EVALUATING GENERALIZATION IN GFLOWNETS FOR MOLECULE DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning bears promise for drug discovery problems such as *de novo* molecular design. Generating data to train such models is a costly and time-consuming process, given the need for wet-lab experiments or expensive simulations. This problem is compounded by the notorious data-hungriness of machine learning algorithms. In small molecule generation the recently proposed GFlowNet method has shown good performance in generating diverse high-scoring candidates, and has the interesting advantage of being an off-policy offline method. Finding an appropriate generalization evaluation metric for such models, one predictive of the desired search performance (i.e. finding high-scoring diverse candidates), will help guide online data collection for such an algorithm. In this work, we develop techniques for evaluating GFlowNet performance on a test set, and identify the most promising metric for predicting generalization. We present empirical results on several small-molecule design tasks in drug discovery, for several GFlowNet training setups, and we find a metric strongly correlated with diverse high-scoring batch generation. This metric should be used to identify the best generative model from which to sample batches of molecules to be evaluated.

1 INTRODUCTION

Drug development is a lengthy and costly process, with the average clinical development time reaching more than nine years and median development cost nearing 1 billion USD (Wouters et al., 2020). Preclinical early-stage drug discovery is typically an iterative optimization process that consists of the generate, assay, learn cycle. In order to proceed to clinical trials, molecules should satisfy multiple objectives such as binding affinity to the target protein, cost of synthesis, and drug-likeness. While multiple biological or computational assays are available in a real-world drug discovery setting, we focus on two computational biophysics assays already well-known in literature - docking and QEDBengio et al. (2021a); Xie et al. (2021). We design GFLowNet reward separately for each of these objectives in separate experiments.

During the generation process, we typically have access to a dataset of previously evaluated candidates, $D = (x_i, y_i)$, where x_i is the molecule candidate, and y_i the value of the property of interest (e.g., to be maximized). This data is often limited in size, relative to the size of the search space. When true $y = f(x)$ is not available or is slow to query, the common approach is to train an estimator, such as neural network $\hat{y} = \hat{f}(x)$, also called proxy, and use $\hat{y}$ as a target for the generatorLiu et al. (2020); Bengio et al. (2021a).

Generative Flow Networks (Bengio et al., 2021a;b, or GFlowNets) have recently been proposed as a method for generating diverse and high scoring candidates. GFlowNets generate diverse and high-quality molecule candidates by sampling molecules with a probability proportional to their reward. Bengio et al. (2021a) show promising results for diverse molecule generation for the sEH protein target compared to previous approaches based on MCMC and RL.

Bengio et al. (2021a) propose a TD-like objective for training GFlowNets, learning a stochastic policy that samples discrete objects proportionally to a reward. However, like traditional reinforcement learning methods, evaluating generalization in this setting is tricky since the training data is generated from the policy itself. In this work, we exploit the off-policy nature of the GFlowNet learning objective to define a metric, *GFNEval* and evaluate the performance of GFlowNets on a test set. We

demonstrate empirically on various tasks and through various experiments the predictive power of the proposed metric on the downstream metric of interest. The main contributions of this paper are as follows:

- Empirical validation of an evaluation metric *GFNEval* to track training and evaluate generalization of GFlowNets
- Qualitative insights on the learning dynamics of GFlowNets derived from the proposed metric *GFNEval*

2 BACKGROUND

2.1 PROBLEM SETTING

We consider the problem of generating molecule graphs $x \in \mathcal{X}$, sequentially, by combining building blocks (molecule fragments) from a set $\mathcal{A}$ (Jin et al., 2020; Kumar et al., 2012). Each molecule x has an associated reward $R(x)$, which quantifies the usefulness of the molecule (for instance, the binding energy with a given target protein). This reward $R(x)$ is a usually a proxy for the actual values obtained from expensive experiments, so it can be called often and cheaply.

The goal is to generate a *diverse set of molecules with* high rewards, instead of a single candidate that maximizes the reward. This is critical in realistic settings where the reward $R(x)$ might not capture the desired properties. In this scenario, having a diverse high-scoring set improves the odds of having candidates that satisfy future screening steps. A natural way to get diverse and high-scoring batches would be to sample from the modes of the reward function $R(x)$.

This problem of molecule generation has been tackled through various approaches. Generative models like VAEs and Normalizing Flows (Shi et al., 2020; Jin et al., 2020; Luo et al., 2021) rely on a given set of high-reward (positive) examples for learning a generative model. These method do not leverage the low scoring (negative) samples leading to poor quality of the generative model. MCMC based methods (Seff et al., 2019; Xie et al., 2021) can suffer from mode-mixing issues in the case of high-dimensional well separated modes. Reward maximizing RL (Segler et al., 2017; Cao & Kipf, 2018; Popova et al., 2019; Gottipati et al., 2020; Angermueller et al., 2020) and evolutionary methods (Brown et al., 2004; Jensen, 2019; Swersky et al., 2020) tend to focus on one or a few dominant modes leading to low diversity in the generated molecules (Bengio et al., 2021a).

2.2 GFLOWNETS

Generative Flow Networks (Bengio et al., 2021a;b, or GFlowNets) present a promising approach for generating diverse and high-scoring molecules. GFlowNets learn a stochastic policy π that generates objects x with a probability proportional to their reward $R(x)$, $\pi(x) \propto R(x)$. We provide a brief overview of GFlowNets and refer the reader to Bengio et al. (2021b) for a more thorough discussion.

GFlowNets operate on a space $\mathcal{X}$ which is *compositional*, that is, object $x \in \mathcal{X}$ can be constructed using a sequence of actions taken from some set $\mathcal{A}$. After each step, we may have a partially constructed object s, defining our state space $\mathcal{S}$. Bengio et al. (2021a) use a GFlowNet to sequentially construct a molecule by inserting a molecule fragment in a partially constructed molecule represented by a graph. A special action indicates that the object is complete, i.e., $s = x \in \mathcal{X}$. Each transition $s \to s' \in \mathcal{E}$ from state s to state s' corresponds to an edge in a directed graph $G = (\mathcal{S}, \mathcal{E})$ where the nodes are the state space $\mathcal{S}$ and the edges are the transitions $\mathcal{E}$. Bengio et al. (2021a) assume that this graph is acyclic, meaning that actions are constructive and cannot be undone. An object $x \in \mathcal{X}$ can be constructed by starting from an initial empty state s_0 and applying actions sequentially. All complete trajectories must end in a special final state s_f. The fully constructed objects in $\mathcal{X} \subset \mathcal{S}$ are *terminating states*. Each object x can be constructed by a trajectory of states $\tau = (s_0 \to s_1 \to \ldots \to x \to s_f)$, and we can define $\mathcal{T}$ as the set of all trajectories. Note that there can be multiple trajectories leading to the same terminal state.

Flows A trajectory flow $F : \mathcal{T} \mapsto \mathbb{R}^+$ is a function that assigns a probability mass to every trajectory τ. Using the trajectory flow, the *edge flow* for an edge $s \to s' \in \mathcal{E}$ can be defined as the sum of the flows of all trajectories containing the edge, $F(s \to s') = \sum_{s \to s' \in \tau} F(\tau)$, and the *state*

flow for a state $s \in \mathcal{S}$ can be defined as $F(s) = \sum_{s \in \tau} F(\tau)$. The flow associated with the final transition in the trajectory $F(x \to s_f)$ is called the terminal flow.

The trajectory flow F is a measure over complete trajectories $\tau \in \mathcal{T}$ and it induces a corresponding probability measure

$$P(\tau) = \frac{F(\tau)}{\sum_{\tau \in \mathcal{T}} F(\tau)} = \frac{F(\tau)}{Z} \tag{1}$$

where Z denotes the total flow, and corresponds to the partition function of the the measure F. The forward transition probabilities $P_F(s|s')$ for each step of a trajectory and the probability $P_F(s)$ of visiting a state can then be defined as

$$P_F(s|s') = \frac{F(s \to s')}{F(s)}, \qquad P_F(s) = \frac{\sum_{\tau \in \mathcal{T}: s \in \tau} F(\tau)}{Z}. \tag{2}$$

Flow Matching Criterion A *consistent flow* satisfies the following *flow consistency equation* $\forall s \in \mathcal{S}$ defined as follows:

$$\sum_{s' \in \text{Parent}(s)} F(s' \to s) = \sum_{s'' \in \text{Child}(s)} F(s \to s''). \tag{3}$$

where $\text{Parent}(s) = \{s' : s' \to s \in \mathcal{E}\}$ denotes the parents for node s and $\text{Child}(s) = \{s' : s \to s' \in \mathcal{E}\}$ denotes the children of node s in G.

A key result underpinning GFlowNets from Bengio et al. (2021a) shows that for a consistent flow F where the terminal flow is assigned the value of the reward, i.e., $F(x \to s_f) = R(x)$, a policy π defined by the forward transition probability $\pi(s'|s) = P_F(s'|s)$ constructs object x with probability proportional to $R(x)$

$$\pi(x) = \frac{R(x)}{Z}. \tag{4}$$

GFlowNets learn to approximate an *edge flow* $F_\theta : \mathcal{E} \mapsto \mathbb{R}^+$ defined over G, such that the terminal flow is equal to the reward $R(x)$ and the flow is *consistent*. Bengio et al. (2021a) proposed a temporal difference-like (Sutton & Barto, 2018) learning objective, called *flow-matching*:

$$\mathcal{L}_{FM}(s; \theta) = \left(\log \frac{\sum_{s' \in \text{Parent}(s)} F_\theta(s' \to s)}{\sum_{s'' \in \text{Child}(s)} F_\theta(s \to s'')} \right)^2 \tag{5}$$

Bengio et al. (2021a) show that given trajectories τ_i sampled from an exploratory training policy $\tilde{\pi}$ with full support, an edge flow $\hat{F}$ learned by minimizing Equation 5 is consistent. At this point, the forward transition probability defined by this flow $P_{F_\theta}(s'|s) = \frac{F_\theta(s \to s')}{\sum_{s'' \in \text{Child}} F_\theta(s \to s'')}$ would sample objects x with a probability $P_F(x)$ proportionally to their reward $R(x)$. In practice, the trajectories for training GFlowNets are sampled from an exploratory policy that has higher entropy than the GFlowNet sampler P_{F_θ}, for example by tempering it or mixing it with a uniform policy. In the case where there is only one trajectory into each state, GFlowNets and maximum-entropy regularized RL can converge to the same solution, but this condition is not satisfied with our molecule construction trajectories (since there are many ways to arrive at the same molecule, e.g., by different orders of adding fragments).

We observe that the objective in Equation 5 is *offline* and *off-policy*. The off-policy nature allows us to train on trajectories that are not sampled from the GFlowNet policy. This allows us to incorporate ideas from off-policy reinforcement learning, such as prioritized replay (Schaul et al., 2016), where we can store molecules encountered during training in a buffer, and incorporate trajectories sampled backward from these molecules in future training batches.

Like other TD based algorithms, the training error alone can be misleading to track training progress since the trajectories are sampled from the policy itself. Additionally, contrary to standard reinforcement learning methods, tracking the reward as a way to measure performance is not suited for GFlowNets since that is not the objective being optimized. This disconnect calls for a metric to evaluate learning progress in GFlowNets. We propose the following properties that are critical for any such metric.

- **Predictive of Downstream Performance**: The metric should be correlated with the downstream metric of interest, for instance the score and diversity of the top generated molecules.
- **Computed Offline**: The metric should be computable on a fixed set of data, available a priori.

In the following sections, we present an evaluation methodology for GFlowNets in the context of molecule design and present an empirical analysis of the metric.

3 EVALUATING GFLOWNETS

We can compute exactly the sampling probability $\pi(x)$ of a terminal state x under a GFlowNet defined by the state flow F_θ by considering all the trajectories terminating at the given terminal state, and going backward recursively as stated in 6. Note that we operate in the $\log$ domain for numerical stability.

$$\log(\pi_\theta(s)) = \log\left(\sum_{s' \in Parent(s)} \exp\left(\log(P_{F_\theta}(s|s')) + \log(\pi_\theta(s'))\right)\right) \tag{6}$$

where we compute this quantity for $s \in \mathcal{X}$. The exact probability of sampling a terminal state x ($\pi(x)$) under a GFlowNet with relatively limited resources is a key component for evaluating the GFlowNet objective. We can define metrics using the sampling probabilities of molecules from a test set (that have not been used for training[1]) to track the progress of training.

A natural way to evaluate the GFlowNet objective of sampling molecules *proportionally* to their rewards is the Spearman's rank correlation coefficient between the probability of sampling molecules from a test set under the GFlowNet and their respective rewards, which we call *GFNEvalS*. Since we do not have access to the true partition function $Z = \sum_{x \in \mathcal{X}} R(x)$, we cannot compute the error between the true distribution and the learned distrbution under the GFlowNet objective, however Spearman's correlation can evaluate the key aspects of the relation between π and R, thus alleviating the need to know the true partition function. Note that since we do not have the true partition function, the scale of the reward $R(x)$ and $\pi_\theta(x)$ can be very different, so for numerical stability we use $\log R(x)$ and $\log \pi_\theta(x)$ instead. Malkin et al. (2022) use this metric to evaluate their proposed learning objective. In this work, we focus on a more thorough analysis of this metric and how it relates to the downstream performance on the task of molecule design. We observe that empirically Spearman's correlation offers better results, but we note that the Pearson correlation coefficient can be used as well (we call this *GFlowEvalP*) given the linear relationship between $\log(\pi(x))$ and $\log R(x)$.

$$\begin{aligned} \textit{GFNEvalS} &= \text{Spearman's } \rho_{\log(\pi(x)),\log(R(x))} \\ \textit{GFNEvalP} &= \text{Pearson's } \rho_{\log(\pi(x)),\log(R(x))} \end{aligned} \tag{7}$$

While this metric might be informative for the learning performance of GFlowNets, it does not provide a lot of information about the skewness of a future sample from the model. Metrics that are based on the distribution of the estimated probabilities should contain such information. For the task of molecule generation we are interested in looking at the high-scoring molecules. For this we can define the following *pHighestkbins* metric, which captures whether the high scoring molecules (in the top 4 bins[2]) are more likely to be generated by the GFlowNet, with the generated samples binned into 10 equally sized bins. We can use this metric, for instance, to compare GFlowNets trained with different reward normalization coefficients, where we expect the peakier rewards to bias the model to generate higher scoring molecules.

$$\text{pHighestkbins} = \frac{\sum_{x \in \text{Top } (10 \times k)\% \text{ scoring states from } D} \pi(x)}{\sum_{x \in D} \pi(x)} \tag{8}$$

[1]In practice it might be hard to ensure a test set which contains only unseen molecules, as they could be sampled during training, even though that might be rare in the high dimensional molecule space. Here we fix a test set a priori and discard training trajectories which generate molecules from this set during training.

[2]Note that the choice of k is set to 4 in this paper which worked best empirically.

In this work, we focus on evaluating the performance of GFlowNets for the downstream molecule design task where the goal is often to propose a set of diverse and high scoring molecules. For this task, we define the *TopKDiverse* score that captures the desired properties of a generated set of samples. *TopKDiverse* represents the average score of K diverse states picked from a sorted list of generated candidates in descending order of their scores. The diversity is enforced by considering candidates from the subset of candidates within a minimum distance to the previously selected high scoring candidate in the TopKDiverse subset, and computing the average score of the molecules in the TopKDiverse set. For our experiments we compute diversity based on the Tanimoto similarity (App. A.2). We compute this metric and other sample statistics on the a set of 10k states sampled from a given GFlowNet.

4 EXPERIMENTS/ EMPIRICAL RESULTS

4.1 EXPERIMENTAL SETUP

In this section we empirically evaluate the proposed metric *GFNEval*. We evaluate the robustness of the metric on different molecule design tasks, with different GFlowNet training strategies and with access to different types of test sets. We observe that *GFNEval* is a reliable metric to track training progress and generalization, as well as derive new insights into the learning dynamics of GFlowNets using the *GFNEval* metric.

Tasks: We consider two different types of rewards: QED (Landrum) and a Neural Network Proxy which approximates the binding energy(AutoDock Vina, Trott & Olson, 2010). The latter covers a common approach used for setups where limited data is available, the in-silico simulator is expensive, and online samples are desired (Bengio et al., 2021b; Liu et al., 2020). For more details about the Proxy see A.3.

Test Data: We design our test sets to be approximately uniform based on scores by sub-sampling it from a bigger set. We generate two different sets of 300k molecules for each objective from which we sub-sample the 10k test sets.

- **Random test set** represents an ideal set, with diverse molecules sampled uniformly randomly from the state space (uniform according to molecule size and diverse based on Tanimoto similarity).
- In practice, however, it is not always be feasible to construct such a test set, and we need to rely on a dataset of known candidates, which might be sampled from different approaches. We thus consider a second test set, the **MCMC test set**, representing this more practical scenario, which is sampled using an MCMC algorithm.

Because GFlowNet's objective is to construct an object x with probability proportional to $R(x)$ we highlight the importance of having an evaluation dataset which covers the reward landscape. See App. A.2 for more details about datasets.

Training Methods:

- GFlowNet Bengio et al. (2021b) (*Base*);
- GFlowNet with a smaller learning rate (*lr*0.2*);
- GFlowNet (*exp+1*) that uses a higher exponent for shaping the reward which should result in higher sampling rate of high scoring states;
- GFlowNet (*best50%*) which constructs half of the training batch with offline trajectory samples from a replay buffer of highest scoring molecules encountered during training

See App. A.4 for more implementation details.

Sample statistics: We compute different statistics for a set of 10k states sampled from a given GFlowNet policy: *Top1000Diverse*, *Top100Diverse*, *Top1000*, *Top100* (average score of the top 100 molecules in the sampled set), *Mean Diversity* (mean pairwise Tanimoto similarity between every two states in the sample set), *Mean Score* (mean score of entire sample set). TopK scores represent the mean score of the highest K scoring states in the sample set.

Evaluation metrics: We compute different evaluation metrics on the test set: *GFNEvalS* (7), *GFNEvalP* (7), (*CE*) Cross-entropy between GFlowNet probabilities and rewards $-\sum r(x) * log(p_{GFN}(x))$, (*Flow Error*) Flow matching error $\mathcal{L}_{FM}$ (5) computed for 1 random trajectory for each state in the test set, *pHighest4bins* (8), *pHighest4bins*SP* (product of *pHighest4bins* score and *GFNEvalS*).

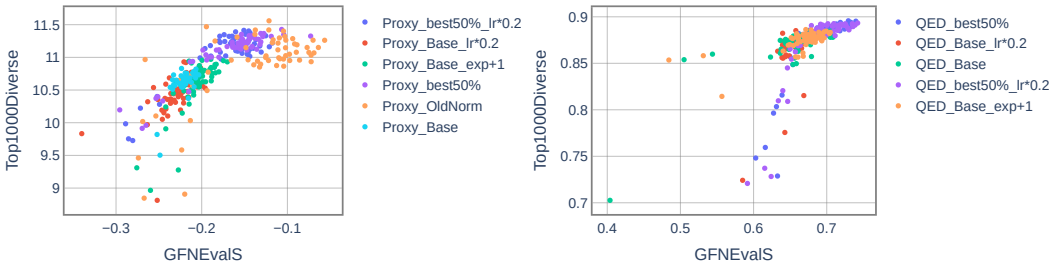
Protocol: During training we periodically evaluate the GFlowNet model at a constant frequency of 500 optimization steps for the Docking Proxy task and 250 steps for QED. We evaluate a total of 20 checkpoints per experiment with 3 random seeds per experiment, for each GFlowNet training method.

4.2 Building drug-like small molecules

Following Bengio et al. (2021b); Liu et al. (2020) we use their published environment for generating small molecules. We use a framework that allows an agent to sequentially generate molecules by parts using a predefined vocabulary of building blocks, also known as fragment-based drug design. The graph representation of the molecule is constructed as a sequence of additive edits: given a molecule and constraints of chemical validity, an atom can be chosen to attach a block to. The action space is thus the product of choosing where to attach a block and choosing which block to attach. There is an extra action to stop the editing sequence. This sequence of edits yields a DAG MDP, as there are multiple action sequences that lead to the same molecule graph. The reward is computed using scores produced by the chosen task (either QED or Docking Proxy) and a normalization scheme. See App. A.1 for more details.

4.3 Experiments

Following the protocol detailed in Sec. 4.1, we empirically demonstrate that the *GFNEvalS* metric 7 satisfies the properties of a good generalization metric (Sec. 2.2), one predictive of the desired performance metrics (i.e. finding high-scoring diverse candidates). In Fig. 1 we can observe visually the correlation between the *GFNEvalS* score for approximately 660 GFlowNet models and the *Top1000Diverse* statistic computed for 10k samples from those models. Except for the models at the beginning of training (100 optimization steps), which are scattered almost randomly in the left bottom part of the plot (see Fig.6 where models are colored based on training step), there is an almost linear trend between the two depicted metrics. In this figure we color each checkpoint based on the training method which helps us observe a significant difference in *Top1000Diverse* scores based on GFlowNet training method.



(a) Docking Proxy (b) QED

Figure 1: Relationship between *Top1000Diverse* statistic and *GFNEvalS* test score (Spearman Correlation) for checkpoints across 33 GFlowNet training runs. GflowNet Model checkpoints are colored based on training method.

To quantify the predictive power of the evaluation metrics for statistics of samples generated from GFlowNet, we can look at the correlation between them. In tables 1a and 1b (for the Docking Proxy and QED tasks respectively) we present the Spearman correlation coefficient between different sample statistics (4.1) and different evaluation metrics (4.1) across all checkpoints saved during training of several GFlowNet variations. We can see a significant correlation of 0.85 (with p-value lower

that 0.0001) between *GFNEvalS* and the *Top100Diverse* scores for both objectives (Tables 1a and 1b). We note that although *GFNEvalS* is highly correlated with the *Top100*/*Top1000* scores as well, the coefficient is slightly smaller, supporting the previous findings (Bengio et al., 2021b) of higher diversity in sampled batches using GFlowNets. We observe a similar trend on the MCMC test set as seen in Tables 3a and 3b from the Appendix. We also observe that the *Flow Error* by itself has very low correlation with all the sample statistics measured, confirming the concern that training error (TD error) is a bad indicator of performance, as discussed in Sec. 2.2. Finally, our second proposed metric *pHighest4bins* is moderately predictive of performance when comparing all GFlowNet models together.

	GFNEvalS	**GFNEvalP**	**Flow Error**	**CrossEntropy**	**pHighest4bins**	**pHighest4bins*SP**
Top 1000 Diverse	**0.85**	0.69	0.22	0.32	0.64	0.66
Top 100 Diverse	**0.84**	0.68	0.22	0.32	0.64	0.66
Top 1000	**0.83**	0.66	0.31	0.35	0.63	0.65
Top 100	**0.83**	0.67	0.27	0.33	0.63	0.65
Mean Diversity	**0.26**	0.15	0.01	0.23	0.24	0.24
Mean Score	**0.85**	0.71	0.37	0.41	0.64	0.65

(a) Docking Proxy

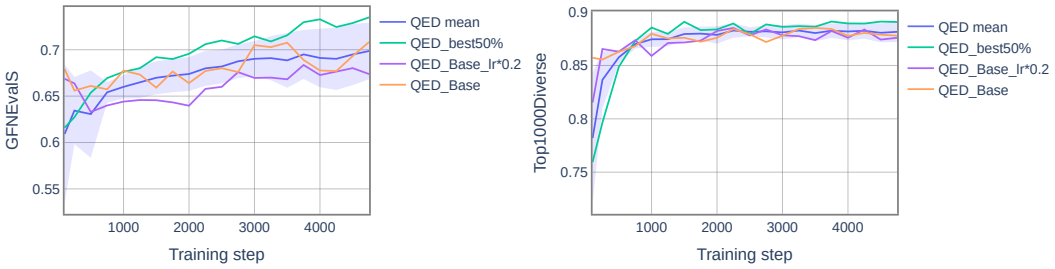
	GFNEvalS	**GFNEvalP**	**Flow Error**	**CrossEntropy**	**pHighest4bins**	**pHighest4bins*SP**
Top 1000 Diverse	**0.85**	0.85	0.17	0.55	0.77	0.81
Top 100 Diverse	0.76	0.76	0.18	0.49	0.78	**0.80**
Top 1000	0.83	**0.84**	0.21	0.52	0.81	**0.84**
Top 100	0.75	0.75	0.19	0.47	0.78	**0.81**
Mean Diversity	0.10	0.09	**0.29**	0.02	0.08	0.05
Mean Score	0.55	0.56	0.42	0.34	**0.70**	**0.70**

(b) QED

Table 1: Spearman's rank correlation coefficient between different sample statistics (rows) and different evaluation metrics (columns) computed on the **Random test sets** using the GFlowNet model used for sampling.

In addition to being predictive of the final downstream performance, we would also like the metric to be useful for tracking the progress within a training run. In Fig. 2 and 7 we show the evolution of both the best evaluation *GFlowEvalS* metric and a target sample statistic, *Top1000Diverse*, over the course of training. In Fig. 2 both metrics have a positive slope during training across experiments (represented with the blue line). We can also visually observe from 3 randomly sampled runs that the *GFNEvalS* is highly correlated within a training run with our desired sample statistic. This is confirmed by a quantitative analysis across all training runs, by calculating the correlation of our metrics only for the 20 checkpoints within each training run and reporting the mean across all runs (see App. Table 4). Although we obtain a smaller average correlation of ~ 0.64 (across tasks and test sets) this metric can still distinguish small differences in *TopKDiverse* scoring GflowNet models. In this analysis the *GFNEvalS*, is only outperformed by the *pHighest4bins*SP* metric on the QED Task with random test set, which motivates the potential of this metric.

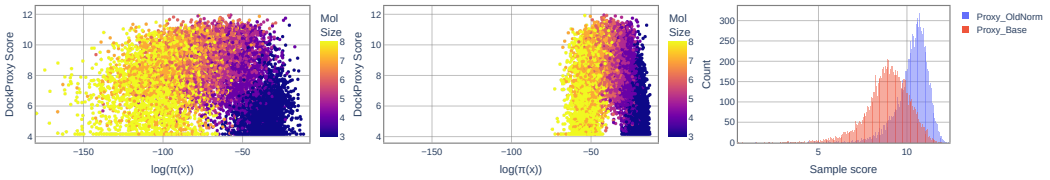
In addition to evaluating the performance of GFlowNets, we can also leverage these metrics to derive qualitative insights about the learning dynamics of GFlowNets. For instance, from Fig. 3, where data points are colored based on molecule size, we can clearly observe how probabilities are highly biased based on the number of blocks (smaller molecules have higher probabilities). This is intuitive given the sequential nature of the generation and the difficulty of obtaining unbiased $\log(\pi_\theta(x))$ given the significant difference in number of trajectories for different molecule sizes. This can help us focus more on finding the appropriate neural network architecture, or adjusting the GFlowNet training distribution uniformly across molecule sizes. This problem is more obvious with the Docking Proxy distribution of scores being influenced in such a manner by molecule size. In the case of QED (Fig. 8), where the smaller the molecule the higher the score, this observation can be easily omitted because of the coincidental relationship of higher $\pi(x)$ with smaller molecules. This plot can also give us some intuitions about the very low overall *GFNEvalS* scores for Docking Proxy (as observed in Fig. 1a) and especially the negative correlation on the lowest scoring molecules from the test (9a). Given that the model is biased in training more on higher rewarding trajectories (because of sampling), in the Docking Proxy task, the model will focus on average size molecules



(a) GFNEvalS (b) Top1000Diverse

Figure 2: Evolution during training for both *GFNEvalS* and *Top1000Diverse* scores for the QED objective. Blue line depicts the average across all training runs with the shaded region representing the standard deviation interval. Figure also contains individual lines for three randomly sampled runs.

and might get very biased probabilities for lower scoring molecules. We present further analysis of other phenomena observed in training in Appendix A.5.



(a) Docking Proxy_OldNorm (b) Docking Proxy_Base (c) Sample Score Histogram

Figure 3: Docking Proxy task. Qualitative analysis of two GflowNets from two different training methods. Left ((a) and (b)) visual representation of GFlowNets learned probabilities for the random test set states. Scatter plots depict relationship between $\log(\pi(x))$ and scores of x colored based on molecule size. Right (c) Histogram of scores for the two 10k sets sampled from those models.

From Fig. 3 we can also observe the potential pitfalls of relying on *GFNEvalS* alone and how it is not actually influenced by the distribution of $\log(\pi_\theta(x))$. This can make it harder, for example, to differentiate between models which have been trained with different reward shaping coefficients. For Fig. 3 we hand pick two extreme cases in order to demonstrate this pitfall, where two GFlowNets are trained with significantly different reward shaping. Although a checkpoint from *Proxy_OldNorm* experiment (Fig 3a) has a slightly lower *GFNEvalS* (-0.220) compared to the one from *Proxy_Base* Fig. 3b (-0.184) it has a higher *Top1000Diverse* score of 11.25 (vs 10.69 for *Proxy_Base*). In this particular case the *pHighest4bins* metric can discriminate between the two models correctly measuring higher value for *Proxy_OldNorm* (0.040) compared to *Proxy_Base* (0.024).

5 Limitations and Future Works

In this work we present a thorough evaluation of the generalization performance of GFlowNets for molecule design, using our propsed metrics *GFNEval* and *pHighestKbins*. A key limitation of the metrics presented in this paper is the cost to compute the exact probability of sampling a molecule under the GFlowNet. This can make evaluation difficult with large test sets and larger training runs. One direction to explore would be ways to approximate this quantity more efficiently, while still retaining the properties of the proposed metrics. We also demonstrate the discriminative power of the metrics within a training run. This property can be helpful in active learning settings with a GFlowNet generator (Bengio et al., 2021a) for early stopping the generator training. Future work should also focus on using these metrics for further analysis of GFlowNet learning dynamics, to make practical recommendations for training.

REFERENCES

Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International Conference on Learning Representations*, 2020.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34, 2021a.

Yoshua Bengio, Tristan Deleu, Edward J. Hu, Salem Lahlou, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations, 2021b.

Nathan Brown, Ben McKay, François Gilardoni, and Johann Gasteiger. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *Journal of chemical information and computer sciences*, 44(3):1079–1087, 2004.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs, 2018.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 2017.

Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Karam M. J. Thomas, Simon Blackburn, Connor W. Coley, Jian Tang, Sarath Chandar, and Yoshua Bengio. Learning to navigate the synthetically accessible chemical space using reinforcement learning, 2020.

Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Chapter 11. junction tree variational autoencoder for molecular graph generation. *Drug Discovery*, pp. 228–249, 2020. ISSN 2041-3211. doi: 10.1039/9781788016841-00228. URL http://dx.doi.org/10.1039/9781788016841-00228.

Ashutosh Kumar, A Voet, and KYJ Zhang. Fragment based drug design: from experimental to computational approaches. *Current medicinal chemistry*, 19(30):5128–5147, 2012.

Greg Landrum. Rdkit: Open-source cheminformatics. URL http://www.rdkit.org.

Cheng-Hao Liu, Maksym Korablyov, Stanisław Jastrzębski, Paweł Włodarczyk-Pruszyński, Yoshua Bengio, and Marwin HS Segler. Retrognn: Approximating retrosynthesis by graph neural networks for de novo drug design. *arXiv preprint arXiv:2011.13042*, 2020.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. In *International Conference on Learning Representations*, volume abs/1608.03983, 2017.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. Graphdf: A discrete flow model for molecular graph generation, 2021.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets, 2022.

Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties, 2019.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9323–9332. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/satorras21a.html.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *ICLR (Poster)*, 2016.

Ari Seff, Wenda Zhou, Farhan Damani, Abigail Doyle, and Ryan P Adams. Discrete object generation with reversible inductive construction. *arXiv preprint arXiv:1907.08268*, 2019.

Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focussed molecule libraries for drug discovery with recurrent neural networks, 2017.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Kevin Swersky, Yulia Rubanova, David Dohan, and Kevin Murphy. Amortized bayesian optimization over discrete spaces. In *Conference on Uncertainty in Artificial Intelligence*, pp. 769–778. PMLR, 2020.

Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Olivier J. Wouters, Martin McKee, and Jeroen Luyten. Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. *JAMA*, 323(9):844–853, 03 2020.

Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. {MARS}: Markov molecular sampling for multi-objective drug discovery. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=kHSu4ebxFXY.

A Appendix

A.1 Molecule domain details

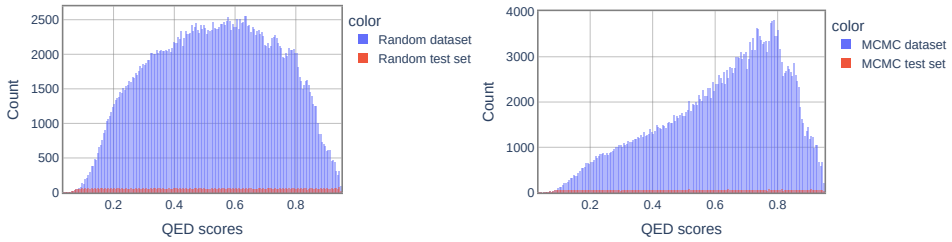
The small molecules considered in this paper are constructed from a set of drug-like building blocks, each of them consisting one or multiple atom sites for linking another building block with a single bond. These building blocks and substitution sites are chosen based on the frequency of appearance in the ligand subset of the PDB database (BRICS decomposition and Bemis-Murcko decomposition of BRICS scaffolds), following our previous works in Liu et al. (2020); Bengio et al. (2021b), where we chose 131 building blocks here. Each action modifies the resulting molecule by connecting a new block to the previous substructure. In order to correctly compute parents of a molecule graph isomorphism must be computed to disregard duplicates and also no duplicate (blocks, stems) can exist in the list of building blocks.

A.2 Dataset details

When generating the Random dataset we sample random molecules uniformly according to number of blocks (between 3 and 8) with the constraint that any two molecules in the dataset must be different according to a biologically motivated similarity metric. We choose a Tanimoto similarity of 0.7 as threshold, as it is commonly used in medicinal chemistry to find similar molecules. For generating the slightly high scoring biased dataset we use a basic MCMC algorithm with uniform kernel rejecting molecules based on their scores with a given probability. First we generate 300k molecules for each of these sets according to the previously mentioned procedure and afterwards we sample from them the corresponding 10k test set molecules relatively uniform according to scores. In this work we only use the test set molecules to evaluate GFlowNet models. Figures 5 and 4 show the histogram of scores for these datasets.

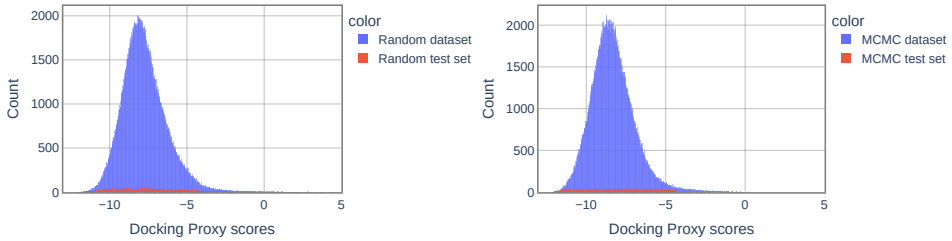
A.3 AutoDock vina Proxy details

All molecules in the dataset described in A.2 were subjected to conformer generation ($n = 30$), all of which were then optimized via a forcefield (MMFF). The lowest conformer was subsequently docked by AutoDock Vina for the sEH protein Bengio et al. (2021a); Trott & Olson (2010). All



(a) Random test set (b) MCMC test set

Figure 4: Histogram of QED scores for the two datasets sources (300k molecules) and the sampled test sets (10k molecules).



(a) Random test set (b) MCMC test set

Figure 5: Histogram of Docking Proxy scores for the two dataset sources (300k molecules) and the sampled test sets (10k molecules).

positive docking scores (indicating ligand dissociation with the protein) were clipped to 0. The average docking score is -7.78 with a standard deviation of 1.68. The proxy was trained using 280,000 molecules in the generated dataset and validated using 10,000 random molecules in the same dataset, where an additional 10,000 molecules were reserved as test set for GFlowNet experiments. We featurized the graphs with typical atom and bond featurizers, as used by Gilmer et al. (2017). The proxy was trained using the recently published E(n) Equivariant Graph Neural Network without edge inference Satorras et al. (2021). Other proxies such as MPNN Gilmer et al. (2017) were also examined but gave worse performance. The hyperparameters were optimized. We employed 3 layers each with 128 hidden units. the Adam optimizer with a learning rate of $5e-4$ and weight-decay of $1e-16$; the learning rate followed a cosine annealing schedule of $T = 100$ (epochs) Loshchilov & Hutter (2017). Early stopping was implemented based on the validation MSE. The optimized validation MSE, MAE were 1.76 and 0.82, respectively.

A.4 Implementation details

For GFlowNet training we follow the same framework and hyperparameters described in Bengio et al. (2021b). The flow predictor F uses an MPNN (Gilmer et al., 2017), which receives the block graph as input. Two of the parameters that we change for the GFlowNet training are the minimum number of blocks allowed for action stop, which is 3 instead of 2, and we use a Minibatch size of 16 trajectories per optimization step. In this work we only work with molecules that have between 3 and 8 blocks. Learning rates for training the models are either the default 5×10^{-4} for *Base* or 1×10^{-4} for the *Base_lr*0.2* experiments. Reward normalization coefficients used for the two tasks are described in the following Table 2. For the $Best50\%$ experiments we sample for each batch 8 trajectories backward from the high scoring molecules in the replay buffer and 8 (online) trajectories from the current GflowNet model.

Table 2: Reward normalization coefficients for the two objectives evaluated (Docking Proxy and QED).

	Docking Proxy	QED	Docking Proxy OldNorm
Reward T	7	5.9	8
Reward β	8.1	8.9	10

A.5 Additional Experiments

In the following tables 3 and 4 we describe the Spearman's rank correlation coefficient between different sample statistics and different evaluation metrics (See Sec. 4.1). In order to determine the robustness of our proposed metrics we evaluate two different test sets (Random and MCMC test sets) and two scoring tasks (QED and Docking Proxy). We can determine from this results that *GFNEvalS* is an appropriate generalization evaluation metric for GFlowNets, predictive of the desired search performance *TOPKDiverse*. This metric can be used to compare models from different GFlowNet training methods (with significant correlations above 0.8) or from the same training run (with a mean correlation of 0.64 across experiments). Not only is *GFNEvalS* the best metric to predict high *Top1000Diverse* scores for a sample set from the model, but is also predicting this score better than other sample statistics (e.g. *TopK*).

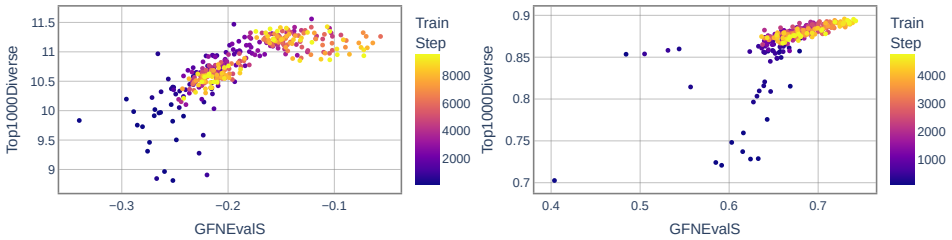
	GFNEvalS	GFNEvalP	Flow Error	CrossEntropy	pHighest4bins	pHighest4bins*SP
Top 1000 Diverse	0.86	0.84	0.19	0.39	0.67	0.69
Top 100 Diverse	0.85	0.82	0.18	0.40	0.67	0.70
Top 1000	0.85	0.84	0.27	0.41	0.66	0.69
Top 100	0.85	0.84	0.24	0.40	0.67	0.69
Mean Diversity	0.30	0.26	0.04	0.20	0.29	0.30
Mean Score	0.84	0.86	0.38	0.43	0.67	0.69

(a) Docking Proxy

	GFNEvalS	GFNEvalP	Flow Error	CrossEntropy	pHighest4bins	pHighest4bins*SP
Top 1000 Diverse	0.85	0.85	0.12	0.55	0.59	0.66
Top 100 Diverse	0.75	0.75	0.14	0.49	0.60	0.67
Top 1000	0.82	0.82	0.16	0.52	0.62	0.69
Top 100	0.75	0.75	0.15	0.48	0.60	0.66
Mean Diversity	0.11	0.10	0.25	0.00	0.17	0.13
Mean Score	0.53	0.53	0.36	0.35	0.62	0.64

(b) QED

Table 3: Spearman's rank correlation coefficient between different sample statistics (rows) and different evaluation metrics (columns) computed on the **MCMC test set** using the model used for sampling.



(a) AutoDockVina Proxy (b) QED

Figure 6: *Top1000Diverse* vs *GFNEvalS* (Spearman Correlation) for 660 GFlowNet models. Data points colored based on training step.

From Fig. 7 we can observe that the average *GFNEvalS* across experiments has the same curvature as the average *Top1000Diverse* scores during training, even reproducing a slight decrease at the end

	GFNEvalS	GFNEvalP	Flow Error	CrossEntropy	pHighest4bins	pHighest4bins*SP
Top 1000 Diverse	0.58	0.39	0.03	0.23	0.40	0.41
Top 100 Diverse	0.55	0.37	0.02	0.25	0.39	0.40
Top 1000	0.54	0.38	0.02	0.18	0.42	0.43
Top 100	0.53	0.36	0.00	0.21	0.41	0.42
Mean Diversity	0.28	0.10	0.08	0.21	0.18	0.19
Mean Score	0.59	0.54	0.10	0.11	0.41	0.43

(a) Docking Proxy - Random test set

	GFNEvalS	GFNEvalP	Flow Error	CrossEntropy	pHighest4bins	pHighest4bins*SP
Top 1000 Diverse	0.61	0.56	0.01	0.28	0.48	0.48
Top 100 Diverse	0.60	0.54	0.00	0.31	0.48	0.49
Top 1000	0.57	0.52	0.06	0.24	0.50	0.51
Top 100	0.58	0.52	0.02	0.29	0.49	0.50
Mean Diversity	0.39	0.24	0.09	0.20	0.31	0.32
Mean Score	0.47	0.55	0.19	0.08	0.48	0.49

(b) Docking Proxy - MCMC test set

	GFNEvalS	GFNEvalP	Flow Error	CrossEntropy	pHighest4bins	pHighest4bins*SP
Top 1000 Diverse	0.69	0.68	0.19	0.44	0.68	0.71
Top 100 Diverse	0.57	0.56	0.18	0.38	0.68	0.69
Top 1000	0.66	0.65	0.23	0.38	0.73	0.75
Top 100	0.57	0.56	0.18	0.37	0.69	0.70
Mean Diversity	0.12	0.14	0.30	0.01	0.26	0.26
Mean Score	0.55	0.57	0.38	0.20	0.70	0.71

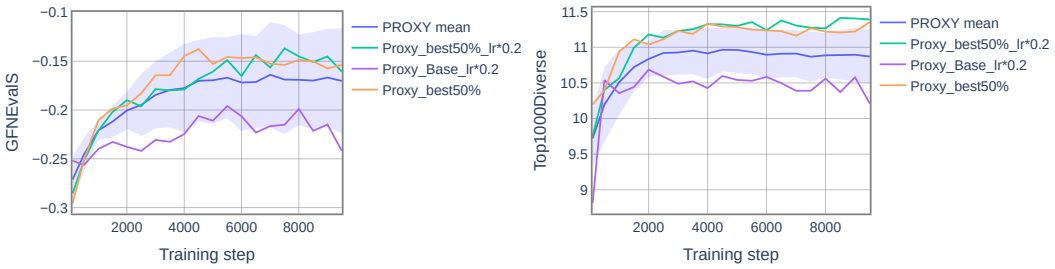
(c) QED - Random test set

	GFNEvalS	GFNEvalP	Flow Error	CrossEntropy	pHighest4bins	pHighest4bins*SP
Top 1000 Diverse	0.68	0.68	0.17	0.40	0.54	0.58
Top 100 Diverse	0.57	0.56	0.16	0.35	0.53	0.56
Top 1000	0.64	0.63	0.20	0.35	0.59	0.62
Top 100	0.57	0.56	0.16	0.34	0.52	0.55
Mean Diversity	0.06	0.06	0.25	0.05	0.21	0.20
Mean Score	0.49	0.49	0.33	0.21	0.58	0.59

(d) QED - MCMC test set

Table 4: Mean Spearman's rank correlation coefficient across training runs. Correlation between different sample statistics (rows) and different evaluation metrics (columns) computed on the two test sets using the model used for sampling.

of training. Also correlation between the metrics is strong within the individual training runs across training steps and between experiments.



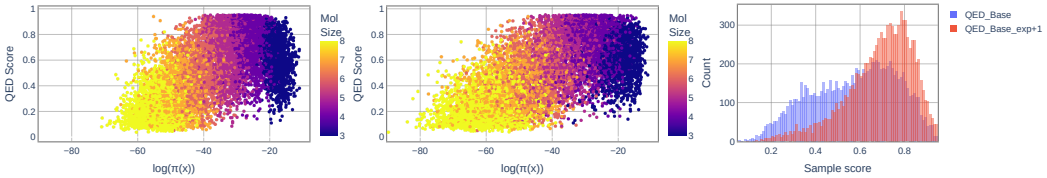
(a) GFNEvalS (b) Top1kDiverse

Figure 7: Evolution during training for both *GFNEvalS* and *Top1000Diverse* scores for the Docking Proxy objective. Blue line depicts the average across all training runs with the shaded region representing the standard deviation interval. Figure also contains individual lines for three randomly sampled runs.

In Fig. 8 we present visually the probabilities of sampling the test set states (Fig. 8a and 8b) and the histogram of scores for a 10k sample set (Fig. 8c) for two GFlowNet models. Their scores are listed in Table 5. Both *GFNEvalS* and *pHighest4bins*SP* are higher for model *QED_Base_exp+1* which indicate a higher *Top1000Diverse* score. This can be also expected from the negatively skewed distributions of scores in Fig. 8c (red).

Table 5: Scores for the two GFN models from Fig. 8 trained on the QED task.

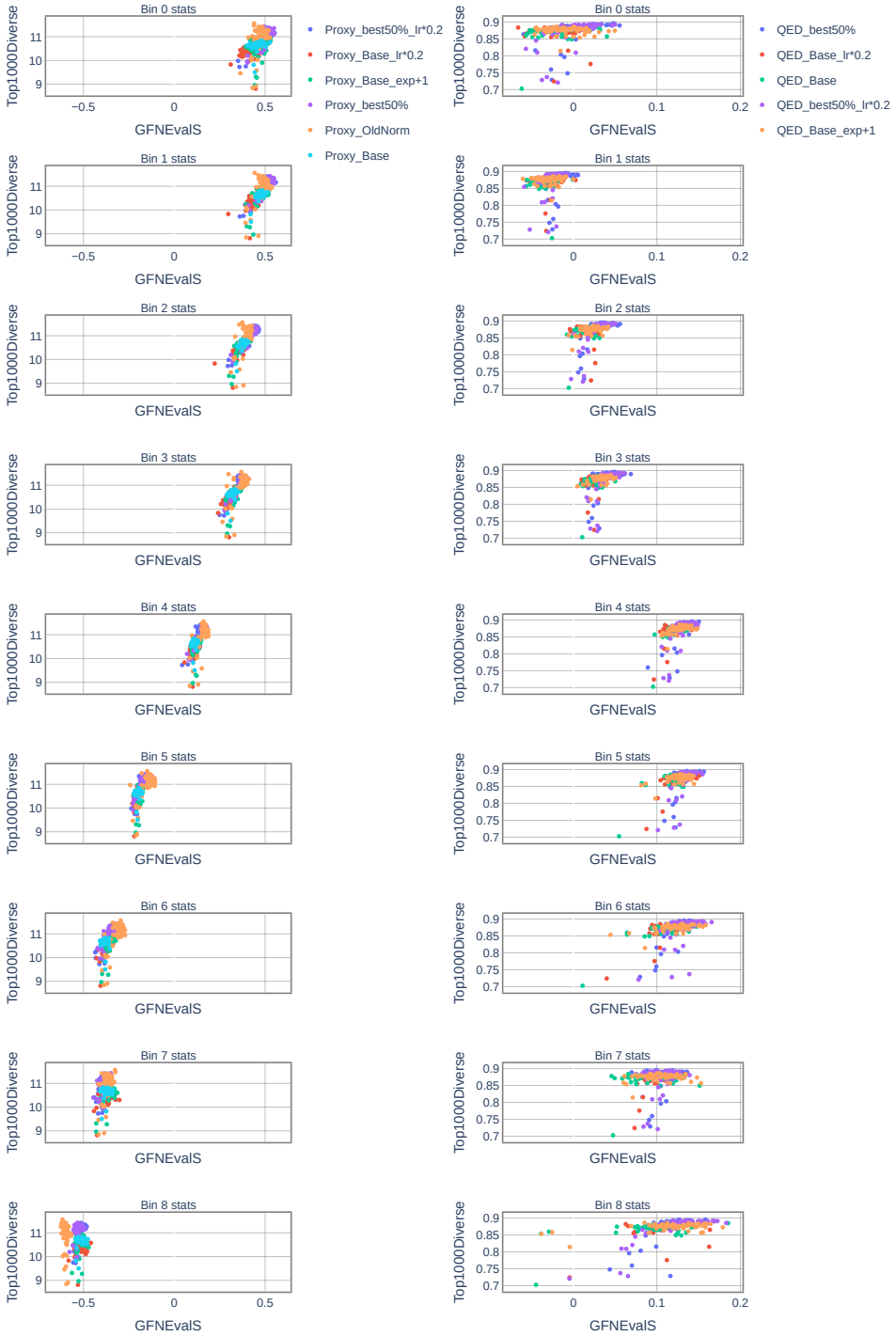
	Top 1000 Diverse	**GFNEvalS**	**pHighest4bins*SP**
QED_Base	0.856	0.656	0.471
QED_Base_exp+1	0.886	0.671	0.555



(a) QED_Base (b) QED_Base_exp+1 (c) Sample Score Histogram

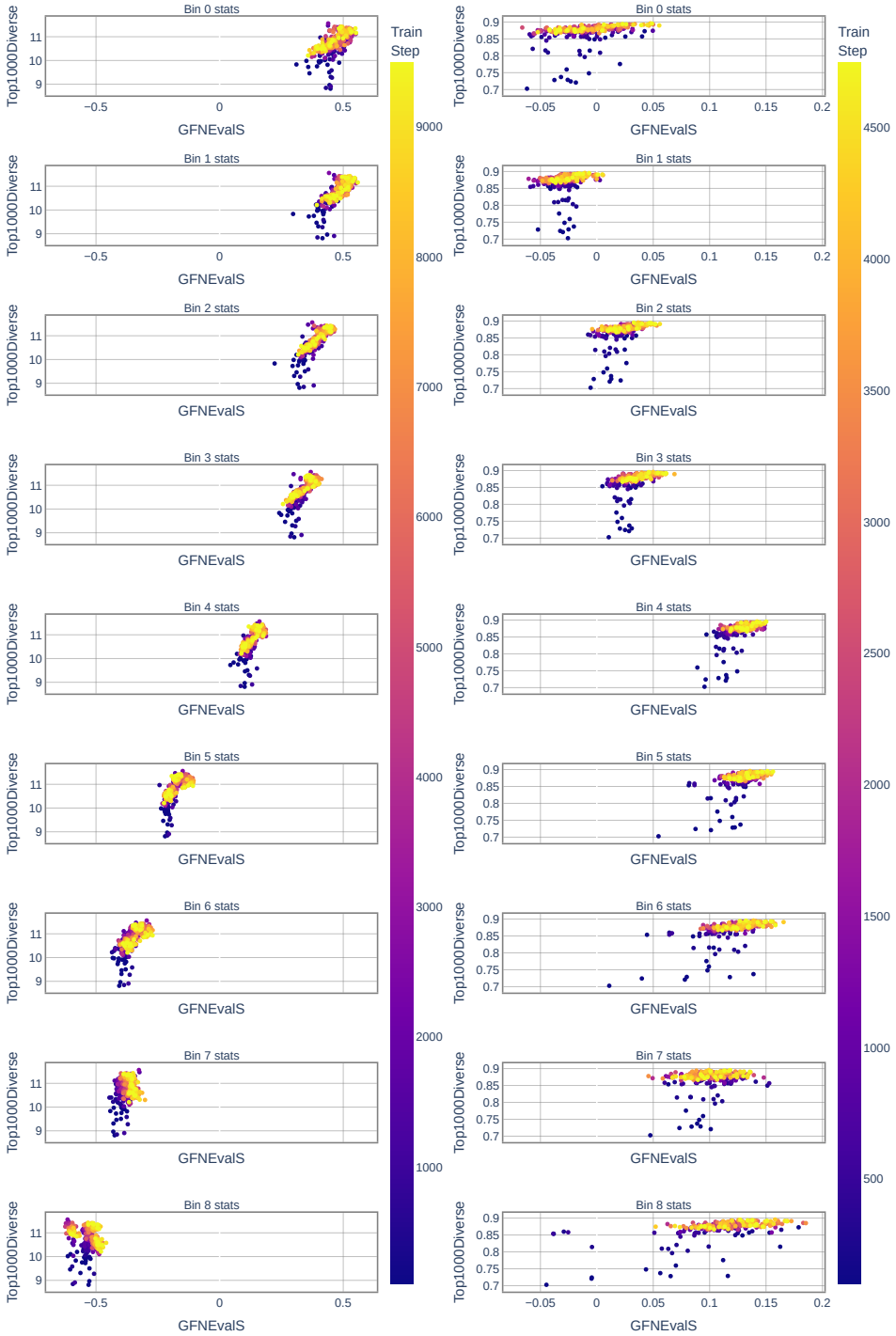
Figure 8: QED Task. Qualitative analysis of two model checkpoints from two different training variations. Left ((a) and (b)) visual representation of GFlowNets learned probabilities for the random test set states. Scatter plots depict relationship between $\log(\pi_\theta(x))$ and scores of x colored based on molecule size. Right (c) Histogram of scores for the two 10k sets sampled from those models.

Figures 9 and 10 show the relationship of the *Top100Diverse* scores for a 10k sample and the *GFNEvalS* score computed for 10 subsets from the evaluation set grouped by scoring bin.



(a) AutoDockVina Proxy (b) QED

Figure 9: Relationship between *Top100Diverse* scores and per bin *GFNEvalS* score colored based on Experiment. Bin 0 contains the highest scoring molecules.



(a) AutoDockVina Proxy (b) QED

Figure 10: Relationship between *Top100Diverse* scores and per bin *GFNEvalS* score colored based on Training step. Bin 0 contains the highest scoring molecules.