative certificates remains an important direction for future work. Other directions include extending Theorem 3.6 to continuous state settings and exploring latent representations as alternative stabilization mechanisms with comparable guarantees.

Acknowledgment

This work is based upon the work supported by the National Center for Transportation Cybersecurity and Resiliency (TraCR) (a U.S. Department of Transportation National University Transportation Center) headquartered at Clemson University, Clemson, South Carolina, USA. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of TraCR, and the U.S. Government assumes no liability for the contents or use thereof. The authors declare no competing interests.

References

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. In *Advances in Neural Information Processing Systems*, 2021.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with GFlowNets. In *International Conference on Machine Learning*, 2022.

Tony Shen, Seonghwan Seo, Grayson Lee, Mohit Pandey, Jason R Smith, Artem Cherkasov, Woo Youn Kim, and Martin Ester. Tacogfn: Target-conditioned gflownet for structure-based drug design. In *Transactions on Machine Learning Research*, 2024.

Dinghuai Zhang, Hanjun Dai, Nikolay Malkin, Aaron C Courville, Yoshua Bengio, and Ling Pan. Let the flows tell: Solving graph combinatorial problems with GFlowNets. In *Advances in Neural Information Processing Systems*, 2023.

Salem Lahlou, Tristan Deleu, Pablo Lemos, Dinghuai Zhang, Alexandra Volokhova, Alex Hernández-García, Léna Néhale Ezzine, Yoshua Bengio, and Nikolay Malkin. A theory of continuous generative flow networks. In *International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2023a.

Kanika Madan, Alex Lamb, Emmanuel Bengio, Glen Berseth, and Yoshua Bengio. Towards improving exploration through sibling augmented GFlowNets. In *International Conference on Learning Representations*, 2025.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in GFlowNets. In *Advances in Neural Information Processing Systems*, 2022a.

Ling Pan, Dinghuai Zhang, Aaron Courville, Longbo Huang, and Yoshua Bengio. Generative augmented flow networks. In *International Conference on Learning Representations*, 2023.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF conference on computer vision and pattern recognition*, 2022.

Satinder P Singh and Richard C Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 1994.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 2021.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 2023.

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*, 2023.

Tiago Silva, Rodrigo Barreto Alves, Eliezer de Souza da Silva, Amauri H Souza, Vikas Garg, Samuel Kaski, and Diego Mesquita. When do gflownets learn the right distribution? In *International Conference on Learning Representations*, 2025a.

Nikolay Malkin, Salem Lahlou, Tristan Deleu, Xu Ji, Edward Hu, Katie Everett, Dinghuai Zhang, and Yoshua Bengio. Gflownets and variational inference. In *arXiv preprint arXiv:2210.00580*, 2022b.

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

Anas Krichel, Nikolay Malkin, Salem Lahlou, and Yoshua Bengio. On generalization for generative flow networks. *arXiv preprint arXiv:2407.03105*, 2024.

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

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, 2015.

Salem Lahlou, Joseph D Viviano, Victor Schmidt, and Yoshua Bengio. torchgfn: A Pytorch GFlowNet library. In *arXiv preprint arXiv:2305.14594*, 2023b.

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*, 2023.

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

Sam Sinai, Richard Wang, Alexander Whatley, Stewart Slocum, Elina Locane, and Eric D Kelsic. Adalead: A simple and robust adaptive greedy search algorithm for sequence design. 2020.

Gerald J Hahn and William Q Meeker. *Statistical intervals: a guide for practitioners*, volume 92. John Wiley & Sons, 2011.

Minsu Kim, Joohwan Ko, Taeyoung Yun, Dinghuai Zhang, Ling Pan, Woochang Kim, Jinkyoo Park, Emmanuel Bengio, and Yoshua Bengio. Learning to scale logits for temperature-conditional GFlowNets. In *International Conference on Machine Learning*, 2024b.

Yihang Chen and Lukas Mauch. Order-preserving GFlowNets. In *International Conference on Learning Representations*, 2024.

Dinghuai Zhang, Ling Pan, Ricky TQ Chen, Aaron Courville, and Yoshua Bengio. Distributional GFlowNets with quantile flows. In *Transactions on Machine Learning Research*, 2024.

Tiago Silva, Eliezer de Souza da Silva, and Diego Mesquita. On divergence measures for training gflownets. *Advances in Neural Information Processing Systems*, 2024.

Rui Hu, Yifan Zhang, Zhuoran Li, and Longbo Huang. Beyond squared error: Exploring loss design for enhanced training of generative flow networks. In *International Conference on Learning Representations*, 2025.

Elaine Lau, Nikhil Vemgal, Doina Precup, and Emmanuel Bengio. DGFN: Double generative flow networks. In *NeurIPS Workshop on Generative AI and Biology*, 2023.

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

Zarif Ikram, Ling Pan, and Dianbo Liu. Evolution guided generative flow networks. In *Transactions on Machine Learning Research*, 2025.

Hyeonah Kim, Minsu Kim, Sanghyeok Choi, and Jinkyoo Park. Genetic-guided gflownets for sample efficient molecular optimization. *Advances in Neural Information Processing Systems*, 2024c.

Minsu Kim, Sanghyeok Choi, Hyeonah Kim, Jiwoo Son, Jinkyoo Park, and Yoshua Bengio. Ant colony sampling with GFlowNets for combinatorial optimization. In *International Conference on Artificial Intelligence and Statistics*, 2025b.

Hyosoon Jang, Yunhui Jang, Minsu Kim, Jinkyoo Park, and Sungsoo Ahn. Pessimistic backward policy for GFlowNets. In *Neural Information Processing Systems*, 2024.

Tri Minh Nguyen, Sherif Abdulkader Tawfik, Truyen Tran, Sunil Gupta, Santu Rana, and Svetha Venkatesh. Hierarchical gflownet for crystal structure generation. In *AI for Accelerated Materials Design-NeurIPS 2023 Workshop*, 2023.

Ling Pan, Moksh Jain, Kanika Madan, and Yoshua Bengio. Pre-training and fine-tuning generative flow networks. In *International Conference on Learning Representations*, 2024.

Haoran He, Can Chang, Huazhe Xu, and Ling Pan. Looking backward: Retrospective backward synthesis for goal-conditioned GFlowNets. In *International Conference on Learning Representations*, 2025.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *arXiv preprint arXiv:1509.02971*, 2015.

A Notation

Symbol	Definition
$\mathcal{S}, s_0, s_f$	Set of all states, initial state, and final state.
$\mathcal{A}$	Set of all actions (edges); $s \to s' \in \mathcal{A}$ indicates a transition with positive support.
$\mathcal{X}$	Set of terminal states, i.e., states which can only transition to s_f.
$\mathcal{T}$	Set of complete trajectories (i.e., starting at s_0 and ending at s_f) with positive support.
$F(s), F(\tau)$	Total flow through a state s or a trajectory τ.
$F(s \to s')$	Total flow through an edge $s \to s'$.
P_F	Forward policy, assigning probabilities to edges or trajectories; for $\tau = (s_0, s_1, \ldots, s_n, s_f)$, $P_F(\tau) = \prod_{i=0}^{n-1} P_F(s_i, s_{i+1})$.
P_B	Backward policy, assigning probabilities to backward transitions or trajectories; for $\tau = (s_0, \ldots, s_f)$, $P_B(\tau \mid s_n) = \prod_{i=0}^{n-1} P_B(s_i, s_{i+1})$.
$R(x), R(\tau)$	Reward function over a terminal state x or a trajectory τ; $R(\tau) = R(x)P_B(\tau \mid x)$.
Z	Estimated partition function.
Z^*	True partition function, equal to the total reward over terminal states.
P_T	Terminal-state distribution induced by P_F.
π_{target}	Ground-truth reward distribution normalized by Z^*.

Table 4: Notation and Definitions

B Proof of Propositions and Theorems

Proposition 3.3. *(TV bound for incremental mode coverage over arbitrary state graph). Let the reward added at each state in a subset $\mathcal{X}_{sub}$ be $R'(x)$. We define the local true partition function $Z^*_{\mathcal{Y}\subseteq\mathcal{X}} = \sum_{x\in\mathcal{Y}} R(x)$ and the local contrast ratio $\Lambda_{\mathcal{Y}} = \frac{Z^*_{\mathcal{Y}}}{Z^*_{\mathcal{Y}} + \sum_{x\in\mathcal{Y}} R'(x)}$. Then, we have*

$$\frac{Z^* - Z^*_{\mathcal{X}_{sub}}}{Z^*}(1 - \Lambda_{\mathcal{X}}) \leq \mathrm{TV}(P_T, \pi_{target}) \leq (1 - \Lambda_{\mathcal{X}})$$

Proof. To prove the upper bound, we use the standard definition of the Total Variation distance:

$$\mathrm{TV}(P, Q) = \frac{1}{2}\sum_x |P(x) - Q(x)|$$

Substitute the expressions for the policies:

$$P_T(x) - \pi_{target}(x) = \frac{R(x)}{Z^*} - \frac{R(x) + R'(x)}{Z^* + \sum_{x\in\mathcal{X}} R'(x)}$$

Substitute $Z^* + \sum_{x\in\mathcal{X}} R'(x) = Z^*/\Lambda_{\mathcal{X}}$:

$$P_T(x) - \pi_{target}(x) = \frac{(1 - \Lambda_{\mathcal{X}})R(x) - \Lambda_{\mathcal{X}} R'(x)}{Z^*}$$

Plug this back, we have

$$\begin{aligned}
2\mathrm{TV}(P_T, \pi_{target}) &= \sum_{x\in\mathcal{X}} \left| \frac{(1-\Lambda_{\mathcal{X}})R(x)}{Z^*} - \frac{\Lambda_{\mathcal{X}} R'(x)}{Z^*} \right| \\
&\leq \sum_{x\in\mathcal{X}} \frac{(1-\Lambda_{\mathcal{X}})R(x)}{Z^*} + \sum_{x\in\mathcal{X}} \frac{\Lambda_{\mathcal{X}} R'(x)}{Z^*} \\
&= (1-\Lambda_{\mathcal{X}}) + \frac{\sum_{x\in\mathcal{X}} R'(x)}{Z^* + \sum_{x\in\mathcal{X}} R'(x)} \\
&= 2(1-\Lambda_{\mathcal{X}})
\end{aligned}$$

Divide both sides by 2, we get:

$$\mathrm{TV}(P_T, \pi_{target}) \leq (1-\Lambda_{\mathcal{X}})$$

To prove the lower bound, we use the alternative definition of TV distance:

$$\mathrm{TV}(P, Q) = \sup_{X\subseteq\mathcal{X}} |P(X) - Q(X)|$$

This means the difference in probability mass on any specific subset X is a lower bound for the TV distance.

Let us choose the subset to be the unchanged region $\mathcal{X}_{unc} = \mathcal{X} \setminus \mathcal{X}_{sub}$. For any state $x \in \mathcal{X}_{unc}$, $R'(x) = 0$.

We have

$$P_T(\mathcal{X}_{unc}) = \frac{\sum_{x\notin\mathcal{X}_{sub}} R(x)}{Z^*} = \frac{Z^* - Z^*_{\mathcal{X}_{sub}}}{Z^*}$$

$$\begin{aligned}
\pi_{target}(\mathcal{X}_{unc}) &= \frac{Z^* - Z^*_{\mathcal{X}_{sub}}}{Z^* + \sum_{x\in\mathcal{X}} R'(x)} \\
&= \frac{\Lambda_{\mathcal{X}}(Z^* - Z^*_{\mathcal{X}_{sub}})}{Z^*}
\end{aligned}$$

$$\begin{aligned}
\mathrm{TV}(P_T, \pi_{target}) &\geq |P_T(\mathcal{X}_{unc}) - \pi_{target}(\mathcal{X}_{unc})| \\
&= \frac{(Z^* - Z^*_{\mathcal{X}_{sub}})(1-\Lambda_{\mathcal{X}})}{Z^*}
\end{aligned}$$

□

Proposition 3.4. *(Loss scale via local contrast). While the TV error depends on the aggregate $\Lambda_{\mathcal{X}}$, the supremum of the training loss is governed by the* ***worst-case*** *local contrast ratio, given by:*

$$\sup \mathcal{L}_{GFN} = \left(\log \min_{\{x\}\subseteq\mathcal{X}_{sub}} \Lambda_{\{x\}} \right)^2$$

where $GFN \in \{FM, DB, TB, subTB\}$.

Proof. We assume the network parameters are currently at the optimum for the previous task. This means the current flow satisfies the balance equation for the old reward. Since only the rewards change and P_B is unchanged, substituting the pre-update state into the loss functions of the new task yields that all non-zero losses satisfy

$$\begin{aligned}
\mathcal{L}_{FM}(x) &= \mathcal{L}_{DB}(s, x) = \mathcal{L}_{TB}(\tau \ni x) \\
&= \mathcal{L}_{subTB}(\tau_{t_1:t_2} \ni x) = \left(\log \frac{R(x)}{R(x) + R'(x)} \right)^2 \\
&= \left(\log \Lambda_{\{x\}} \right)^2
\end{aligned}$$

The supremum of the loss over the entire state space is determined by the state x that maximizes this squared log term. Since $\Lambda_{\{x\}} \leq 1$, we have

$$\sup|\mathcal{L}_{GFN}| = \left(\log \min_{\{x\}\subseteq\mathcal{X}_{sub}} \Lambda_{\{x\}}\right)^2$$

□

Theorem 3.5. *(Training loss to TV distance bound). The relationship between the training loss bound and the resulting TV distance depends on the scope of the objective (trajectory-level vs. transition-level):*

Trajectory-level Objective. *If the trajectory loss is bounded by $\mathcal{L}_{TB}(\tau) \leq c^2, \forall\tau \in \mathcal{T}$, the TV error is bounded by:*

$$\mathrm{TV}(P_T, \pi_{target}) \leq 1 - e^{-2c}$$

This bound is independent of the trajectory length, as TB optimizes the full path consistency directly.

Transition-level Objective. *If the local transition loss is bounded, i.e., $\mathcal{L}_{DB}(s, s') \leq c^2$ or $\mathcal{L}_{FM}(s') \leq c^2$, the global consistency relies on the accumulation of local estimates. For trajectories of maximum number of transitions L, the error bound degrades linearly with depth in the log-domain:*

$$\mathrm{TV}(P_T, \pi_{target}) \leq 1 - e^{-2Lc}$$

Proof. We first prove the trajectory-level bound and then derive the DB and FM cases by bounding the corresponding trajectory log-ratios.

Let Z be the learnable partition function. When TB loss is bounded by c^2, then $\forall\tau$ with $x \in \tau$ and $x \in \mathcal{X}$, we have:

$$e^{-c} \leq \frac{ZP_F(\tau)}{R(x)P_B(\tau \mid x)} \leq e^c$$

Aggregate this inequality over all possible trajectories in the entire space $\mathcal{T}$, we have

$$e^{-c} \leq \frac{Z}{Z^*} \leq e^c$$
$$Z^*e^{-c} \leq Z \leq Z^*e^c$$

Also from the TB definition, we have

$$\begin{aligned} P_F(\tau) &\geq e^{-c}\frac{R(x)P_B(\tau \mid x)}{Z} \\ &\geq e^{-c}\frac{R(x)P_B(\tau \mid x)}{e^cZ^*} \\ &\geq e^{-2c}\frac{R(x)P_B(\tau \mid x)}{Z^*} \end{aligned}$$

Sum over all trajectories that lead to x, we have

$$P_T(x) \geq e^{-2c}\frac{R(x)}{Z^*} = e^{-2c}\pi_{target}(x)$$

We use another alternative definition of TV distance:

$$\mathrm{TV}(P, Q) = 1 - \sum_{x\in\mathcal{X}} \min(P, Q)$$

If $P_T(x) \geq \pi_{target}(x)$, then $\min(P_T(x), \pi_{target}(x)) = \pi_{target}(x)$. Since $e^{-2c} \leq 1$, $\pi_{target}(x) \geq e^{-2c}\pi_{target}(x)$. If $P_T(x) < \pi_{target}(x)$, then $\min(P_T(x), \pi_{target}(x)) = P_T(x) \geq e^{-2c}\pi_{target}(x)$.

In both cases, we have

$$\min(P_T, \pi_{target}) \geq e^{-2c}\pi_{target}(x)$$

Substitute into the TV equation:

$$\begin{aligned} \mathrm{TV}(P_T, \pi_{target}) &\leq 1 - \sum_{x\in\mathcal{X}} e^{-2c}\pi_{target(x)} \\ &= 1 - e^{-2c} \end{aligned}$$

Extension to DB and FM. Consider a complete trajectory $\tau = (s_0 \to s_1 \to \cdots \to s_n = x \to s_f)$, where $n \le L$. Under our convention, $P_F(x, s_f) = 1$, so

$$P_F(\tau) = \prod_{t=0}^{n-1} P_F(s_t, s_{t+1}), \qquad P_B(\tau \mid x) = \prod_{t=0}^{n-1} P_B(s_t, s_{t+1}).$$

In the rest of the proof, $F(s)$ and $F(s, s')$ denote the positive estimates used in the respective training objectives; exact flow conservation is not assumed.

Detailed balance. Take $Z = F(s_0)$ and impose $F(x) = R(x)$ at each terminal state. Define the signed DB residual

$$r_{\mathrm{DB}}(s, s') := \log \frac{F(s)P_F(s, s')}{F(s')P_B(s, s')}.$$

Because $\mathcal{L}_{\mathrm{DB}}(s, s') = r_{\mathrm{DB}}(s, s')^2 \le c^2$, we have $|r_{\mathrm{DB}}(s, s')| \le c$. The state-flow terms telescope along any trajectory:

$$\begin{aligned}
\sum_{t=0}^{n-1} r_{\mathrm{DB}}(s_t, s_{t+1}) &= \log \prod_{t=0}^{n-1} \frac{F(s_t)P_F(s_t, s_{t+1})}{F(s_{t+1})P_B(s_t, s_{t+1})} \\
&= \log \frac{F(s_0)P_F(\tau)}{F(s_n)P_B(\tau \mid x)} \\
&= \log \frac{ZP_F(\tau)}{R(x)P_B(\tau \mid x)}.
\end{aligned}$$

Consequently,

$$\left| \log \frac{ZP_F(\tau)}{R(x)P_B(\tau \mid x)} \right| \le \sum_{t=0}^{n-1} |r_{\mathrm{DB}}(s_t, s_{t+1})| \le nc \le Lc.$$

Thus $\mathcal{L}_{\mathrm{TB}}(\tau) \le L^2 c^2$ for every $\tau \in \mathcal{T}$.

Flow matching. Let $F(s, s') > 0$ be the learned edge-flow estimates on transitions with $s' \neq s_f$, and fix $F(x, s_f) = R(x)$ for $x \in \mathcal{X}$. Define the incoming and outgoing flow sums by

$$I(s) := \sum_{u:\, u \to s \in \mathcal{A}} F(u, s), \qquad s \notin \{s_0, s_f\},$$

$$O(s) := R(s) + \sum_{v:\, s \to v \in \mathcal{A},\ v \neq s_f} F(s, v), \qquad s \neq s_f.$$

In particular, $O(x) = R(x)$ for each terminal state. Set $Z = O(s_0)$ and define, for $s \to s' \in \mathcal{A}$ with $s' \neq s_f$,

$$P_F(s, s') = \frac{F(s, s')}{O(s)}, \qquad P_B(s, s') = \frac{F(s, s')}{I(s')}.$$

Together with $P_F(x, s_f) = 1$, these are normalized forward and backward policies. The backward policy P_B is an auxiliary construction for the proof; it need not be separately learned. Importantly, $I(s)$ and $O(s)$ are not assumed equal.

The signed FM residual is

$$r_{\mathrm{FM}}(s) := \log \frac{I(s)}{O(s)}, \qquad \mathcal{L}_{\mathrm{FM}}(s) = r_{\mathrm{FM}}(s)^2.$$

For each nonfinal transition, the common edge-flow estimate cancels:

$$\frac{P_F(s, s')}{P_B(s, s')} = \frac{I(s')}{O(s)}.$$

Therefore, for any complete trajectory ending at $x = s_n$,

$$\frac{ZP_F(\tau)}{R(x)P_B(\tau \mid x)} = \frac{O(s_0)}{O(s_n)} \prod_{t=0}^{n-1} \frac{I(s_{t+1})}{O(s_t)}$$
$$= \prod_{t=1}^{n} \frac{I(s_t)}{O(s_t)}.$$

Taking logarithms gives the required telescoping identity:

$$\log \frac{ZP_F(\tau)}{R(x)P_B(\tau \mid x)} = \sum_{t=1}^{n} r_{\mathrm{FM}}(s_t).$$

Since the FM bound applies to every visited noninitial state, including $s_n = x$, it follows that

$$\left| \log \frac{ZP_F(\tau)}{R(x)P_B(\tau \mid x)} \right| \leq \sum_{t=1}^{n} |r_{\mathrm{FM}}(s_t)| \leq nc \leq Lc.$$

Hence the corresponding TB loss, evaluated with $Z = O(s_0)$ and P_B, is at most L^2c^2 on every trajectory.

□

Theorem 3.6. *(Probabilistic TV bound via trajectory sampling). Given the ground-truth target distribution π_{target}, we define a target distribution over trajectories $\hat{\pi}(\tau) = \pi_{target}(x_\tau)P_B(\tau|x_\tau)$. Sample m trajectories $\tau_1, \ldots, \tau_m$ from $\hat{\pi}$ independently by sampling $x \sim \pi_{target}$ and $\tau \sim P_B(\cdot|x)$. Sample another n trajectories independently using P_F. Let $c = \max_{i \leq m+n} \sqrt{\mathcal{L}_{TB}(\tau_i)}$, with confidence $1 - 2\alpha$, the global TV error is bounded by:*

$$\mathrm{TV}(P_T, \pi_{target}) \leq e^{2c} + 1 - \alpha^{\frac{1}{m}} - \alpha^{\frac{1}{n}} \leq e^{2c} - 1 + \frac{\log(1/\alpha)}{m} + \frac{\log(1/\alpha)}{n}$$

Proof. Let $P_F(\tau)$ be the forward policy's probability of trajectory τ. Let $\hat{\pi}(\tau)$ be the target trajectory distribution induced by the reward and backward policy:

$$\hat{\pi}(\tau) = \frac{R(x)}{Z^*} P_B(\tau|x) = \pi_{target}(x)P_B(\tau|x)$$

Note that the marginal of $\hat{\pi}$ over states is exactly the target distribution: $\sum_{\tau \to x} \hat{\pi}(\tau) = \pi_{target}(x)$.

Define the "good" set of trajectories $G_{\mathcal{T}} = \{\tau \in \mathcal{T} \mid \mathcal{L}_{TB}(\tau) \leq c^2\}$ and its probability mass measured under target trajectory distribution $\hat{\pi}(G_{\mathcal{T}})$ as $1 - \epsilon_c$.

The one-sided distribution-free tolerance bound [Hahn and Meeker, 2011] gives, for any $\xi \in (0, 1)$

$$Pr(\epsilon_c > \xi) \leq (1 - \xi)^m$$

Setting $\xi = 1 - \alpha^{1/m}$, we get: with probability at least $1 - \alpha$,

$$\epsilon_c \leq 1 - \alpha^{1/m} = 1 - e^{-\log(1/\alpha)/m} \leq \frac{\log(1/\alpha)}{m}$$

Similarly, let $P_F(G_{\mathcal{T}}) = 1 - \eta_c$, since we observed n independent samples from P_F and all fell into $G_{\mathcal{T}}$, we get: with confidence $1 - \alpha$:

$$\eta_c \leq 1 - \alpha^{1/n} \leq \frac{\log(1/\alpha)}{n}$$

By the union bound, we have $\epsilon_c \leq 1 - \alpha^{1/m}$ and $\eta_c \leq 1 - \alpha^{1/n}$ together hold with confidence $1 - 2\alpha$.

For any good trajectory $\tau \in G_{\mathcal{T}}$, the condition $\mathcal{L}_{TB}(\tau) \leq c^2$ implies:

$$\left|\log \frac{ZP_F(\tau)}{R(x)P_B(\tau|x)}\right| \leq c$$

$$e^{-c} \leq \frac{KP_F(\tau)}{\hat{\pi}(\tau)} \leq e^c$$

where $K = \frac{Z}{Z^*}$.

Now we derive the bound for K, note

$$K \sum_{\tau \in G_{\mathcal{T}}} P_F(\tau) \geq \sum_{\tau \in G_{\mathcal{T}}} e^{-c}\hat{\pi}(\tau) = e^{-c}(1-\epsilon_c)$$

Since $\sum_{\tau \in G_{\mathcal{T}}} P_F(\tau) \leq 1$, we have

$$K \geq e^{-c}(1-\epsilon_c)$$

So $\frac{P_F(\tau)}{\hat{\pi}(\tau)}$ satisfies

$$\frac{P_F(\tau)}{\hat{\pi}(\tau)} \leq \frac{1}{1-\epsilon_c}e^{2c}$$

The TV distance between the two distributions over trajectories is bounded by:

$$2\mathrm{TV}(P_F, \hat{\pi}) = \sum_{\tau \in G_{\mathcal{T}}} |P_F(\tau) - \hat{\pi}(\tau)| + \sum_{\tau \in G_{\mathcal{T}}^{\complement}} |P_F(\tau) - \hat{\pi}(\tau)|$$

For the first term $\sum_{\tau \in G_{\mathcal{T}}} |P_F(\tau) - \hat{\pi}(\tau)|$, we split the good set $G_{\mathcal{T}}$ into $G_{\mathcal{T}}^+$ and $G_{\mathcal{T}}^-$ such that

$$P_F(\tau) \geq \hat{\pi}(\tau), \forall \tau \in G_{\mathcal{T}}^+$$

$$P_F(\tau) < \hat{\pi}(\tau), \forall \tau \in G_{\mathcal{T}}^-$$

We have

$$\begin{aligned}
&\sum_{\tau \in G_{\mathcal{T}}} |P_F(\tau) - \hat{\pi}(\tau)| \\
=&\sum_{\tau \in G_{\mathcal{T}}^+} (P_F(\tau) - \hat{\pi}(\tau)) + \sum_{\tau \in G_{\mathcal{T}}^-} (\hat{\pi}(\tau) - P_F(\tau)) \\
=&\sum_{\tau \in G_{\mathcal{T}}^+} (P_F(\tau) - \hat{\pi}(\tau)) + (1-\epsilon_c) - \sum_{\tau \in G_{\mathcal{T}}^+} \hat{\pi}(\tau) \\
&- (1-\eta_c) + \sum_{\tau \in G_{\mathcal{T}}^+} P_F(\tau) \\
=&2 \sum_{\tau \in G_{\mathcal{T}}^+} (P_F(\tau) - \hat{\pi}(\tau)) + \eta_c - \epsilon_c \\
=&2 \sum_{\tau \in G_{\mathcal{T}}^+} \hat{\pi}(\tau)(\frac{P_F(\tau)}{\hat{\pi}(\tau)} - 1) + \eta_c - \epsilon_c \\
\leq&2(1-\epsilon_c)(\frac{1}{1-\epsilon_c}e^{2c} - 1) + \eta_c - \epsilon_c \\
=&2(e^{2c} - 1) + \eta_c + \epsilon_c
\end{aligned}$$

For the second term $\sum_{\tau \in G_{\mathcal{T}}^{\complement}} |P_F(\tau) - \hat{\pi}(\tau)|$, we directly use the triangle inequality and get

$$\begin{aligned}\sum_{\tau \in G_{\mathcal{T}}^{\complement}} |P_F(\tau) - \hat{\pi}(\tau)| &\leq \sum_{\tau \in G_{\mathcal{T}}^{\complement}} P_F(\tau) + \sum_{\tau \in G_{\mathcal{T}}^{\complement}} \hat{\pi}(\tau) \\ &= \eta_c + \epsilon_c\end{aligned}$$

Combine the first term and the second term, we have

$$\mathrm{TV}(P_F, \hat{\pi}) \leq (e^{2c} - 1) + \epsilon_c + \eta_c$$

For the TV distance between marginal distributions, we have

$$\mathrm{TV}(P_T, \pi_{target}) = \frac{1}{2} \sum_x |P_T(x) - \pi_{target}(x)|$$

Since $P_T(x) = \sum_{\tau \to x} P_F(\tau)$ and $\pi_{target}(x) = \sum_{\tau \to x} \hat{\pi}(\tau)$:

$$\begin{aligned}|P_T(x) - \pi_{target}(x)| &= \left| \sum_{\tau \to x} (P_F(\tau) - \hat{\pi}(\tau)) \right| \\ &\leq \sum_{\tau \to x} |P_F(\tau) - \hat{\pi}(\tau)|\end{aligned}$$

Summing over all x:

$$\begin{aligned}\mathrm{TV}(P_T, \pi_{target}) &\leq \frac{1}{2} \sum_x \sum_{\tau \to x} |P_F(\tau) - \hat{\pi}(\tau)| \\ &= \mathrm{TV}(P_F, \hat{\pi}) \\ &\leq (e^{2c} - 1) + \epsilon_c + \eta_c \\ &\leq (e^{2c} - 1) + 1 - \alpha^{1/m} + 1 - \alpha^{1/n} \\ &= e^{2c} + 1 - \alpha^{1/m} - \alpha^{1/n} \\ &\leq (e^{2c} - 1) + \frac{\log(1/\alpha)}{m} + \frac{\log(1/\alpha)}{n}\end{aligned}$$

□

Corollary 3.7. *(Subgraph Certification via trajectory sampling). Let $\mathcal{X}_{sub} \subseteq \mathcal{X}$ be a subset of end-states. Define the restricted target distribution over $\mathcal{X}_{sub}$ by $\pi_{target}^{sub}(x) = R(x) / \sum_{x \in \mathcal{X}_{sub}} R(x)$, the corresponding restricted target trajectory distribution as $\hat{\pi}_{sub}(\tau) = \pi_{target}^{sub}(x_\tau) P_B(\tau | x_\tau)$. Sample m trajectories $\tau_1, \ldots, \tau_m$ from $\hat{\pi}_{sub}$. Sample another n trajectories that end within $\mathcal{X}_{sub}$ independently using P_F. Suppose we observe $\mathcal{L}_{TB}(\tau_i) \leq c^2$ for all trajectories in both sets. Let P_T^{sub} denote the terminal flow P_T renormalized to $\mathcal{X}_{\mathrm{sub}}$. Then, with confidence $1 - 2\alpha$,*

$$\mathrm{TV}(P_T^{sub}, \pi_{target}^{sub}) \leq e^{2c} + 1 - \alpha^{\frac{1}{m}} - \alpha^{\frac{1}{n}} \leq e^{2c} - 1 + \frac{\log(1/\alpha)}{m} + \frac{\log(1/\alpha)}{n}$$

Proof. To prove this corollary, we first develop a mapping from the global forward policy P_F and partition function Z to the subgraph restricted counterparts P_F^{sub} and Z_{sub}.

Consider the subgraph induced by $\mathcal{X}_{sub}$. We have the partition function for the subgraph

$$Z_{sub} = \sum_{x \in \mathcal{X}_{sub}} F(x) = Z \sum_{x \in \mathcal{X}_{sub}} P_T(x)$$

The backward policy over the subgraph yields the same value as the global one, i.e., $P_B^{sub} = P_B$. The forward policy over this subgraph P_F^{sub} can be represented as the forward policy conditioned on the trajectory ending in the subgraph:

$$P_F^{sub}(\tau) = P_F(\tau \mid x_\tau \in \mathcal{X}_{sub}) = \frac{P_F(\tau)}{\sum_{x \in \mathcal{X}_{sub}} P_T(x)}$$

Now we map the TB loss to the subgraph context, we have the TB loss over the subgraph become:

$$\begin{aligned}
\mathcal{L}_{TB}^{sub}(\tau) &= \left(\log \frac{Z_{sub} P_F^{sub}(\tau)}{R(x) P_B^{sub}(\tau|x)}\right)^2 \\
&= \left(\log \frac{Z P_F(\tau) \sum_{x \in \mathcal{X}_{sub}} P_T(x)}{R(x) P_B(\tau|x) \sum_{x \in \mathcal{X}_{sub}} P_T(x)}\right)^2 \\
&= \left(\log \frac{Z P_F(\tau)}{R(x) P_B(\tau|x)}\right)^2 \leq c^2
\end{aligned}$$

Since all preconditions of Theorem 3.6 are satisfied for the subgraph, we apply it directly to obtain the desired bound.

□

Theorem 3.10. *(Fidelity trade-off under reference flow). The fidelity of the recovered policy depends on the ratio between the training loss and the augmentation magnitude. Let the total reference flow be $\Delta = \sum_{\tau \in \mathcal{T}} \delta(\tau)$. If the reference training loss is bounded by $\mathcal{L}_{aug}(\tau) \leq c^2$, the terminal distribution P_T induced by the learned forward policy satisfies:*

$$\mathrm{TV}(P_T, \pi_{target}) \leq \frac{(1 - e^{-2c})(1 + \Delta/Z^*)}{1 + (1 - e^{-c})\Delta/Z^*} \leq (1 - e^{-2c})(1 + \frac{\Delta}{Z^*})$$

Proof. Define $\delta(x) = \sum_{\tau \ni x} \delta(\tau)$, note $\Delta = \sum_{x \in \mathcal{X}} \delta(x)$. We have the target distribution under reference flows as:

$$\pi_{aug}(x) = \frac{R(x) + \delta(x)}{Z^* + \Delta}$$

From $\mathcal{L}_{\mathrm{aug}}(\tau) \leq c^2$

$$e^{-c}\left(R(\tau) + \delta(\tau)\right) \leq F(\tau) + \delta(\tau) \leq e^{c}\left(R(\tau) + \delta(\tau)\right).$$

Summing the upper inequality over all trajectories gives

$$Z + \Delta \leq e^{c}\left(Z^* + \Delta\right)$$

and therefore

$$Z \leq Z^* \left[e^{c} + (e^{c} - 1)\,\frac{\Delta}{Z^*}\right]$$

Summing the lower inequality over trajectories terminal at x gives

$$F(x) + \delta(x) \geq e^{-c}\left(R(x) + \delta(x)\right)$$

Since $F(x) \geq 0$,

$$F(x) \geq R(x) \left[e^{-c} - \left(1 - e^{-c}\right) \frac{\delta(x)}{R(x)}\right]_+$$

From $P_T(x) := \frac{F(x)}{Z}$,

$$P_T(x) \geq \pi_{target}(x) \frac{[e^{-c} - (1 - e^{-c})\, \delta(x)/R(x)]_+}{e^c + (e^c - 1)\, \Delta/Z^*}$$

Because $\frac{\left[e^{-c} - \left(1 - e^{-c}\right)\delta(x)/R(x)\right]_+}{e^c + (e^c - 1)\Delta/Z^*} \leq 1$,

$$\min\left(P_T(x), \pi_{target}(x)\right) \geq \pi_{target}(x) \frac{[e^{-c} - (1 - e^{-c})\, \delta(x)/R(x)]_+}{e^c + (e^c - 1)\, \Delta/Z^*}$$

Summing over x and applying Jensen's inequality to the convex positive-part function gives

$$\sum_{x \in \mathcal{X}} \min\left(P_T(x), \pi_{target}(x)\right) \geq \frac{\left[e^{-c} - (1 - e^{-c}) \sum_{x \in \mathcal{X}} \pi_{target}(x) \delta(x)/R(x)\right]_+}{e^c + (e^c - 1)\, \Delta/Z^*}$$

Moreover,

$$\sum_{x \in \mathcal{X}} \pi_{target}(x) \frac{\delta(x)}{R(x)} = \sum_{x \in \mathcal{X}} \frac{R(x)}{Z^*} \frac{\delta(x)}{R(x)} = \frac{1}{Z^*} \sum_{x \in \mathcal{X}} \delta(x) = \frac{\Delta}{Z^*}$$

Therefore,

$$\sum_{x \in \mathcal{X}} \min\left(P_T(x), \pi_{target}(x)\right) \geq \frac{[e^{-c} - (1 - e^{-c})\, \Delta/Z^*]_+}{e^c + (e^c - 1)\, \Delta/Z^*}$$

Using

$$\mathrm{TV}\left(P_T, \pi_{target}\right) = 1 - \sum_{x \in \mathcal{X}} \min\left(P_T(x), \pi_{target}(x)\right)$$

we get

$$\mathrm{TV}\left(P_T, \pi_{target}\right) \leq \frac{\left(1 - e^{-2c}\right)\left(1 + \Delta/Z^*\right)}{1 + \left(1 - e^{-c}\right)\Delta/Z^*} \leq \left(1 - e^{-2c}\right)\left(1 + \frac{\Delta}{Z^*}\right)$$

□

Theorem 3.11. *(Probabilistic TV bound with optimizable reference-flow threshold). Sample $m(\geq 2)$ trajectories $\tau_1, \ldots, \tau_m$ independently from the target $\hat{\pi}$, sample another $n(\geq 2)$ trajectories $\tau_{m+1}, \ldots, \tau_{m+n}$ independently using P_F. For each $c > 0$, we compute the minimum reference flow $\delta_c(\tau_i)$ according to Equation (15). Define $M_c := \max_{i \in \{1,\ldots,m+n\}} \frac{\delta_c(\tau_i)}{R(\tau_i)}$, and $\mathcal{C} = \{c > 0 \mid M_c < \frac{1}{e^c - 1}\}$. With confidence $1 - 2\alpha$, the following bound holds simultaneously for every $c \in \mathcal{C}$:*

$$\begin{aligned}
\mathrm{TV}(P_T, \pi_{target}) &\leq \beta_m(\alpha) + \beta_n(\alpha) + \left(\frac{e^c + (e^c - 1)M_c}{e^{-c} - (1 - e^{-c})M_c} - 1\right) \\
&\leq \frac{2\log(2/\alpha)}{m} + \frac{2\log(2/\alpha)}{n} + \left(\frac{e^c + (e^c - 1)M_c}{e^{-c} - (1 - e^{-c})M_c} - 1\right)
\end{aligned}$$

where $\beta_k(\alpha)$ is solved from

$$(1 - \beta_k(\alpha))^{k-1}[1 + (k-1)\beta_k(\alpha)] = \alpha$$

Proof. Define $\rho(\tau) = \frac{ZP_F(\tau)}{R(\tau)}$, and the "$c$-dependent" good set of trajectories $G_{\mathcal{T}} \subseteq \mathcal{T}$ as those satisfying

$$G_{\mathcal{T}} = \{\tau \in \mathcal{T} \mid \frac{\delta_c(\tau)}{R(\tau)} \leq M_c\}$$

Equation (15) implies the good set

$$G_{\mathcal{T}} = \{\tau \in \mathcal{T} \mid \frac{\delta_c(\tau)}{R(\tau)} \leq M_c\} = \{\tau \in \mathcal{T} \mid L_c \leq \rho(\tau) \leq U_c\}$$

where $L_c = e^{-c} - (1 - e^{-c})M_c$ and $U_c = e^c + (e^c - 1)M_c$.

Next, we show that for every $c \in \mathcal{C}$, every c-dependent good set contains the empirical range of the samples. To see this, let $\rho_{min} = \min_{i \leq m+n} \frac{ZP_F(\tau_i)}{R(\tau_i)}$ and $\rho_{max} = \max_{i \leq m+n} \frac{ZP_F(\tau_i)}{R(\tau_i)}$. We then have

$$[\rho_{min}, \rho_{max}] \subseteq [L_c, U_c]$$

Hence, every c-dependent bad set (i.e., complementary to the good set), whether evaluated forward or backward, lies in the corresponding empirical-range complement, which all share a single order-statistic bound. Let ϵ_c be the probability mass of the bad set under the target trajectory distribution, η_c be the probability mass of the bad set under the forward policy, and $\beta_k(\alpha)$ solve

$$(1 - \beta_k(\alpha))^{k-1}[1 + (k-1)\beta_k(\alpha)] = \alpha$$

By the two-sided distribution-free tolerance bound [Hahn and Meeker, 2011] and the union bound, with confidence $1 - 2\alpha$, **simultaneously for every** $c \in \mathcal{C}$, one has

$$\epsilon_c \leq \beta_m(\alpha), \quad \eta_c \leq \beta_n(\alpha)$$

A computationally simpler but looser bound follows by controlling the tails separately. Assigning failure probability $\alpha/2$ to each tail gives, with confidence $1 - 2\alpha$, one has

$$\epsilon_c \leq \frac{2\log(2/\alpha)}{m}, \quad \eta_c \leq \frac{2\log(2/\alpha)}{n}$$

For any good trajectory $\tau \in G_\tau$, we have

$$\left|\log \frac{ZP_F(\tau) + \delta_c(\tau)}{R(x)P_B(\tau|x) + \delta_c(\tau)}\right| \leq c$$

$$e^{-c} \leq \frac{KP_F(\tau) + \delta_c(\tau)/Z^*}{\hat{\pi}(\tau) + \delta_c(\tau)/Z^*} \leq e^c$$

where $K = \frac{Z}{Z^*}$

Now we derive the bound for K, since $\sum_{\tau \in G_\mathcal{T}} P_F(\tau) \leq 1$,

$$\begin{aligned}
K &\geq \sum_{\tau \in G_\mathcal{T}} e^{-c}\hat{\pi}(\tau) - (1 - e^{-c}) \sum_{\tau \in G_\mathcal{T}} \frac{\delta_c(\tau)}{Z^*} \\
&= e^{-c}(1 - \epsilon_c) - (1 - e^{-c}) \sum_{\tau \in G_\mathcal{T}} \frac{R(\tau)}{Z^*} \frac{\delta_c(\tau)}{R(\tau)} \\
&\geq e^{-c}(1 - \epsilon_c) - (1 - e^{-c})M_c \sum_{\tau \in G_\mathcal{T}} \frac{R(\tau)}{Z^*} \\
&\geq e^{-c}(1 - \epsilon_c) - (1 - e^{-c})(1 - \epsilon_c)M_c
\end{aligned}$$

Since $M_c < \frac{1}{e^c-1}$, we have

$$(1-\epsilon_c)\left(e^{-c}-(1-e^{-c})M_c\right) > 0$$

Plug this back, we get

$$P_F(\tau) \leq \frac{e^c\hat{\pi}(\tau)+(e^c-1)\delta_c(\tau)/Z^*}{(1-\epsilon_c)\left(e^{-c}-(1-e^{-c})M_c\right)}$$

$$\begin{aligned}\frac{P_F(\tau)}{\hat{\pi}(\tau)} &\leq \frac{e^c+(e^c-1)\delta_c(\tau)/R(\tau)}{(1-\epsilon_c)\left(e^{-c}-(1-e^{-c})M_c\right)} \\ &\leq \frac{e^c+(e^c-1)M_c}{(1-\epsilon_c)\left(e^{-c}-(1-e^{-c})M_c\right)}\end{aligned}$$

The TV distance between the two distributions over trajectories is bounded by:

$$\begin{aligned}2\mathrm{TV}(P_F,\hat{\pi}) = &\sum_{\tau\in G_{\mathcal{T}}}|P_F(\tau)-\hat{\pi}(\tau)|+ \\ &\sum_{\tau\in G_{\mathcal{T}}^{\complement}}|P_F(\tau)-\hat{\pi}(\tau)|\end{aligned}$$

For the first term $\sum_{\tau\in G_{\mathcal{T}}}|P_F(\tau)-\hat{\pi}(\tau)|$, we split the good set $G_{\mathcal{T}}$ into $G_{\mathcal{T}}^+$ and $G_{\mathcal{T}}^-$ such that

$$P_F(\tau) \geq \hat{\pi}(\tau), \forall\tau \in G_{\mathcal{T}}^+$$

$$P_F(\tau) < \hat{\pi}(\tau), \forall\tau \in G_{\mathcal{T}}^-$$

We have

$$\begin{aligned}&\sum_{\tau\in G_{\mathcal{T}}}|P_F(\tau)-\hat{\pi}(\tau)| \\ =&\sum_{\tau\in G_{\mathcal{T}}^+}(P_F(\tau)-\hat{\pi}(\tau))+\sum_{\tau\in G_{\mathcal{T}}^-}(\hat{\pi}(\tau)-P_F(\tau)) \\ =&\sum_{\tau\in G_{\mathcal{T}}^+}(P_F(\tau)-\hat{\pi}(\tau))+(1-\epsilon_c)-\sum_{\tau\in G_{\mathcal{T}}^+}\hat{\pi}(\tau)-(1-\eta_c)+\sum_{\tau\in G_{\mathcal{T}}^+}P_F(\tau) \\ =&2\sum_{\tau\in G_{\mathcal{T}}^+}(P_F(\tau)-\hat{\pi}(\tau))+\eta_c-\epsilon_c \\ =&2\sum_{\tau\in G_{\mathcal{T}}^+}\hat{\pi}(\tau)(\frac{P_F(\tau)}{\hat{\pi}(\tau)}-1)+\eta_c-\epsilon_c \\ \leq&2(1-\epsilon_c)(\frac{e^c+(e^c-1)M_c}{(1-\epsilon_c)\left(e^{-c}-(1-e^{-c})M_c\right)}-1)+\eta_c-\epsilon_c \\ =&2(\frac{e^c+(e^c-1)M_c}{e^{-c}-(1-e^{-c})M_c}-1)+\eta_c+\epsilon_c\end{aligned}$$

For the second term $\sum_{\tau\in G_{\mathcal{T}}^{\complement}}|P_F(\tau)-\hat{\pi}(\tau)|$, we directly use the triangle inequality and get

$$\begin{aligned}\sum_{\tau\in G_{\mathcal{T}}^{\complement}}|P_F(\tau)-\hat{\pi}(\tau)| &\leq \sum_{\tau\in G_{\mathcal{T}}^{\complement}}P_F(\tau)+\sum_{\tau\in G_{\mathcal{T}}^{\complement}}\hat{\pi}(\tau) \\ &= \eta_c+\epsilon_c\end{aligned}$$

Combine the first term and the second term, we have

$$\mathrm{TV}(P_F, \hat{\pi}) \leq \left(\frac{e^c + (e^c - 1)M_c}{e^{-c} - (1 - e^{-c})M_c} - 1 \right) + \epsilon_c + \eta_c$$

For the TV distance between marginal distributions, we have

$$\mathrm{TV}(P_T, \pi_{target}) = \frac{1}{2} \sum_x |P_T(x) - \pi_{target}(x)|$$

Since $P_T(x) = \sum_{\tau \to x} P_F(\tau)$ and $\pi_{target}(x) = \sum_{\tau \to x} \hat{\pi}(\tau)$:

$$|P_T(x) - \pi_{target}(x)| = \left| \sum_{\tau \to x} (P_F(\tau) - \hat{\pi}(\tau)) \right| \leq \sum_{\tau \to x} |P_F(\tau) - \hat{\pi}(\tau)|$$

Summing over all x:

$$\begin{aligned}
\mathrm{TV}(P_T, \pi_{target}) &\leq \frac{1}{2} \sum_x \sum_{\tau \to x} |P_F(\tau) - \hat{\pi}(\tau)| \\
&= \mathrm{TV}(P_F, \hat{\pi}) \\
&\leq \left(\frac{e^c + (e^c - 1)M_c}{e^{-c} - (1 - e^{-c})M_c} - 1 \right) + \epsilon_c + \eta_c \\
&\leq \left(\frac{e^c + (e^c - 1)M_c}{e^{-c} - (1 - e^{-c})M_c} - 1 \right) + \beta_m(\alpha) + \beta_n(\alpha) \\
&\leq \left(\frac{e^c + (e^c - 1)M_c}{e^{-c} - (1 - e^{-c})M_c} - 1 \right) + \frac{2\log(2/\alpha)}{m} + \frac{2\log(2/\alpha)}{n}
\end{aligned}$$

□

C Connection Between Our Theoretical Findings and Existing GFlowNets Training Approaches

We analyze existing advances in GFlowNet training through the lens of our theoretical results, interpreting how diverse algorithmic design choices can implicitly improve stability and certification performance.

- **Annealing Schedules.** Kim et al. [2024b] proposed scaling logits (temperature annealing), while Chen and Mauch [2024] used order-preserving flows. While these methods do not explicitly use a reference flow variable δ, they achieve a similar stabilizing effect by manipulating the target distribution. High initial temperatures flatten the energy landscape, keeping log-ratios $\log \frac{P_F}{R}$ small, thereby mitigating abrupt increases in flow mismatch and postponing large loss explosions.
- **Sub-trajectory and Partial Losses.** Madan et al. [2023] introduced Sub-Trajectory Balance (SubTB) to assign credit to partial trajectories, while Shen et al. [2023] parametrized policies over transitions. By decomposing the global trajectory loss into local constraints, these methods reduce the variance of the flow mismatch. In our framework, lower variance implies that a smaller reference flow $\delta(\tau)$ is sufficient to satisfy the stability condition $\mathcal{L} \leq c^2$, directly improving the certification performance.
- **Divergence and Distributional Objectives.** Distributional objectives [Zhang et al., 2024] and alternative divergence-based losses [Silva et al., 2024, Hu et al., 2025] replace mean squared flow mismatches with smoother optimization criteria. These formulations suppress extreme gradient spikes, implicitly bounding the loss and improving training stability.

- **Guided Exploration.** GAFN [Pan et al., 2023] and Double GFN [Lau et al., 2023] inject stochasticity through reference transitions or dual-network designs. By forcing the policies to stay random enough to keep looking for better solutions, they promote the representativeness of end-state coverage and therefore improve verification effectiveness. Kim et al. [2024a] utilized backward sampling for local search around discovered modes. Lau et al. [2024] combined GFlowNets with Q-functions, while Ikram et al. [2025], Kim et al. [2024c], and Kim et al. [2025b] integrated evolutionary and genetic algorithms to guide exploration. Additionally, Kim et al. [2025a] leveraged a teacher policy to focus sampling specifically on regions where the student policy exhibits high loss. These methods aggressively expand the support of the discovered subgraph $\mathcal{X}_{sub}$. In the context of our bounds, they implicitly sample more high-loss/undervisited "backward" sampled trajectories associated with high reward. By encouraging the training process to observe and minimize loss on these specific trajectories, these methods improve the verification bound.
- **Backward Policy Optimization.** Jang et al. [2024] proposed pessimistic backward policies. Through the lens of Theorem 3.6, this biases P_B to align backward samples with forward-visited trajectories. This alignment improves verification performance.
- **Replay Buffer.** Shen et al. [2023] introduced reward-prioritized replay, which, through the lens of Theorem 3.6, *biases the empirical sampling distribution* toward backward-sampled trajectories (that terminate in high-reward states) and repeatedly revisiting these trajectories helps improve the verification bounds.
- **Structure and Pre-training.** Nguyen et al. [2023] and Shen et al. [2024] formulated GFlowNets for hierarchical and conditional generation, while Pan et al. [2024] utilized pre-training pipelines. He et al. [2025] further extended this to retrospective synthesis. These methods mitigate the "incremental mode coverage" challenge described in Section 3.1. By decomposing the search space or conditioning on specific goals, these methods increase the local contrast ratio $\Lambda_{\mathcal{X}}$ in Proposition 3.4. Theoretically, this ensures high-reward states maintain non-negligible probability, preventing loss explosions.

D More Details of Stable GFlowNets

Adaptive threshold selection. The loss threshold c controls the extent of stabilization by capping the effective training loss. Since the typical scale of $\mathcal{L}_{\mathrm{TB}}$ is problem-dependent (e.g., reward scale, environment size, and policy entropy), a fixed global choice is brittle. We therefore treat c as an adaptive parameter and update it online using an exponential moving-average rule, analogous to the soft target updates used in reinforcement learning [Lillicrap et al., 2015]. Concretely, at iteration t, we set

$$c_{\text{new}} \leftarrow (1-\beta)\, c_{\text{old}} + \beta \max_{\tau \in \mathcal{T}_{batch}} \sqrt{\mathcal{L}_{\mathrm{TB}}(\tau)}, \tag{21}$$

where $\beta \in (0,1)$ is a small smoothing coefficient (we use $\beta = 0.05$ for our experiments). This update tracks the prevailing loss scale while avoiding abrupt changes, ensuring that stabilization remains neither overly conservative (too small c) nor inactive (too large c). The aggregation operator is chosen heuristically, and Appendix F.2.1 empirically compares max, mean, and median aggregation, as well as training without reference flow.

Computing $\mathcal{M}_{\mathrm{TV}}$ and $\mathcal{B}_{\mathrm{TV}}$ via 1D optimization. We compute the probabilistic certificate $\mathcal{B}_{\mathrm{TV}}$ (and similarly the estimated TV bound $\mathcal{M}_{\mathrm{TV}}$ from Theorem 3.10) by exploiting a one-dimensional structure. For a fixed threshold c, the bound is monotone in the reference-flow magnitude, so tightening the certificate reduces to a bounded scalar optimization over that single degree of freedom. In practice, we use `scipy.optimize.minimize_scalar` with a bounded search interval. The interval's upper bound is $\max_i \sqrt{\mathcal{L}_{\mathrm{TB}}(\tau_i)}$, the lower bound is obtained from $c > 0$ and the condition $\frac{\delta_c(\tau_i)}{R(\tau_i)} < \frac{1}{\exp(c)-1}$ for every trajectory τ_i, which leads to:

$$c > \max_{i \in \{i \mid ZP_F(\tau_i) > 2R(\tau_i)\}} \log\left(\frac{ZP_F(\tau_i)}{R(\tau_i)} - 1\right)$$

Full algorithm. The full algorithm is as below. When global backward sampling is available, replace $\mathcal{X}_{sub}$ with the full terminal state space $\mathcal{X}$ and omit the subset-update step.

Algorithm 1 Stable GFlowNets with TV Monitoring and Probabilistic Guarantees

Require: TV target d; confidence $1-2\alpha$; initial threshold c; patience W; subset size K; batch sizes m, n; maximum rounds $T_{\max}$.

Initialize $\mathcal{B}_{\text{TV}} \leftarrow 1$, $\mathcal{X}_{sub} \leftarrow \emptyset$, $w \leftarrow 0$, and $t \leftarrow 0$.
while $\mathcal{B}_{\text{TV}} > d$ and $t < T_{\max}$ **do**
  $t \leftarrow t+1$; $b_1 \leftarrow \infty$.
  Sample m forward trajectories $\mathcal{T}_F$ from P_F.
  $\mathcal{X}_{old} \leftarrow \mathcal{X}_{sub}$.
  Update $\mathcal{X}_{sub}$ by retaining up to K distinct highest-reward states from $\mathcal{X}_{old} \cup \{x_\tau : \tau \in \mathcal{T}_F\}$, where x_τ is the terminal state of τ.
  Sample n backward trajectories $\mathcal{T}_B$: draw $x \in \mathcal{X}_{sub}$ with probability $R(x)/\sum_{y\in\mathcal{X}_{sub}} R(y)$, then draw $\tau \sim P_B(\cdot \mid x)$.
  $w \leftarrow \mathbf{1}\{\mathcal{X}_{sub} = \mathcal{X}_{old}\}(w+1)$.
  if $w \geq W$ **then**
    Compute $\mathcal{B}_{\text{TV}}$ at confidence $1-2\alpha$ using Thm. 3.11 for trajectories ending in $\mathcal{X}_{sub}$.
    Set b_1 to the first term of Eq. (20).
    if $\mathcal{B}_{\text{TV}} \leq d$ **then**
      break
    end if
  end if
  if $b_1 < d$ **then**
    Keep the model fixed and accumulate additional certification samples.
  else
    Compute $\mathcal{L}_{TB}$ on $\mathcal{T}_F \cup \mathcal{T}_B$.
    Update c using Eq. (21).
    Compute the reference flow using Eq. (15).
    Compute $\mathcal{L}_{aug}$ using Eq. (14).
    Update P_F, P_B, and $\log Z$ by minimizing $\mathcal{L}_{aug}$.
    $\mathcal{B}_{\text{TV}} \leftarrow 1$. ▷ Invalidate the pre-update certificate
  end if
end while
Compute $\mathcal{M}_{\text{TV}}$ for saved checkpoints over $\mathcal{X}_{sub}$ using Eq. (17).
return P_F, P_B, $\log Z$, $\mathcal{B}_{\text{TV}}$, $\mathcal{M}_{\text{TV}}$, $\mathcal{X}_{sub}$.

E Experimental Details

E.1 Implementations

We implement GFlowNet training with DB, FM, TB, and SubTB losses following Lahlou et al. [2023b]. For adaptive teacher networks, we adopt the implementation of Kim et al. [2025a]. For weighted detailed balance (WDB), we follow Silva et al. [2025a] by reweighting each transition's loss inversely by the number of terminal states reachable from that transition, and then normalizing the weights within the sampled trajectory so they sum to 1. We parameterize P_F, P_B, and the flow function F with the same MLP architecture but separate parameters.

For optimization stability, we follow Shen et al. [2023] and clip gradient norms to 10.0 and clamp policy logits to $[-50, 50]$. Although gradient clipping is not enabled in the original torchgfn [1] code, we found it consistently improves baseline stability and performance; we therefore apply it to all methods to ensure a fair comparison and to isolate the gains from Stable GFlowNets. With gradient clipping, we also find that model performance is less sensitive to partition-function initialization, so we fix initial $\log Z_\theta = 0$ and do not tune it across environments.

For the L14-RNA1 task, we use a reward-prioritized replay buffer of size $1,000$ following Kim et al. [2025a]. We also adopt ϵ-greedy exploration [Jain et al., 2022] with $\epsilon = 0.05$: with probability 0.05, the forward policy takes a uniformly random action instead of sampling from P_{F_θ}.

For the sEH environment [Bengio et al., 2021], we use a molecule-graph fragment MDP where the agent sequentially attaches fragments from a library of 72 unique blocks, growing molecules up to 8 blocks. Let $R_{\text{raw}}(x)$ denote the untransformed output of the pretrained sEH proxy. The reward used

[1] `https://github.com/GFNOrg/torchgfn/blob/master/tutorials/examples/train_hypergrid.py`

for training is

$$R(x) = \left(\frac{\max\{R_{\text{raw}}(x), 0.01\}}{8}\right)^{10}$$

For evaluation, we define a mode as a distinct generated molecule satisfying $R_{\text{raw}}(x) \geq 7.5$, with uniqueness determined by canonical SMILES. The threshold 7.5 applies to the raw proxy output, not to the transformed training reward $R(x)$. Scaffold counts and Tanimoto-separated counts are computed from the same set of qualifying, canonical-unique molecules. To ensure full reproducibility, we enforce deterministic PyTorch and CUDA execution, evaluate the pretrained PyTorch Geometric proxy on the CPU, and verify exact agreement across five repeated runs.

All RQ2 experiments are repeated over five random seeds $470, 3825, 4444, 8888, 9528$. For RQ1, we use the results obtained by seed 470. For RQ3, we run the extended L14-RNA1 experiments with seed 0 due to the limit of our computational resources.

E.2 Hyperparameters

For Regular Tree and Hypergrid, we use an MLP with two hidden layers of 256 units each; the Hypergrid architecture matches the setting in Malkin et al. [2022a]. For L14-RNA1, we follow Kim et al. [2025a] and use a hidden size of 128 for short runs (10,000 training rounds). Prior work typically evaluates over fewer rounds, but in our longer runs we find wider networks perform better; therefore for extended training (300,000 rounds), we use a hidden size of 256. For sEH, we parameterize both forward and backward policies using a 3-layer MLP with 256 hidden units and LeakyReLU activations.

We use a batch size of 32. The learning rate for $\log Z_\theta$ is set to be $100\times$ the learning rate of the forward and backward policies. Hyperparameters are selected by grid search: policy learning rates in $\{10^{-4}, 5\times10^{-4}, 10^{-3}\}$ and activation functions in ReLU, LeakyReLU. We use a learning rate of 10^{-3} for Regular Tree, 10^{-4} for Hypergrid and L14-RNA1, and 5×10^{-4} for sEH, with LeakyReLU in all experiments.

For Stable GFlowNets, we split each batch evenly: half (i.e., 16 trajectories) is from forward sampling and half from backward sampling. We set the initial loss threshold for stabilization to $c = 10$, and the moving-average coefficient to $\beta = 0.05$. We set the TV target to $d = 0.01$ and the confidence level to 0.9 ($\alpha = 0.05$), which is relatively strict. For L14-RNA1, we set the patience parameter to 10 and the high-reward state size to $|\mathcal{X}_{\text{sub}}| = 10{,}000$. Under these settings, only the simplest Regular Tree environment terminates early when the certification condition is met.

Sampling composition. In RQ2, Stable uses 16 forward-sampled and 16 backward-sampled trajectories per round. Its internal ablation, TB + backward sampling, uses the same sampling mixture but disables reference-flow stabilization. StableTeacher uses 16 forward, 8 backward, and 8 teacher-sampled trajectories per round. Thus, backward and teacher samples replace part of the training batch rather than increasing its size. For the extended L14-RNA1 experiments, StableTeacher uses 48 trajectories per round, comprising 16 samples from each source.

E.3 Computational Resources

All experiments were conducted on an internal compute cluster managed by the SLURM workload manager, utilizing reproducible job scripts. Each compute node was equipped with a single NVIDIA A10 GPU (24GB VRAM) and an AMD EPYC 7413 24-core CPU. The software environment relied on Python 3.10.18 and PyTorch 2.5.1, compiled with CUDA 12.4.

F Additional Experimental Results

F.1 Additional Results for RQ1: Learned patterns in the Hypergrid environment in Figure 3

Figures 5-7 visualize samples from the trained models at different training stages. These patterns offer a possible explanation for the larger fluctuations and higher max-to-rest loss ratios observed in the $H = 16$ Hypergrid. More frequent visits to high-reward states may expose the model to poorly fitted trajectories more often. Meanwhile, TB typically concentrates its samples in a single quadrant

of the displayed projections, leaving high-reward regions elsewhere poorly represented. Trajectories reaching these underexplored regions can therefore produce large training losses. Overall, these observations suggest that instability depends not only on reward sparsity but also on the process of discovering additional high-reward modes: once an existing mode is well fitted, encountering a previously underexplored mode can produce a loss that dominates the batch, causing the max-to-rest ratio to spike.

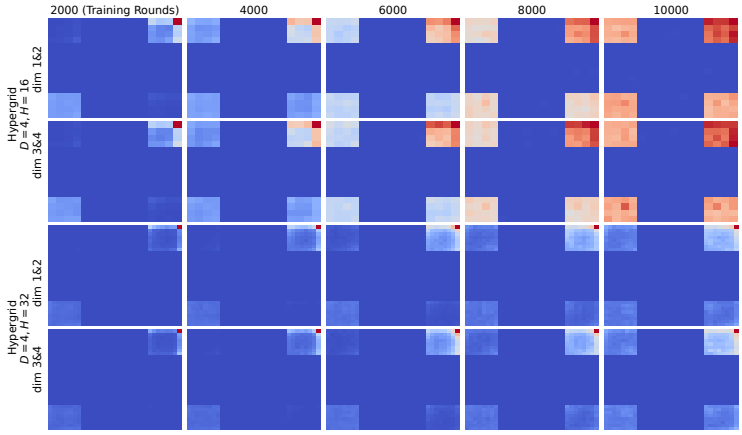

Figure 5: Learned patterns of DB. Each subfigure uses 100,000 samples.

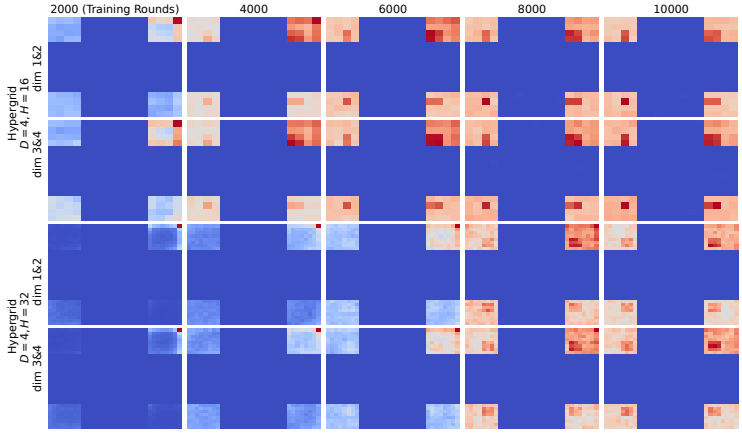

Figure 6: Learned patterns of FM. Each subfigure uses 100,000 samples.

F.2 Additional Results for RQ2

F.2.1 Ablating the Reference Flow in Stable GFlowNets

We ablate the reference-flow loss used in Stable GFlowNets by comparing different aggregation rules for updating the adaptive loss threshold in Equation (21). Specifically, we consider the maximum, mean, and median batch losses, as well as a *None* variant that removes the reference-flow loss while keeping the remaining training procedure unchanged. This ablation is conducted under the RQ2 setting. For L14-RNA1, we use a larger MLP with hidden-layer widths 256/256, whereas Table 2 uses 128/128.

Overall, the maximum and mean aggregations provide the most reliable behavior. On Hypergrid, both achieve strong performance, with the mean slightly better at $H = 16$ and the maximum slightly better at the harder $H = 32$ setting. On sEH, mean aggregation yields slightly higher average mode coverage than maximum aggregation, but with substantially greater variability across seeds. These results suggest that effective stabilization requires balancing the attenuation of extreme losses with the preservation of useful learning signals. Maximum and mean aggregation remain responsive to

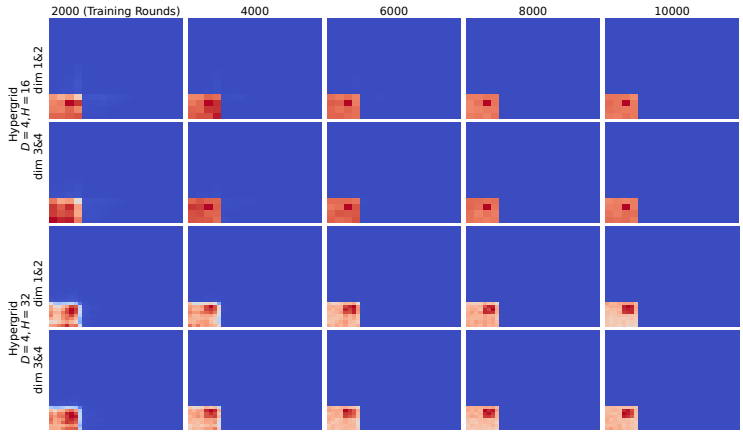

Figure 7: Learned patterns of TB. Each subfigure uses 100,000 samples.

rare, high-loss trajectories, allowing the threshold to adapt to their magnitude while reference flow limits residuals that exceed the threshold. In contrast, median aggregation can keep the threshold low when most trajectories have small residuals, resulting in stronger attenuation of the remaining high-loss trajectories. This may suppress useful signals for learning poorly fitted modes, providing a possible explanation for the median variant's poor performance on Hypergrid and sEH.

The L14-RNA1 results suggest a stability–exploration trade-off. Removing reference flow yields the highest average number of modes discovered during training, but also produces a substantially larger Peak Max-to-Rest Ratio, indicating that individual trajectories can dominate the batch loss. Thus, the *None* variant may favor exploration by preserving stronger learning signals from poorly fitted trajectories, at the cost of greater loss imbalance. This interpretation is consistent with Figure 3 and Table 2. On L14-RNA1, where loss imbalance is relatively mild compared with sEH, stronger attenuation may make learning more conservative and limit mode discovery. In contrast, reference-flow stabilization provides substantial benefits on the much larger sEH task, where extreme loss imbalance is more pronounced. Together, these observations suggest that the value of stabilization depends on balancing useful exploratory signals against the disruptive effects of highly concentrated losses.

Taken together, these results support the use of maximum or mean aggregation for the adaptive threshold. Using the maximum yields a slightly more conservative reference-flow adjustment, performs best on the hardest Hypergrid setting, and achieves lower variance on sEH, whereas the mean provides comparable performance with smoother threshold updates. The *None* variant can improve exploration on L14-RNA1, but it sacrifices the stabilizing mechanism needed for robust behavior across environments.

F.2.2 Integrate Adaptive Teacher and Stable GFlowNets

We evaluate a simple integration of Stable GFlowNets with the Adaptive Teacher method of Kim et al. [2025a]. In addition to forward- and backward-sampled trajectories, we include a third set of teacher-sampled trajectories. Each batch contains 16 forward, 16 backward, and 16 teacher trajectories (batch size 48). We use the same batch size for all baselines to ensure a fair comparison.

Figure 8 reports the long-run results on L14-RNA1. StableTeacher discovers 8,858 modes during training, covering 98.78% of the modes in this environment while the total number of sampled trajectories, including repeated samples, amounts to only 5.36% of the number of possible terminal states. Measured by the total reward mass accumulated by the Top-10,000 terminal states, StableTeacher also achieves higher values than Teacher early in training. The smoothed loss curves indicate that StableTeacher responds systematically when new modes are uncovered, while maintaining a more balanced max-to-rest ratio throughout training. In contrast, Teacher shows an increasing max-to-rest ratio later in training, suggesting that a small number of trajectories dominate the batch as high-reward modes emerge, which is consistent with less stable optimization.

Table 5: Effect of adaptive loss-threshold aggregation on Hypergrid, L14-RNA1, and sEH. We compare maximum, mean, and median aggregation rules for updating the adaptive threshold in Equation (21), together with a *None* variant that removes the reference-flow loss. We report the Hypergrid empirical total L_1 distance, L14-RNA1 mode counts discovered during training, the Peak Max-to-Rest Ratio, and sEH training scaffold counts. The Max-to-Rest Ratio is defined as the largest trajectory loss in a batch divided by the sum of the remaining trajectory losses; we report its peak value over training. Hypergrid evaluations use 10^5 samples from the final forward policy P_F (mean $\pm$ std over 5 seeds). For L14-RNA1, we use a larger MLP with hidden-layer widths 256/256, whereas Table 2 uses 128/128.

	Hypergrid (Empirical Total $L_1 \downarrow$)		**L14-RNA1**		**sEH**
Method	$D = 4, H = 16$	$D = 4, H = 32$	# modes ↑	Peak Max-to-Rest Ratio	# scaffolds ↑
Max	0.290 ± 0.002	$\mathbf{0.713 \pm 0.002}$	1754.2 ± 123.0	11.5 ± 2.0	14142.6 ± 2388.2
Mean	$\mathbf{0.287 \pm 0.003}$	0.730 ± 0.001	1745.4 ± 37.6	9.5 ± 2.4	$\mathbf{14721.4 \pm 4974.3}$
Median	1.885 ± 0.000	1.875 ± 0.000	1744.6 ± 43.6	16.8 ± 13.2	1283.6 ± 1010.0
None (TB + backward sampling)	0.316 ± 0.003	$\mathbf{0.713 \pm 0.003}$	$\mathbf{1830.4 \pm 58.7}$	81.3 ± 39.3	8680.2 ± 2254.2

We also compare TB against Stable GFlowNets in long runs. With gradient clipping, TB performs well early on but tends to saturate, becoming less likely to discover additional modes. Its training loss continues to decrease, yet exploration slows. Stable GFlowNets, in contrast, continue to adapt as new modes appear: their loss remains higher than TB in later stages, reflecting active correction driven by newly discovered high-reward outcomes, whereas standard TB tends to settle into its "comfort zone". We observe a concurrent increase in the max-to-rest ratio, which we hypothesize is transient as the loss threshold increases and will subside once the newly discovered modes are sufficiently represented in the training distribution.

Collectively, these results suggest that Stable GFlowNets enables stronger incremental mode coverage, complementing aggressive exploration in settings where sampling from the full set of terminal states is difficult.

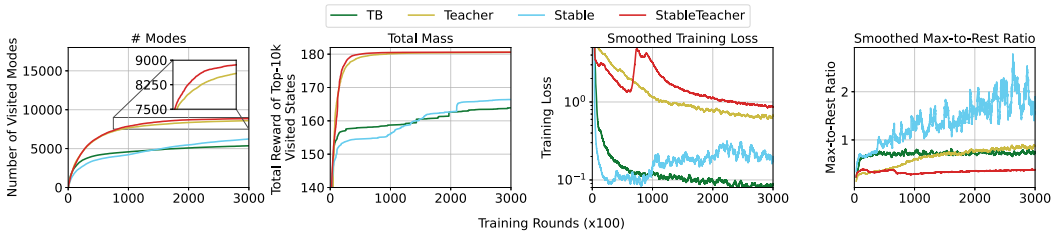

Figure 8: Extended experiment results for L14-RNA1. Total mass represents the cumulative reward captured by the top-K high-reward states discovered during training.

F.2.3 sEH Mode Counts under Different Definitions

Table 6 reports three complementary molecular diversity metrics for sEH: the number of unique modes, the number of distinct Bemis-Murcko scaffolds, and the number of Tanimoto-separated molecules.

For the **# modes** metric, we define a mode as a uniquely generated molecule with raw proxy reward $R_{\text{raw}}(x) \geq 7.5$, where uniqueness is determined by canonical SMILES rather than Tanimoto similarity. Specifically, we retain nonempty terminal-molecule SMILES satisfying $R_{\text{raw}}(x) \geq 7.5$, parse them using `Chem.MolFromSmiles`, and canonicalize and deduplicate them using `Chem.MolToSmiles(mol, canonical=True)`.

For the **Bemis-Murcko scaffold** metric, we compute the scaffold of each canonical-unique molecule using `MurckoScaffold.MurckoScaffoldSmiles(mol=mol, includeChirality=False)` and report the number of distinct scaffold SMILES. Under RDKit's standard representation, two molecules

Table 6: Molecular mode and diversity metrics for sEH. Results are reported as mean ± standard deviation.

Method	# modes	Bemis–Murcko scaffolds	Tanimoto < 0.7
TB	137.2 ± 77.9	137.2 ± 77.9	55.2 ± 22.8
DB	32.2 ± 12.5	32.2 ± 12.5	16.8 ± 5.6
SubTB	68.4 ± 41.9	68.4 ± 41.9	35.8 ± 14.5
WDB	3.2 ± 2.8	3.2 ± 2.8	2.8 ± 1.9
Teacher	5.6 ± 5.9	5.6 ± 5.9	5.2 ± 5.1
Stable	$\mathbf{14142.6 \pm 2388.2}$	$\mathbf{14142.6 \pm 2388.2}$	$\mathbf{3692.4 \pm 938.7}$
StableTeacher	3728.8 ± 4197.8	3722.2 ± 4187.8	457.8 ± 403.7
TB + backward sampling	8680.6 ± 2254.2	8680.2 ± 2254.2	2058.4 ± 402.7

share a scaffold when removing their side chains yields the same typed ring-and-linker structure, irrespective of chirality.

The **Tanimoto <0.7** metric is computed independently. Canonical-unique molecules are processed in encounter order using `Chem.RDKFingerprint`, and a molecule is retained only if its Tanimoto similarity is strictly below 0.7 with respect to every previously retained representative.

As shown in Table 6, Stable substantially outperforms all baselines under each definition of molecular diversity. It discovers 14142.6 ± 2388.2 unique modes and Bemis-Murcko scaffolds, together with 3692.4 ± 938.7 Tanimoto-separated molecules. This consistent advantage across different diversity criteria indicates that the improvement is not an artifact of a particular diversity definition. The per-seed discovery traces (Figure 9) further show that mode discovery accelerates as training progresses.

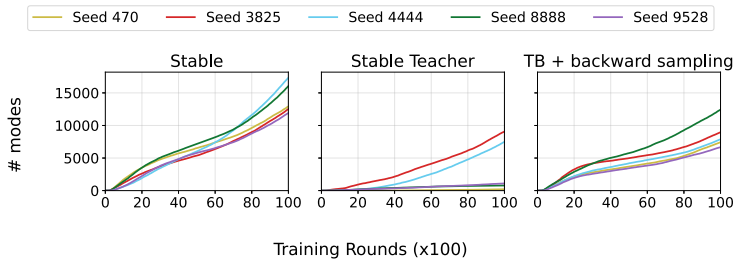

Figure 9: Per-seed molecular mode discovery for sEH. The per-seed traces of Stable GFlowNets show progressively faster mode discovery as training progresses.

F.3 Additional Results for RQ3

We compare FCS and $\mathcal{M}_{\mathrm{TV}}$ as monitoring signals for distributional error, rather than as competing TV certificates. FCS measures flow consistency and corresponds to one component of an existing TV bound, whereas $\mathcal{M}_{\mathrm{TV}}$ is a Monte Carlo estimate of the bound. Neither quantity, as evaluated here, provides a finite-sample confidence guarantee. We therefore assess how well they track changes in true TV, including their correlation across training checkpoints, their sensitivity to subgraph size and sampling budget, and their predictive error where calibration is applied. Formal probabilistic certification is instead provided by $\mathcal{B}_{\mathrm{TV}}$.

Table 7 reports the values at all ten training checkpoints. In most environments, $\mathcal{M}_{\mathrm{TV}}$ follows the overall trend of the true TV error, and the calibrated $\mathcal{M}_{\mathrm{TV}}$ is closer in scale. For example, in HyperGrid ($D = 4, H = 32$), it captures both the general decrease in true TV and the increase at the 80% checkpoint. In L14-RNA1, it also reflects the increases from 40% to 50% and from 60% to 70%.

Table 8 further compares the monitoring behavior of $\mathcal{M}_{\mathrm{TV}}$ and FCS across subgraph sizes and sampling budgets. To match the information available to $\mathcal{M}_{\mathrm{TV}}$, here, FCS uses Monte Carlo estimates of terminal state flows ($\mathrm{budget} = 1$) rather than their exact values. Overall, $\mathcal{M}_{\mathrm{TV}}$ is not highly sensitive to the number of sampled trajectories: across settings, it consistently follows the trend

Table 7: Checkpoint-level values underlying Table 3. Columns correspond to ten evenly spaced checkpoints. Calibrated $\mathcal{M}_{\mathrm{TV}}$ uses an environment-specific affine regression fitted on the first five checkpoints and then frozen: its first five entries are in-sample fitted values, whereas its last five entries are held-out predictions.

Environment	Metric	Checkpoint									
		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
RegularTree $D = 5$	$\mathcal{B}_{\mathrm{TV}}$	0.1554	0.1347	0.1260	0.1169	0.1086	0.0867	0.0747	0.0747	0.0747	0.0747
	$\mathcal{M}_{\mathrm{TV}}$	0.0030	0.0017	0.0010	0.0006	0.0003	0.0002	0.0003	0.0003	0.0003	0.0003
	Calibrated $\mathcal{M}_{\mathrm{TV}}$	0.0010	0.0006	0.0003	0.0002	0.0001	8.77×10^{-5}	9.50×10^{-5}	9.50×10^{-5}	9.50×10^{-5}	9.50×10^{-5}
	FCS	0.0012	0.0007	0.0004	0.0003	0.0002	0.0001	6.32×10^{-5}	6.32×10^{-5}	6.32×10^{-5}	6.32×10^{-5}
	True TV	0.0010	0.0006	0.0003	0.0002	0.0001	7.85×10^{-5}	6.35×10^{-5}	6.35×10^{-5}	6.35×10^{-5}	6.35×10^{-5}
RegularTree $D = 10$	$\mathcal{B}_{\mathrm{TV}}$	0.3009	0.1042	0.0914	0.0864	0.0839	0.0826	0.0805	0.0804	0.0815	0.0802
	$\mathcal{M}_{\mathrm{TV}}$	0.1269	0.0068	0.0035	0.0026	0.0021	0.0020	0.0018	0.0018	0.0016	0.0017
	Calibrated $\mathcal{M}_{\mathrm{TV}}$	0.0044	0.0010	0.0010	0.0009	0.0009	0.0009	0.0009	0.0009	0.0009	0.0009
	FCS	0.0046	0.0020	0.0010	0.0007	0.0006	0.0005	0.0005	0.0005	0.0005	0.0005
	True TV	0.0044	0.0017	0.0009	0.0007	0.0006	0.0005	0.0005	0.0005	0.0004	0.0005
RegularTree $D = 15$	$\mathcal{B}_{\mathrm{TV}}$	1.0000	0.1170	0.1024	0.0890	0.0835	0.0822	0.0824	0.0821	0.0818	0.0812
	$\mathcal{M}_{\mathrm{TV}}$	0.7088	0.0097	0.0056	0.0031	0.0025	0.0020	0.0019	0.0019	0.0017	0.0016
	Calibrated $\mathcal{M}_{\mathrm{TV}}$	0.0058	0.0014	0.0014	0.0013	0.0013	0.0013	0.0013	0.0013	0.0013	0.0013
	FCS	0.0064	0.0029	0.0015	0.0008	0.0006	0.0005	0.0004	0.0004	0.0004	0.0004
	True TV	0.0057	0.0027	0.0014	0.0008	0.0006	0.0005	0.0005	0.0005	0.0004	0.0004
HyperGrid $D = 4, H = 8$	$\mathcal{B}_{\mathrm{TV}}$	1.0000	1.0000	0.7077	0.3794	0.5045	0.4374	0.6033	0.3363	0.3599	0.2542
	$\mathcal{M}_{\mathrm{TV}}$	0.3994	0.1807	0.1273	0.0776	0.0744	0.0631	0.0640	0.0476	0.0433	0.0498
	Calibrated $\mathcal{M}_{\mathrm{TV}}$	0.0933	0.0375	0.0238	0.0111	0.0103	0.0074	0.0077	0.0035	0.0024	0.0040
	FCS	0.1197	0.0323	0.0208	0.0135	0.0101	0.0074	0.0085	0.0065	0.0070	0.0058
	True TV	0.0945	0.0350	0.0215	0.0142	0.0110	0.0094	0.0100	0.0069	0.0077	0.0064
HyperGrid $D = 4, H = 16$	$\mathcal{B}_{\mathrm{TV}}$	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	$\mathcal{M}_{\mathrm{TV}}$	0.5431	0.4037	0.3524	0.3941	0.3531	0.2851	0.2283	0.2243	0.1606	0.1957
	Calibrated $\mathcal{M}_{\mathrm{TV}}$	0.1223	0.0966	0.0871	0.0948	0.0873	0.0747	0.0642	0.0635	0.0517	0.0582
	FCS	0.0864	0.0530	0.0651	0.0621	0.0377	0.0399	0.0274	0.0278	0.0298	0.0319
	True TV	0.1223	0.0989	0.0930	0.0920	0.0819	0.0883	0.0710	0.0662	0.0608	0.0666
HyperGrid $D = 4, H = 32$	$\mathcal{B}_{\mathrm{TV}}$	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	$\mathcal{M}_{\mathrm{TV}}$	0.8421	0.6878	0.5429	0.4314	0.4222	0.3791	0.3826	0.4094	0.3189	0.3332
	Calibrated $\mathcal{M}_{\mathrm{TV}}$	0.1916	0.1435	0.0984	0.0637	0.0609	0.0475	0.0485	0.0569	0.0287	0.0332
	FCS	0.2162	0.1139	0.1203	0.0965	0.0573	0.0490	0.0444	0.0542	0.0453	0.0451
	True TV	0.2076	0.1199	0.0940	0.0756	0.0611	0.0515	0.0474	0.0595	0.0381	0.0439
L14-RNA1	$\mathcal{B}_{\mathrm{TV}}$	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	$\mathcal{M}_{\mathrm{TV}}$	0.4990	0.4100	0.3465	0.2740	0.3545	0.2631	0.3104	0.2553	0.2273	0.2001
	Calibrated $\mathcal{M}_{\mathrm{TV}}$	0.1066	0.0934	0.0840	0.0733	0.0852	0.0716	0.0787	0.0705	0.0663	0.0623
	FCS	0.0375	0.0199	0.0302	0.0072	0.0207	0.0132	0.0265	0.0225	0.0143	0.0148
	True TV	0.1082	0.0910	0.0942	0.0735	0.0756	0.0537	0.0626	0.0480	0.0381	0.0343

of the subgraph True TV. Although FCS can be numerically closer to the subgraph True TV in absolute value for small K, it increasingly underestimates the model TV as K grows. This is because FCS corresponds to only one component of the TV bound in Corollary 2 of Silva et al. [2025a]. As K increases, covering the larger subspace with limited samples becomes more difficult, so the induced subspace size β remains small relative to K (Table 9). Consequently, FCS becomes relatively insensitive to the true size of the certified subgraph, and the gap between FCS and the full TV bound can grow through the second term. In contrast, $\mathcal{M}_{\mathrm{TV}}$ tracks the subgraph True TV much more consistently across K, achieving a linear-fit R^2 above 0.99 when $m \geq 100$, whereas FCS yields substantially lower R^2 values.

Table 8: Robustness to subgraph size for Monte Carlo estimator $\mathcal{M}_{\mathrm{TV}}$, FCS baseline, and subgraph True TV on L14-RNA1 using StableTeacher trained for 300,000 rounds, across varying numbers of sampled trajectories. Percentages indicate the reward mass captured by each top-K subgraph. We also include the ground-truth top-200,000 high-reward states ("Oracle"), which capture 99.93% of the total rewards. The linear-fit R^2 is computed across subgraph sizes and measures association with subgraph TV.

Metric / Setting	$K = 10$ (28.56%)	$K = 10^2$ (62.57%)	$K = 10^3$ (87.27%)	$K = 10^4$ (97.63%)	Oracle (99.93%)	Linear-fit R^2 vs. Subgraph True TV
$\mathcal{M}_{\mathrm{TV}}$ ($m = 10$)	0.2170	0.2783	0.3414	0.4059	0.3915	0.9689
$\mathcal{M}_{\mathrm{TV}}$ ($m = 100$)	0.2195	0.2897	0.3610	0.4328	0.4544	0.9991
$\mathcal{M}_{\mathrm{TV}}$ ($m = 1000$)	0.2169	0.2882	0.3711	0.4573	0.4696	0.9952
FCS (epochs = 1, bucket = 100)	0.0611	0.0875	0.0879	0.0993	0.1001	0.8548
FCS (epochs = 5, bucket = 100)	0.0704	0.0930	0.0931	0.0986	0.0955	0.6896
FCS (epochs = 1, bucket = 1000)	0.0613	0.0747	0.0809	0.0803	0.0806	0.7713
FCS (epochs = 5, bucket = 1000)	0.0577	0.0749	0.0813	0.0809	0.0810	0.7502
Subgraph True TV	0.0373	0.0608	0.0857	0.1082	0.1182	1.0000

Table 9: Induced subspace size β used in FCS scaling as the top-K subgraph grows.

Metric / Setting	$K = 10$	$K = 10^2$	$K = 10^3$	$K = 10^4$	Oracle
Subspace size β (epochs $= 1$, bucket $= 1000$)	10 ± 0	94.8 ± 1.3	294.8 ± 6.7	368.8 ± 7.2	388.6 ± 9.9

We also compare $\mathcal{M}_{\mathrm{TV}}$ across selected training methods. For Hypergrid, we evaluate policies trained with seed 470 for 30,000 rounds, using checkpoints every 3,000 rounds. For L14-RNA1, we use policies trained with seed 0 for 300,000 rounds and evaluate them every 10,000 rounds. At each checkpoint, estimation is repeated with five Monte Carlo seeds ($\{42, 43, 44, 45, 46\}$). Lines and shaded regions show the mean and sample standard deviation of these estimates for each fixed policy, rather than variability across independent training runs.

First, $\mathcal{M}_{\mathrm{TV}}$ can monitor different GFlowNet training methods, rather than being restricted to Stable GFlowNets. In the simplest Hypergrid setting ($H = 8$), the estimates decrease for all evaluated methods. On larger Hypergrids, Stable and Teacher achieve substantial reductions, whereas standard TB shows limited improvement, with Teacher's gains emerging later at $H = 32$. These observations are consistent with the benefits of directing training on high-reward or poorly fitted trajectories: Stable uses reward-weighted backward sampling, while Teacher targets regions where the student incurs high loss; on L14-RNA1, the estimates also decrease overall, including for TB, under a training setup that uses reward-prioritized replay. Beyond explicitly prioritizing high-reward trajectories (i.e., exploration), SubTB achieves particularly low $\mathcal{M}_{\mathrm{TV}}$, suggesting that effective variance reduction (i.e., stabilization) [Madan et al., 2023] provides another route to enhance GFlowNets training.

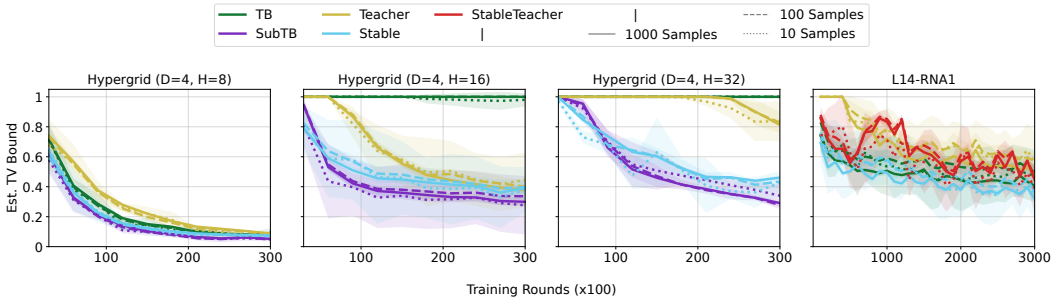

Figure 10: Evolution of empirical M_{TV} estimates (Theorem 3.10). Hypergrid policies (training seed 470) are evaluated every 3,000 rounds through 30,000; L14-RNA1 policies (training seed 0) every 10,000 through 300,000. Lines and shading show the mean $\pm$ one sample standard deviation over five Monte Carlo seeds for each fixed policy. For RNA, all four methods (TB, Teacher, Stable, and StableTeacher) share a fixed reference consisting of the deduplicated union of the available terminal state buffer.

F.4 Computational Overhead of Adaptive Reference Flow

We measured the cost of computing the adaptive reference flow relative to the total training-update cost of Stable GFlowNets. Because a Stable GFlowNet training update consists of the standard TB update plus reference-flow computation, the latter directly measures the additional optimization cost over TB. As shown in Table 10, the overhead remains modest across all evaluated environments.

Table 10: Reference-flow computation cost relative to a training update.

Reference-flow computation / training update (%)	Hypergrid $D = 4, H = 16$	Hypergrid $D = 4, H = 32$	L14-RNA1	sEH
Wall-time ratio	3.21 ± 0.85	2.83 ± 0.06	3.50 ± 0.12	1.31 ± 0.09
CPU+GPU-time ratio	4.56 ± 2.06	3.88 ± 1.49	2.75 ± 0.14	3.81 ± 1.88

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately describe the paper's theoretical contributions and the corresponding algorithmic development.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitation in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Our propositions are limited to the incremental mode coverage setting described in Section 3.1, while the theorems are developed under the standard GFlowNets training setting. Proofs of all propositions, theorems, and the corollary are provided in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the source code for our algorithms, along with scripts for reproducing the experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the source code for our algorithms, along with a README file to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We report the experimental details in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We conduct experiments over five random seeds and report the standard deviation or range of the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the computational platform used for our experiments in Appendix E.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work is theoretical and methodological in nature, uses standard benchmark environments, and does not involve human subjects, private or sensitive data, or deployment in high-risk real-world settings. We do not identify any ethical concerns in the paper.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not identify any risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite and acknowledge the original works that provide the experimental baselines and environments in Appendix E.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide documentation for the code in the accompanying README file.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for grammar suggestions, programming assistance, and editing purposes, and did not affect the core methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.