On Divergence Measures for Training GFlowNets

Tiago da Silva, Eliezer da Silva, Diego Mesquita

Keywords — GFlowNets, Variational Bayesian inference

TL;DR
- we revisit the relationship between GFlowNets and VI in continuous spaces,
- we empirically demonstrate that the well-known difficult of training GFlowNets by minimizing traditional divergence measures arises from the large gradient variance of the corresponding stochastic learning objectives,
- we develop variance-reduced gradient estimators for $\alpha-$ and KL-divergences,
- we verify that the resulting learning algorithms significantly improve upon conventional log-squared losses in terms of convergence speed,
- we re-open the once-dismissed research line focused on VI-inspired algorithmic improvements of GFlowNets.

I. Background: GFlowNets

GFlowNets are amortized algorithms for sampling from distributions over compositional objects, i.e., over objects that can be sequentially constructed from an initial state through the application of simple actions (e.g., graphs via edge-addition).

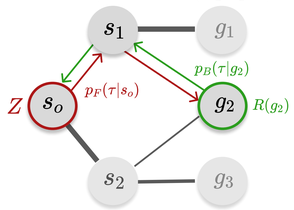

Figure 1: For finite state spaces, the state graph might be represented as a DAG on $\mathcal{S}$.

In a nutshell, a GFlowNet is composed of two three ingredients.
1. An extension $\mathcal{S}$ of the target distribution support's $\mathcal{X}$.
2. A measurable pointed DAG $\mathcal{G}$ on $\mathcal{S}$ dictating how the states in $\mathcal{S}$ are connected to one another. We refer to $\mathcal{G}$ as the state graph.
3. A forward and backward policies defining the stochastic transitions within $\mathcal{G}$.

Our objective is to learn a forward $p_F(\tau)$ and a backward $p_B(\tau|x)$ policies such that the marginal of $p_F(\tau)$ over $\mathcal{X}$ matches a given unnormalized density $r : \mathcal{X} \to \mathbb{R}_+$.

$$p_F(\tau) = \prod_{(s,s')\in\tau} p_F(s' \mid s) \text{ and } \int_{\mathcal{T}} 1_{\tau \rightsquigarrow x} p_F(\tau) \mathrm{d}\tau = r(x); \tag{1}$$

$\mathcal{T}$ denotes the space of trajectories in $\mathcal{G}$ and $\tau \rightsquigarrow$, the event in which τ finishes on $x \in \mathcal{X}$.

II. Background: GFlowNets and VI

To ensure that Equation (1) is satisfied, we parameterize $p_F(\tau)$ with a neural network and search for a parameter configuration satisfying

$$p_F(\tau) \propto p_B(\tau|x) r(x) \tag{2}$$

for every trajectory τ finishing on x. This condition is sufficient for Equation (1); indeed,

$$\int_{\mathcal{T}} 1_{\tau \rightsquigarrow x} p_F(\tau) \mathrm{d}\tau = \int_{\mathcal{T}} 1_{\tau \rightsquigarrow x} p_B(\tau|x) r(x) \mathrm{d}\tau = r(x) \underbrace{\int_{\{\tau : \tau \rightsquigarrow x\}} p_B(\tau|x) \mathrm{d}}_{=1} = r(x). \tag{3}$$

When $p_B(\tau|x)$ is fixed, Equation (2) corresponds to a standard VI problem having $p_F(\tau)$ as proposal and $p_B(\tau) :\propto p_B(\tau|x) r(x)$ as target. As such, we solve

$$\operatorname{argmin}_{p_F \in \mathcal{H}} D(p_F(\tau), p_B(\tau)) \tag{4}$$

for a given divergence measure D and a hypothesis space $\mathcal{H}$ for the policy networks.

D is conventionally set as the expected log-square difference between p_F and p_B,

$$\underbrace{\mathcal{L}_{TB}(p_F(\tau), p_B(\tau|x))}_{\text{Trajectory balance objective}} := D(p_F(\tau), p_B(\tau|x)) = \underbrace{\mathbb{E}_{\tau\sim q}}_{\text{Off-policy sampling}} \left[\left(\log \frac{Z p_F(\tau)}{p_B(\tau)} \right)^2 \right], \tag{5}$$

which requires learning the constant Z corresponding to the partition function of r.

Attempts to utilize traditional divergence measures, such as the $\alpha-$ and KL-divergences, for training GFlowNets have failed. Our objectives are to **understand the reason for this** and **improve the effectiveness of these learning objectives**.

III. KL-, Renyi-α, and Tsallis-α divergences

We investigate the training of GFlowNets with four different divergence measures: forward KL, reverse KL, Renyi-α and Tsallis-α, which we recall in the next definitions.

Forward and reverse Kullback-Leibler divergences are respectively defined as

$$\mathcal{D}_{\mathrm{KL}}(p_B(\tau) \parallel p_F(\tau)) = \mathbb{E}_{\tau\sim p_B(\tau)} \left[\log \frac{p_B(\tau)}{p_F(\tau)} \right] \text{ and } \mathcal{D}_{\mathrm{KL}}(p_F(\tau) \parallel p_B(\tau)) = \mathbb{E}_{\tau\sim p_F(\tau)} \left[\log \frac{p_F(\tau)}{p_B(\tau)} \right]. \tag{6}$$

Renyi-α divergence is defined as

$$\mathcal{R}_\alpha(p_F(\tau) \parallel p_B(\tau)) = \frac{1}{\alpha - 1} \log \int_{\mathcal{T}} p_F(\tau)^\alpha p_B(\tau)^{1-\alpha} \mathrm{d}\tau. \tag{7}$$

Tsallis-α divergence is defined as

$$\mathcal{T}_\alpha(p_F(\tau) \parallel p_B(\tau)) = \frac{1}{\alpha - 1} \left(\int_{\mathcal{T}} p_F(\tau)^\alpha p_B(\tau)^{1-\alpha} \mathrm{d}\tau - 1 \right). \tag{8}$$

Importantly, we can only evaluate each of these divergences up to a (multiplicative or additive) constant, as $p_B(\tau)$ cannot be directly computed. We demonstrate that, for the adaptive gradient-based optimization algorithms utilized in practice, this is not an issue.

Let θ be the parameters of the neural network parameterizing p_F and

$$b(\tau;\theta) = \frac{p_F(\tau)}{p_B(\tau|x) r(x)} \text{ and } s(\tau;\theta) = \log p_F(\tau). \tag{9}$$

Then,

$$\nabla_\theta \mathcal{D}_{\mathrm{KL}}(p_B(\tau), p_F(\tau)) \overset{C}{=} -\mathbb{E}_{\tau\sim p_F(\tau)}[b(\tau;\theta) \nabla_\theta s(\tau;\theta)] \text{ and} \tag{10}$$

$$\nabla_\theta \mathcal{D}_{\mathrm{KL}}(p_F(\tau), p_B(\tau)) = \mathbb{E}_{\tau\sim p_F(\tau)}[\nabla_\theta s(\tau;\theta) + \log b(\tau;\theta) \nabla_\theta s(\tau;\theta)], \tag{11}$$

in which $\overset{C}{=}$ denotes equality up to a multiplicative constant.

For the Renyi-α and Tsallis-α divergences, we similarly derive

$$\nabla_\theta \mathcal{R}_\alpha(p_F(\tau) \parallel p_B(\tau)) = \frac{\mathbb{E}[\nabla_\theta b(\tau;\theta)^{\alpha-1} + b(\tau;\theta)^{\alpha-1} \nabla_\theta s(\tau;\theta)]}{(\alpha-1)\mathbb{E}[b(\tau;\theta)^{\alpha-1}]} \tag{12}$$

and

$$\nabla_\theta \mathcal{T}_\alpha(p_F(\tau) \parallel p_B(\tau)) \overset{C}{=} \frac{\mathbb{E}[\nabla_\theta b(\tau;\theta)^{\alpha-1} + b(\tau;\theta)^{\alpha-1} \nabla_\theta s(\tau;\theta)]}{(\alpha-1)}. \tag{13}$$

IV. Variance-reduced gradient estimation of divergences

To train a GFlowNet, we stochastically update $p_F(\tau)$ with Monte Carlo estimates of the above gradients. However, this suffers from high variance and training instability; see Fig. 2.

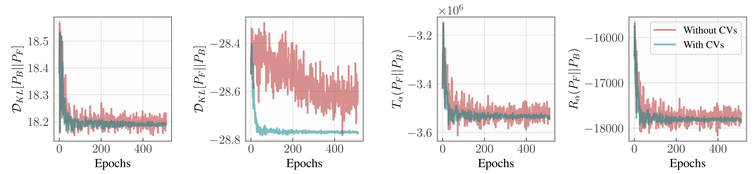

Figure 2: Monte Carlo estimation of the gradients leads to highly unstable training.

To mitigate these issues, we devise control variates (CVs) for provable variance-reduced and unbiased gradient estimation of divergence measures in the context of GFlowNets.

A CV is a zero-expectation random variable. As our objective is to estimate $\mathbb{E}_{\tau\sim p_F(\tau)}[f(\tau)]$ for some f, we introduce $\nabla_\theta \log p_F(\tau)$ as our CV and select a baseline a s.t.

$$f(\tau) - a \cdot \nabla_\theta \log p_F(\tau) \tag{14}$$

has minimum variance. We demonstrated that the optimal choice for a is

$$a^\star = \operatorname{argmin}_a \operatorname{Tr} \operatorname{Cov}\left[f(\tau) - a \cdot \nabla_\theta \log p_F(\tau) \right] = \frac{\mathbb{E}_{p_F(\tau)}\left[(\nabla_\theta \log p_F(\tau))^T \left(f(\tau) - \mathbb{E}_{p_F(\tau')}(f(\tau')) \right) \right]}{\mathbb{E}_{p_F(\tau)}\left[(\nabla_\theta \log p_F(\tau))^T (\nabla_\theta \log p_F(\tau)) \right]}.$$

A Monte Carlo approximation of this quotient cannot be written as a Jacobian-vector product. Hence, it **cannot be efficiently computed in autodiff frameworks**. To circumvent this, we utilize the first-order delta method to obtain a linear approximation to $a^\star$,

$$\hat{a} = \frac{\langle \sum_{n=1}^N \nabla_\theta s(\tau_n;\theta), \sum_{n=1}^N \nabla_\theta f(\tau_n) \rangle}{\varepsilon + \| \sum_{n=1}^N \nabla_\theta s(\tau_n;\theta) \|^2} \tag{15}$$

(given a batch $\{\tau_1, ..., \tau_N\}$ of trajectories). As $\nabla_\theta f(\tau_n)$ and $\nabla_\theta s(\tau_n;\theta)$ are naturally computed by standard learning algorithms, the computational overhead imposed by $\hat{a}$ is negligible.

Leave-one-out (LOO) estimator. We also construct a sample-dependent baseline a based on the LOO estimator. Given a batch $\{\tau_1, ..., \tau_N\}$ of trajectories, we let

$$a(\tau_i) = \frac{1}{N-1} \sum_{n=1, n\neq i}^N f(\tau_n). \tag{16}$$

Clearly, the i.i.d.-ness of the trajectories ensures the unbiasedness of the resulting estimator, which can be efficiently computed in an autodiff framework through the formula

$$\nabla_\theta \frac{1}{N} \left\langle \mathrm{sg}\left(\boldsymbol{f} - \frac{1}{N-1}(1 - \boldsymbol{I})\boldsymbol{f} \right), \boldsymbol{p} \right\rangle, \tag{17}$$

in which $\boldsymbol{f} = (f(\tau_i))_{i=1}^N$ and $\boldsymbol{p} = (\nabla_\theta s(\tau_i;\theta))_{i=1}^N$ are computed during training.

For a REINFORCE-based expectation of the form $\mathbb{E}[\nabla_\theta f(\tau) + f(\tau) \nabla_\theta \log p_F(\tau)]$, we utilize a sample-invariant baseline for the first term and a LOO baseline for the second term.

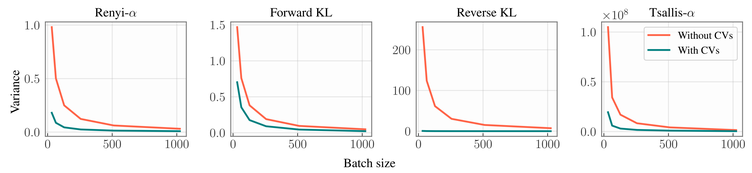

Figure 3: Our proposed CVs drastically reduce the gradient variance.

V. Experimental analysis

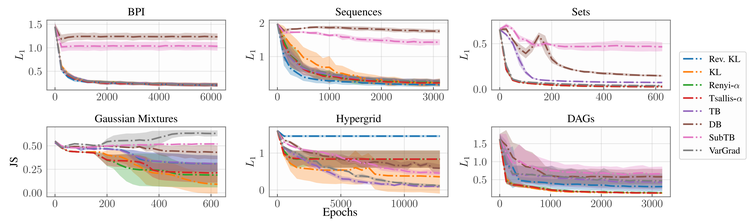

Figure 4: Our proposed CVs drastically reduce the gradient variance.

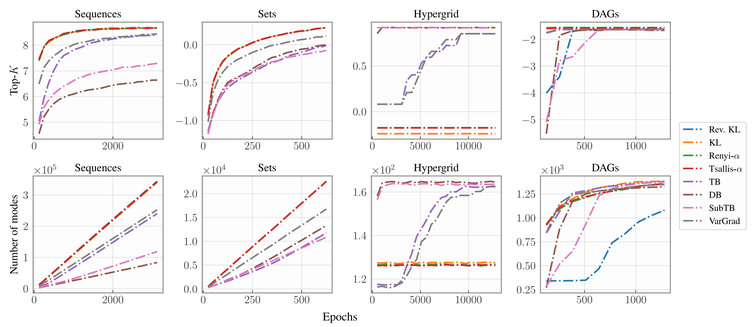

Figure 5: Our proposed CVs drastically reduce the gradient variance.

Our proposed estimators lead to faster training convergence (in terms of the TV distance between the sampling and target distributions) and mode coverage wrt traditional log-squared-based learning objectives across a broad range of generative tasks.

VI. What lies ahead?

Our work narrows the gap between GFlowNets and VI and paves the way for the development of VI-based enhancements of GFlowNet training. We believe that the design of importance-weighted learning objectives is a promising direction for future works.

On-policy vs. off-policy learning. Albeit effective, divergence-based objectives are constrained to either on-policy or simple off-policy (e.g., ε-greedy) sampling strategies, which hampers the performance of the trained GFlowNet for highly sparse target distributions. Choosing a loss function is a problem that should be considered in a case-by-case basis.