When do GFlowNets learn the right distribution?

Diego Mesquita

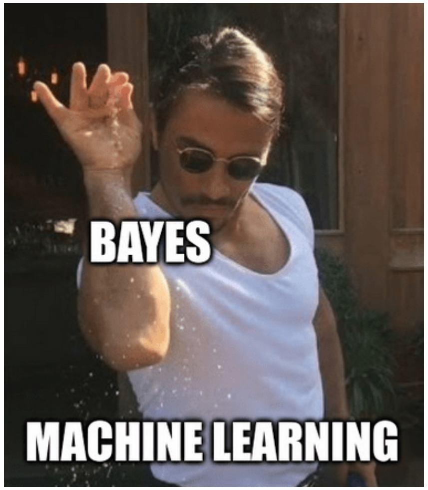
FGV EMAp

Quick introduction

Diego Mesquita

2012-2017: BSc+MSc in CS @UFC (Ceará, Brasil)
2017-2021: PhD in CS @Aalto university (Finland)
2022 onwards: Faculty @FGV EMAp (RJ, Brasil)

GREEN AI LAB



Fundamental research on Machine Learning on diverse subtopics, often with Bayesian sprinkles

Contact info

diego.mesquita@fgv.br

weakly-informative.github.io

X: @wkly_infrmtive

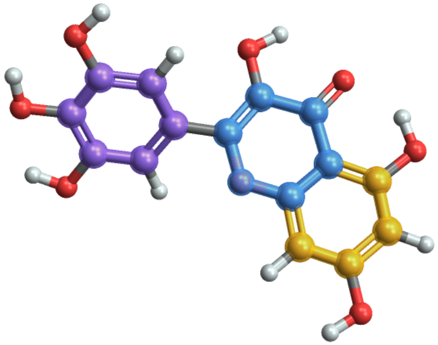
Generative Flow Networks

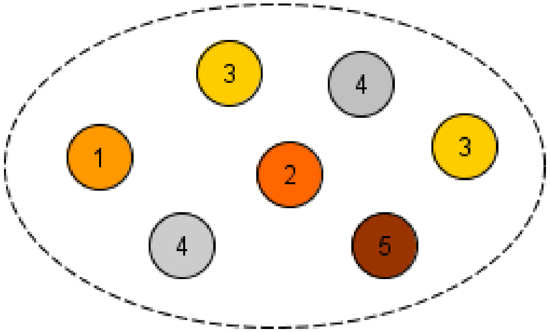
(background)

Deep generative model tailored to sample from distributions over compositional objects, i.e.:

$$\pi(x) \propto R(x) \quad \forall x \in \mathcal{X}$$

Promises non-asymptotic convergence guarantees!





Flow Network based Generative Models for Non-iterative Diverse Candidate Generation. Bengio et al., ICML 2021.

GFlowNet foundations. Bengio et al., JMLR 2023.

Generative Flow Networks

(background)

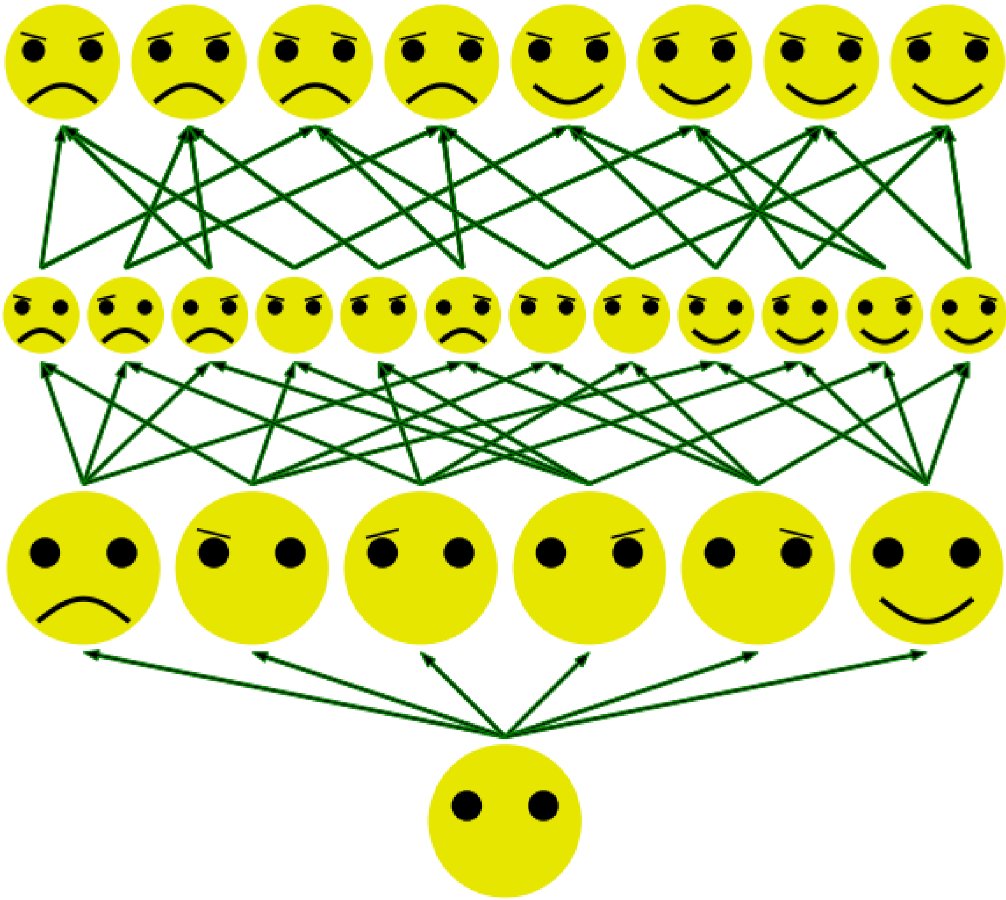


Figure from Emannuel Bengio's tutorial: https://colab.research.google.com/drive/1fUMwgu2OhYpQagpzU5mhe9_Esib3Q2VR

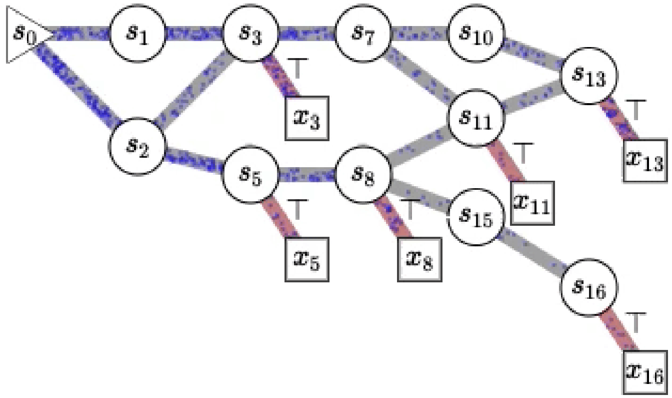
Generative Flow Networks

(background)

Core idea: cast sampling as a network flow problem

$$\sum_{s'} F(s,s') = \sum_{s'} F(s',s) \quad \forall s \in \mathcal{S} - \mathcal{X}$$

$$\sum_{s} F(s,x) = R(x) \quad \forall x \in \mathcal{X}$$



Then, we can define the forward state transition policy

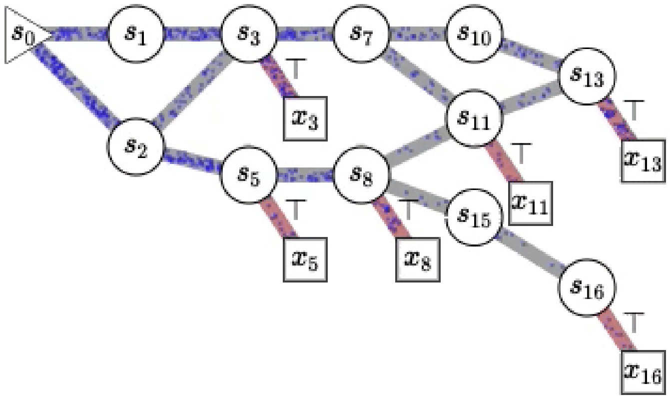
$$p_F(s'|s) = \frac{F(s,s')}{\sum_{s''} F(s,s'')}$$

Figure from Yoshua Bengio's blogpost: https://yoshuabengio.org/2022/03/05/generative-flow-networks/

Generative Flow Networks

(background)

$$p_F(s'|s) = \frac{F(s,s')}{\sum_{s''} F(s,s'')} \qquad p_F(\tau) = \prod_{(s,s')\in\tau} p_F(s'|s) \qquad p^{\top}(x) = \sum_{\tau \rightsquigarrow x} p_F(\tau) \propto R(x)$$



As long as, for any fully-supported μ:

$$F = \arg\min_F \mathbb{E}_{s\sim\mu}[\mathcal{L}(F,R,s)]$$

$$\mathcal{L}(F,R,s) = \begin{cases} \left(\log \sum_{s'} F(s,s') - \log \sum_{s'} F(s',s)\right)^2 & \text{if } s \notin \mathcal{X} \\ \left(\log R(s) - \log \sum_{s'} F(s',s)\right)^2 & \text{otherwise} \end{cases}$$

Figure from Yoshua Bengio's blogpost: https://yoshuabengio.org/2022/03/05/generative-flow-networks/

Generative Flow Networks

(practical considerations)

Trajectory balance $$p_F(\tau;\phi_F) = Z_{\phi_Z}^{-1} R(x) \prod p_B(s', s; \phi_B)$$

$$\mathcal{L}_{TB}(\tau, \phi_F, \phi_B, \phi_Z) = \left(\log Z_{\phi_Z} - \log R(x) + \sum_{s \to s' \in \tau} \log \frac{p_F(s, s'; \phi_F)}{p_B(s', s; \phi_B)} \right)^2$$

All policy and flow networks must be appropriate neural networks!

We integrate DB over transitions and TB over trajectories.

Generative Flow Networks

(practical considerations)

Trajectory balance $$p_F(\tau;\phi_F) = Z_{\phi_Z}^{-1} R(x) \prod p_B(s', s; \phi_B)$$

$$\mathcal{L}_{TB}(\tau, \phi_F, \phi_B, \phi_Z) = \left(\log Z_{\phi_Z} - \log R(x) + \sum_{s \to s' \in \tau} \log \frac{p_F(s, s'; \phi_F)}{p_B(s', s; \phi_B)} \right)^2$$

Detailed balance
$$F(s;\phi_S) p_F(s, s'; \phi_F) = F(s'; \phi_S) p_B(s', s; \phi_B) \quad \forall s \in \mathcal{S} - \mathcal{X}$$
$$F(s;\phi_S) p_F(s_f | s; \phi_F) = R(s) \quad \forall s \in \mathcal{X}$$

$$\mathcal{L}_{DB}(s, s', \phi_F, \phi_B, \phi_S) = \begin{cases} \left(\log \frac{p_F(s,s';\phi_F)}{p_B(s',s;\phi_B)} + \log \frac{F(s;\phi_S)}{F(s';\phi_S)} \right)^2 & \text{if } s' \neq s_f, \\ \left(\log \frac{F(s;\phi_S) p_F(s_f|s;\phi_F)}{R(s)} \right)^2 & \text{otherwise.} \end{cases}$$

All policy and flow networks must be appropriate neural networks!

We integrate DB over transitions and TB over trajectories.

When do GFlowNets learn the right distribution?

Tiago da Silva, Eliezer Silva, Rodrigo Alves, Amauri Souza,
Samuel Kaski, Vikas Garg, Diego Mesquita

When do GFlowNets learn the right distribution?

The GFlowNet literature is still in its infancy, with many open theoretical questions (with clear practical repercussions), e.g.:

How sensitive are GFlowNets to balance violations?

Are there cases when balance never be achieved?

How to correctly diagnose a GFlowNet's sampling quality?

Propagation of errors in GFlowNets

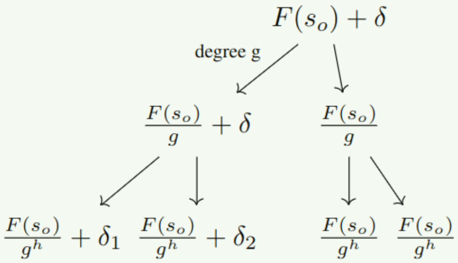
How do balance violations influence our approximation?

What happens when we introduce extra flow near the root?

Tree-structured SG + uniform traget

$$\epsilon\left(\delta, g, F(s_0)\right) \leq \mathrm{TV}\left(\tilde{p}_T, \pi\right) \leq \epsilon\left(\delta, g^h, F(s_0)\right)$$

$$\text{with } \epsilon(\delta, x, t) := \left(1 - \frac{1}{x}\right)\frac{\delta}{t+\delta}$$



irreducible error from *safe* branches: $\frac{1}{2}\left(1 - \frac{1}{g}\right)\frac{\delta}{F+\delta}$

reamaining error: $\frac{1}{2}\frac{1}{g^h(F+\delta)}\sum_{i=1}^{g^{h-1}}\left|g^h\delta_i - \delta\right|$

maximum $\rightarrow$ upper bound

minimum $\rightarrow$ lower bound

Propagation of errors in GFlowNets

Overall, the potential for damage increases with height, which is a proxy for the amount of mass below and edge.

General SGs and target distributions

$$\frac{\delta}{F(s_0)+\delta}\left(1-\sum_{x\in\mathcal{D}_{s^\star}}\pi(x)\right)\leq \mathrm{TV}\left(\tilde{p}_T,\pi\right)\leq \frac{\delta}{F(s_0)+\delta}\left(1-\min_{x\in\mathcal{D}_{s^\star}}\pi(x)\right),$$

where $\mathcal{D}_{s^\star}$ denotes the set of terminal descendants of $s^\star$.

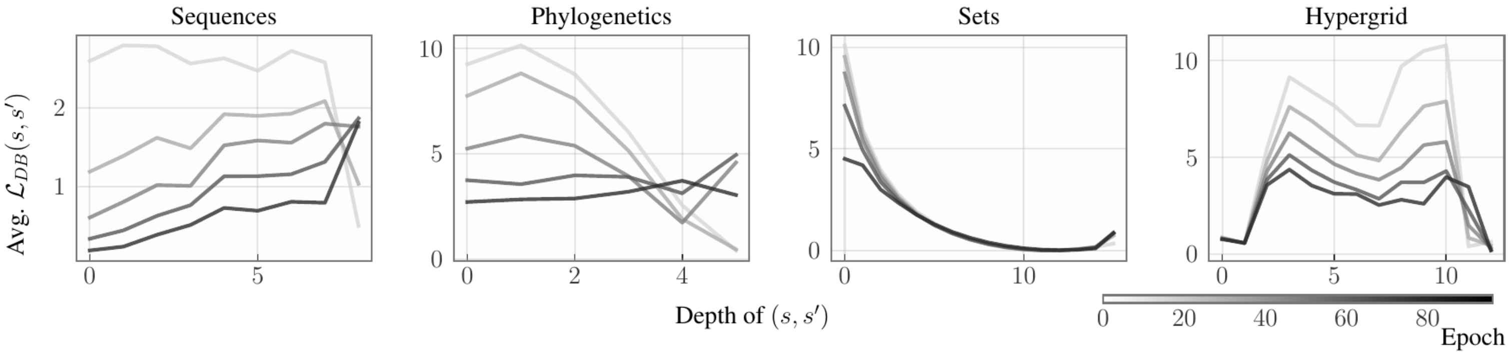
The **upper bound *intuitively* increases with the number of descendants**, as the likelihood of seeing smaller terminal rewards increases

Breaking down the DB loss

Our results suggest early-state transitions have a higher potential impact on the TV. **What happens in practice?**

$$\mathcal{L}_{\mathrm{DB}}(p_F, p_B, F) = \mathbb{E}_\tau \left[\frac{1}{\#\tau} \sum_{(s,s') \in \tau} \left(\log \frac{F(s) p_F(s'|s)}{F(s') p_B(s|s')} \right)^2 \right]$$

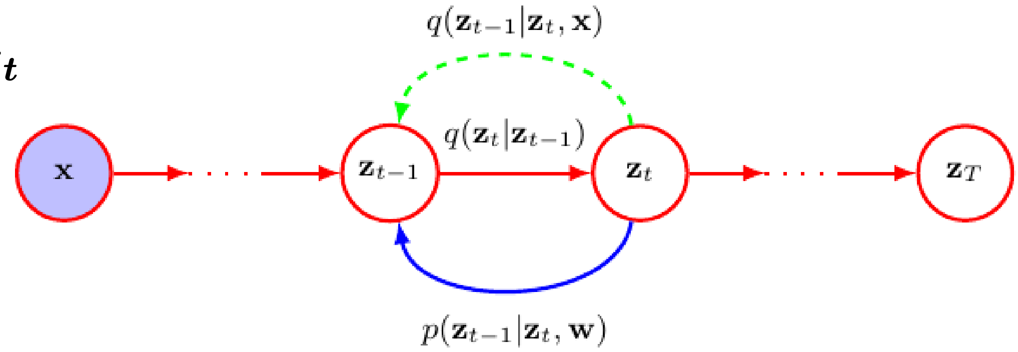
Transitions close to the initial state dominate the DB loss



Inspiration from discrete DDPMs

$$\boldsymbol{z}_t = \sqrt{1-\beta_t}\boldsymbol{z}_{t-1} + \sqrt{\beta_t}\boldsymbol{\varepsilon}_t$$

$$\beta_1 < \beta_2 < \ldots < \beta_T$$



$$\mathcal{L}(\boldsymbol{w}) = \int q(\boldsymbol{z}_1|\boldsymbol{x}) \log p(\boldsymbol{x}|\boldsymbol{z}_1, \boldsymbol{w})\, d\boldsymbol{z}_1 - \sum_{t=2}^{T} \frac{1}{2\beta_t} \int \|m_t(\boldsymbol{x}, \boldsymbol{z}_t) - \mu(\boldsymbol{z}_t, \boldsymbol{w}, t)\|^2 q(\boldsymbol{z}_t|\boldsymbol{x})\, d\boldsymbol{z}_t$$

As we get far from $\boldsymbol{x}$, β_t increases and the respective loss term turns less relevant.

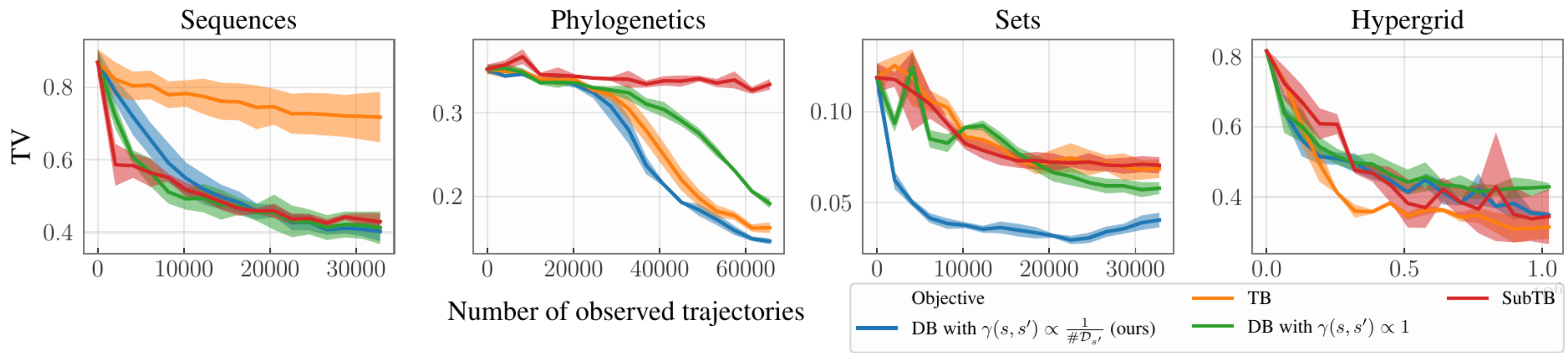
Figure from Chris & Hugh Bishop's deep learning book: bishopbook.com

Weighing the DB loss

To alleviate the emphasis on early transitions, we weigh down transitions reaching larger portions of the reward's support

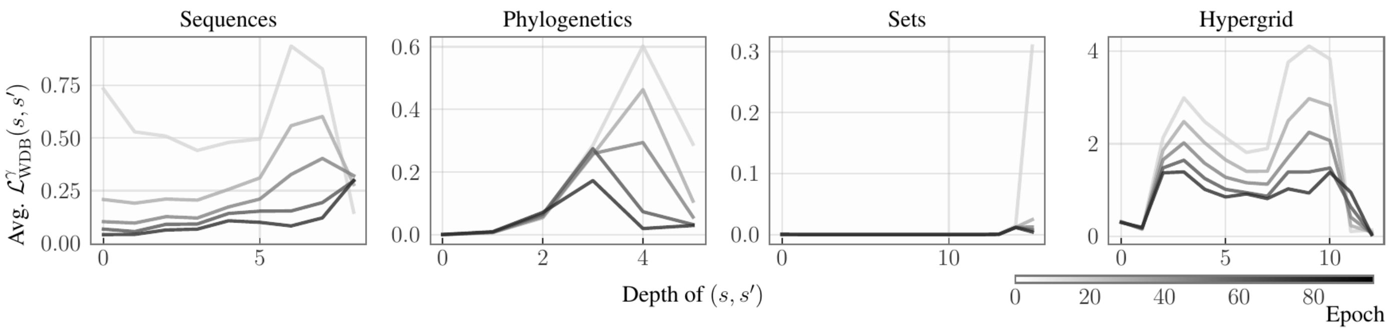
$$\mathcal{L}^{\gamma}_{\text{WDB}}(p_F, p_B, F) = \mathbb{E}_{\tau}\left[\frac{1}{\sum_{(s,s')\in\tau}\gamma(s,s')}\sum_{(s,s')\in\tau}\gamma(s,s')\left(\log\frac{F(s)p_F(s'|s)}{F(s')p_B(s|s')}\right)^2\right],$$

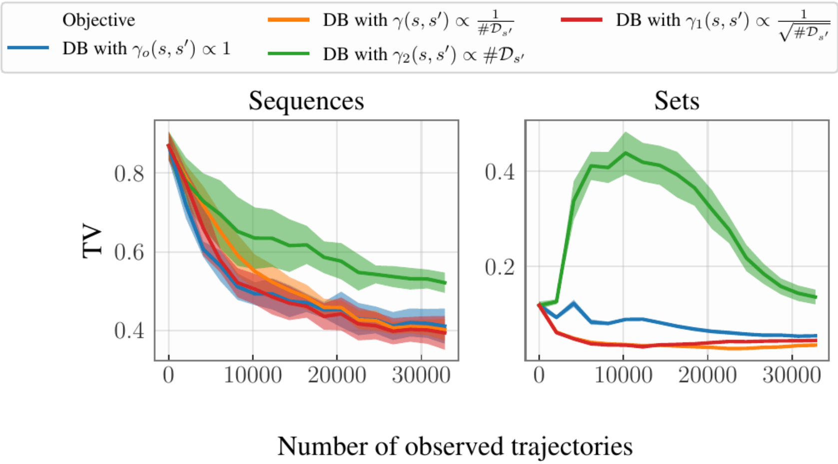
$$\text{e.g., with } \gamma(s,s') = \frac{1}{\#\mathcal{D}_{s'}}$$



Weighing the DB loss

Sanity check: WDB attenuates early-stage transitions





While monotonically decreasing functions empirically work well, finding the best scheme is an open research question

When do GFlowNets learn the right distribution?

The GFlowNet literature is still in its infancy, with many open theoretical questions (with clear practical repercussions), e.g.:

How sensitive are GFlowNets to balance violations?

Are there cases when balance never be achieved?

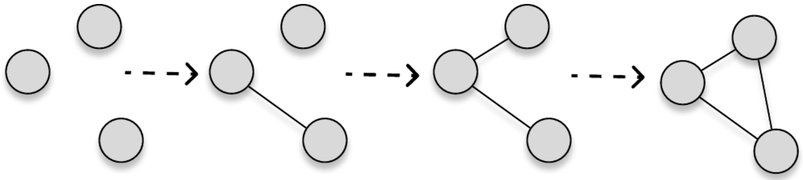
How to correctly diagnose a GFlowNet's sampling quality?

Representational Limits

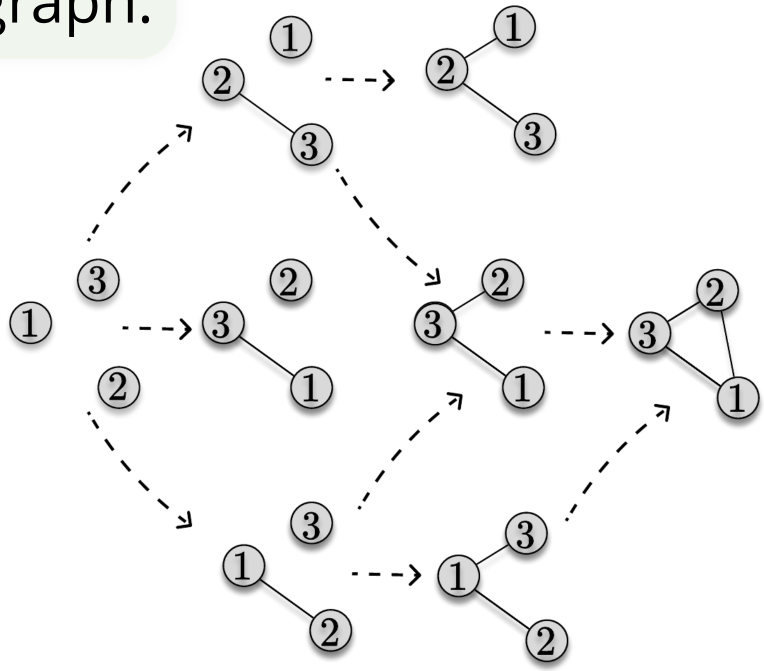
Graphs are the most promising domain for GFlowNets and, for this case graph neural networks (GNNs) are broadly used to parameterize policy networks.

Pro: invariances greatly reduce state graph.

Con: GNNs are not universal approximators. (So what?)

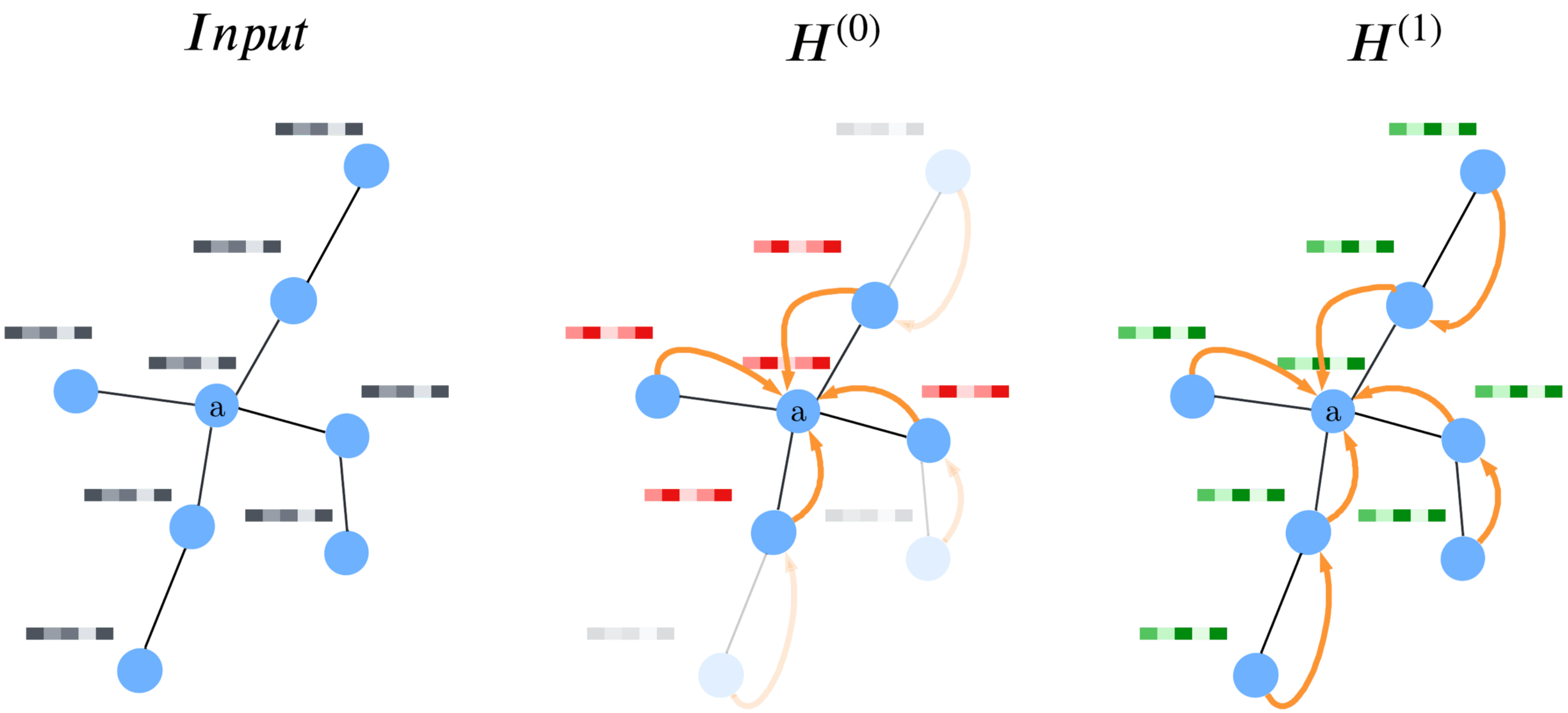


w/ GNN policy



w/ MLP policy

Representational Limits
(Message-passing GNNs, 1-WL test)



Representational Limits
(Message-passing GNNs, 1-WL test)

Message-passing GNNs

For each layer $\ell = 1 \ldots L$, each node v in G updates its embedding according to :

$$\boldsymbol{m}_v^{(\ell)} = \texttt{Aggregate}\left(\{\!\{\boldsymbol{h}_u^{(\ell-1)} : u \in \mathcal{N}_G(v)\}\!\}\right)$$

$$\boldsymbol{h}_v^{(\ell)} = \texttt{Update}\left(\boldsymbol{m}_v^{(\ell)}, \boldsymbol{h}_v^{(\ell-1)}\right)$$

$\{\!\{h_v^{(L)}\}\!\}_{v \in V(G)}$ is the representation of G

1-WL isomorphism test

Given two graphs G, G' run sincronously for all their nodes:

$$\boldsymbol{h}_v^{(\ell)} = \texttt{Hash}\left(\boldsymbol{h}_v^{(\ell)}, \{\!\{\boldsymbol{h}_u^{(\ell-1)} : u \in \mathcal{N}_G(v)\}\!\}\right)$$

$$\boldsymbol{h}_{v'}^{(\ell)} = \texttt{Hash}\left(\boldsymbol{h}_{v'}^{(\ell)}, \{\!\{\boldsymbol{h}_{u'}^{(\ell-1)} : u' \in \mathcal{N}_{G'}(v')\}\!\}\right)$$

If $\{\!\{h_v^{(L)}\}\!\}_{v \in V(G)} \neq \{\!\{h_{v'}^{(L)}\}\!\}_{v' \in V(G')}$ stop.

Message-passing GNNs are at most as expressive as the 1-WL test

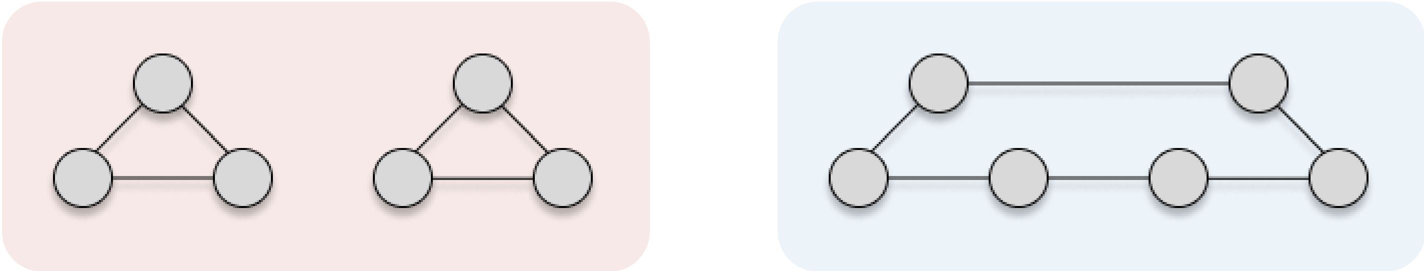
1-WL converges in diameter(G) steps --- colors are only relabeled

If the 1-WL test fails, we say graphs are 1-WL indistinguishable

Failure cases of 1-WL

1-WL naturally does not solve general isomorphism.

Unatributted regular graphs are notorious failure examples.



$$\boldsymbol{h}_v^{(1)} = \texttt{Hash}\left(\text{Gray}, \{\!\{\text{Gray}, \text{Gray}\}\!\}\right) \quad \Rightarrow \quad \boldsymbol{h}_v^{(1)} \leftarrow \text{Blue}$$

$$\boldsymbol{h}_v^{(2)} = \texttt{Hash}\left(\text{Blue}, \{\!\{\text{Blue}, \text{Blue}\}\!\}\right) \quad \Rightarrow \quad \boldsymbol{h}_v^{(2)} \leftarrow \text{Green}$$

$$\boldsymbol{h}_v^{(3)} = \texttt{Hash}\left(\text{Green}, \{\!\{\text{Green}, \text{Green}\}\!\}\right) \quad \Rightarrow \quad \boldsymbol{h}_v^{(3)} \leftarrow \text{Magenta}$$

$$\vdots$$

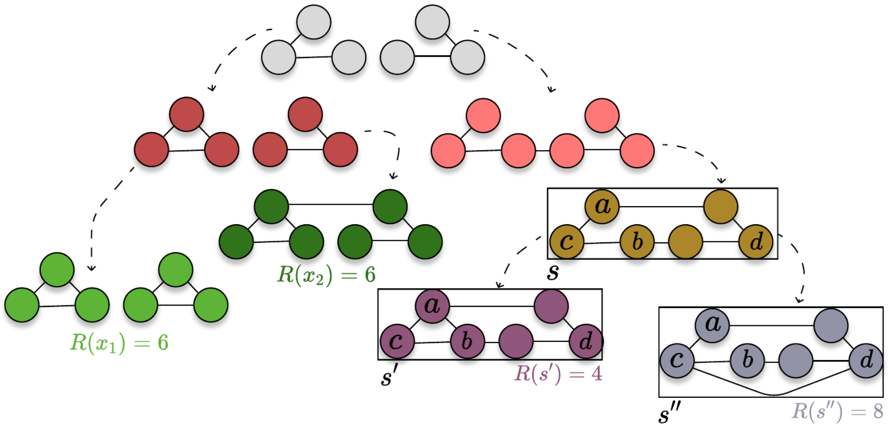
Implications for 1-WL GFlowNets

Graphs are 1-WL indistinguishable if their multiset of embeddings do not differ. Likewise, we say node/edge pairs are indistinguishable if they are mapped to the same color.

$$p_F(s'|s) \propto \exp\left\{\text{MLP}\left(\psi_1\left(\{\phi_{a|s}, \phi_{b|s}\}\right)\right)\right\}$$

$$p_F(s''|s) \propto \exp\left\{\text{MLP}\left(\psi_1\left(\{\phi_{c|s}, \phi_{d|s}\}\right)\right)\right\}$$

Impliying both branches get the same probability mass.



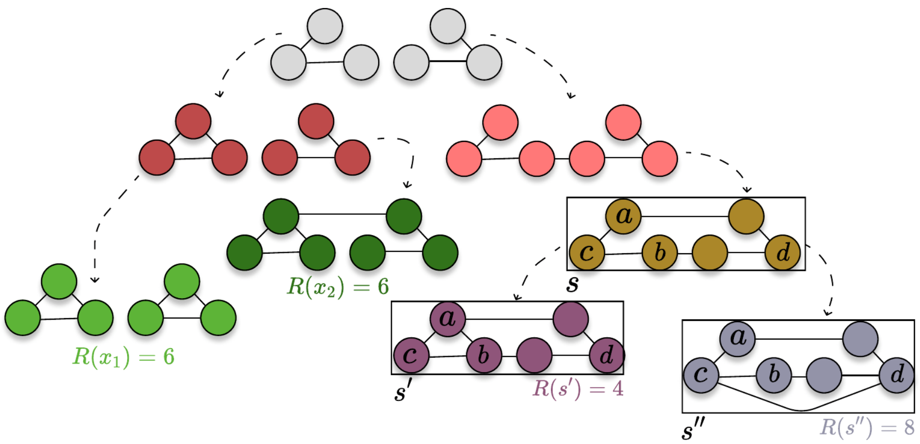
There is a broad class of SG + reward combinations for which 1-WL GFlowNets can never achieve balance.

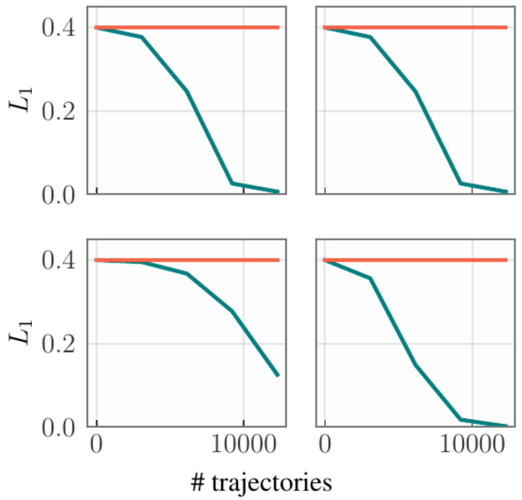
Look-ahead GFlowNets

Insight: 1-WL indistinguishable actions may lead to 1-WL distinguishable graphs (as in our previous example)

Conclusion: By incorporating children as inputs in the forward policy (**LA-GFlowNets**), we can boost the expressiveness of 1-WL GFlowNets.

LA-GFlowNet policy: $p_F(s'|s) \propto \exp\left\{\mathrm{MLP}\left(\psi_1\left(\{\phi_{a|s}, \phi_{b|s}\}\right) \parallel \psi_2\left(\{\phi_{w|s'}\}_{w \in V(s')}\right)\right)\right\}$





When do GFlowNets learn the right distribution?

The GFlowNet literature is still in its infancy, with many open theoretical questions (with clear practical repercussions), e.g.:

How sensitive are GFlowNets to balance violations?

Are there cases when balance never be achieved?

How to correctly diagnose a GFlowNet's sampling quality?

Convergence diagnostics for GFlowNets

So far, we measured empirical performance as TV to the target.

$$p_T(x) := \sum_{\tau:\ s_o \rightsquigarrow x} p_F(\tau) = \mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\frac{p_F(\tau)}{p_B(\tau|x)} \right]$$

For small supports, enumerating all paths to x is easy and need not rely on Monte Carlo estimates.

For large supports, this is intractable in many ways, and results are unnormalized, rendering naive TV unfeasible.

Previous works evaluate average reward of samples, number of visited modes --- metrics which can be easily gamed

Convergence diagnostics for GFlowNets

Our solution: compute average TV over small cuts of the support

Flow Consistency in Subgraphs (FCS)

Let P_S be a positive probability distribution on β-sized subsets of $\mathcal{X}$, $\beta \geq 2$.
For each $S \subseteq \mathcal{X}$, define the restrictions of p_T and R to the set S as

$$p_T^{(S)}(x) = \frac{\mathbf{1}_{\{x \in S\}} p_T(x)}{\sum_{y \in S} p_T(y)} \text{ and } R^{(S)}(x) = \frac{\mathbf{1}_{\{x \in S\}} R(x)}{\sum_{y \in S} R(y)} \text{ for } x \in \mathcal{X}.$$

We define FCS as the expected TV between $p_T^{(S)}$ and $R^{(S)}$:

$$\text{FCS}(p_T, R) = \mathbb{E}_{S \sim P_S}[\text{TV}(p_T^{(S)}, R^{(S)})].$$

If terminal probabilities are correctly computed, the FCS is zero iff TV is zero. In practice, we use MC estimates.

Diagnosing GFlowNets

FCS is provably correct and can detect pathologies that widely adopted (although misleading) metrics cannot, e.g., avg. top-k reward.

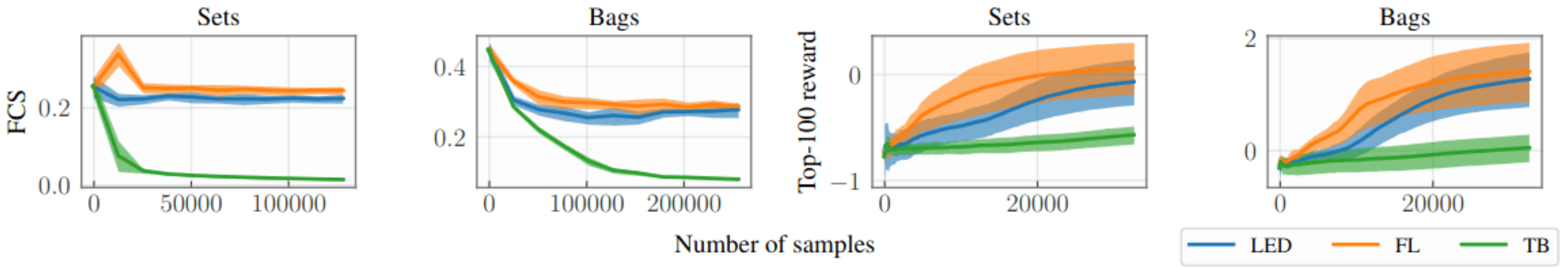


Figure 7: **FCS correctly detects** that neither FL- nor LED-GFlowNets learn to sample from the target (left and mid-left plots), contrasting with the performance of a standard GFlowNet. On the contrary, the average reward of the sampled objects during training incorrectly suggests that FL- and LED-GFlowNets achieve a faster convergence than a standardly trained model (mid-right and right).

We hope FCS will help declutter the literature and promote a fair comparison between different methods.

What's next?

GFlowNets are blooming right now, with many opportunities for applications and methodological work.

Major research questions

1. Under wich circunstances and how well do GFlowNets generalize?
2. How can we optimally assign credit to intermediate states, accelerating convergence?
3. How can we design GFlowNets that work well in mixed discrete/real supports?

Major applications: LLMs, drug discovery, causal discovery, combinatorial optimization, phylogenetics, ...

Questions?

Contact info

diego.mesquita@fgv.br

weakly-informative.github.io

X: @wkly_infrmtive

FGV EMAp