An Introduction to Generative Flow Networks

Juan Ramirez
juan.ramirez@mila.quebec
Outline

- Motivation
- Flow Networks
- Generative Flow Networks
- Training GFlowNets
- GFlowNets in Context: MCMC and Generative Modeling
- Applications
Important References


Setting

We want to do **inference** with an intractable distribution:

- Sampling.
- Computing expectations.

Sampling is intractable, but:

- Objects can be built **compositionally**.
- There is access to a reward function.
Structure Learning

Let $X_1, X_2, \ldots, X_d$ be rvs. We want to construct a “minimal” graph $G$ such that:

$$p(x_1, x_2, \ldots, x_d) \in \mathcal{L}(G)$$
Structure Learning
Flow Networks
Flow Networks

- Dag $G = (S, A)$ with source and sink states $s_0, s_f \in S$
- Trajectories $\tau = (s_1, \ldots, s_n)$, where $s_t \rightarrow s_{t+1} \in A$
- Let $\mathcal{T}$ be the set of all trajectories.
- Forward transition probabilities $\hat{P}_F(s' | s)$

$$\sum_{s' \in \text{Child}(s)} \hat{P}_F(s' | s) = 1 \quad \hat{P}_F(\tau) := \prod_{t=1}^{n-1} \hat{P}_F(s_{t+1} | s_t)$$
Flow Networks

Instead of modeling probabilities on the graph, consider flows:

\[ F : \mathcal{T} \mapsto \mathbb{R}^+ \]

\[ F(s) := F(\{\tau \in \mathcal{T} : s \in \tau\}) = \sum_{\tau \in \mathcal{T} : s \in \tau} F(\tau) \]

\[ F(s \rightarrow s') := F(\{\tau \in \mathcal{T} : s \rightarrow s' \in \tau\}) = \sum_{\tau \in \mathcal{T} : s \rightarrow s' \in \tau} F(\tau) \]
Flow Networks

*Fig 2 in GFlowNet Foundations
Flow Networks

Flows induce a probability distribution over the graph:

\[ F(s_0) = \sum_{\tau \in T} F(\tau) = Z, \]

\[ F(s_f) = \sum_{\tau \in T} F(\tau) = Z. \]

\[ P(s) := \frac{F(s)}{Z} \]

\[ P(s \rightarrow s' \mid s) = \frac{F(s \rightarrow s')}{F(s)} \]
Flow Networks

- The flow does induce a distribution over the **terminating states**.
- It tells us the likelihood of sampling each object.

\[
P_T(s) := P(s \rightarrow s_f) = \frac{F(s \rightarrow s_f)}{Z} \quad \sum_{s \in S^f} P_T(s) = 1.
\]
Flow Networks

If we could learn a flow that matches a reward function on the states:

- A probability distribution over terminating states follows.
- The probabilities are proportional to the flow (and thus, to the reward).
- And the graph $G$ can be used to efficiently sample objects (with that probability).

But parameterizing a flow is expensive! -> One value per trajectory in $G$. 
Markovian Flow Networks

Instead, consider **Markovian** flow networks:

\[
P(s \rightarrow s' | \tau) = P(s \rightarrow s' | s)
\]

- Cheaper to model.
- Induces a unique forward transition probability \(P(s'|s)\).
- Prop. 23 in *GFlowNet Foundations*: the set of markovian flows is expressive enough to represent **all flow functions** over trajectories.
Markovian Flow Networks

For a flow to be valid, it must follow that:

\[
F(s) = \sum_{s' \in \text{Child}(s)} F(s \rightarrow s')
\]

\[
F(s') = \sum_{s \in \text{Par}(s')} F(s \rightarrow s')
\]
Generative Flow Networks - GFlowNets

A **GFlowNet** is a (Markovian) flow network where:

\[
\forall s \in S^f \quad F(s \to s_f) = R(s)
\]

- **R** is a given reward function.
  - If \( s \) is not a valid terminating state, set a reward of 0.
- **F** is parameterized with (say) a NN.

If **F**: (i) is a flow and (ii) satisfies the equation above, then samples are drawn proportional to **R**.
Training GFlowNets
Training GFlowNets

The NeurIPS 2021 GFlowNet paper enforces flow matching:

$$\sum_{s \in \text{Pa}(s')} F_{\theta}(s \rightarrow s') - \sum_{s'' \in \text{Ch}(s')} F_{\theta}(s' \rightarrow s'') = R(s')$$

Which leads to the following objective:

$$\mathcal{L}(\phi) = \mathbb{E}_\pi \left[ \left( \log \frac{\sum_s F_{\phi}(s \rightarrow s')}{R(s')} + \sum_{s''} F_{\phi}(s' \rightarrow s'') \right)^2 \right]$$
Training GFlowNets

The expectation is over all trajectories -> **intractable**.

\[
\mathcal{L}(\phi) = \mathbb{E}_\pi \left[ \log \frac{\sum_s F_\phi(s \rightarrow s')}{R(s')} + \sum_{s''} F_\phi(s' \rightarrow s'') \right]^2
\]

In practice, trajectories are sampled.

- If there is structure, the GFlowNet could generalize across trajectories.
- Trade-off between sampling likely trajectories and exploration.
Alternative Loss Functions

Detailed balance (Bengio et al., 2021):

\[ F_\theta(s)P_F(s' \mid s) = F_\theta(s')P_B(s \mid s') \]

Trajectory balance (Malkin et al., 2022):

\[ Z \prod_{t=1}^{n} P_F(s_t \mid s_{t-1}) = R(s_n) \prod_{i=1}^{n} P_B(s_{t-1} \mid s_t) \]

*Trajectory Balance: Improved Credit Assignment in GFlowNets. Nikolay Malkin et al., NeurIPS 2022.*
GFlowNets in Context
Markov Chain Monte Carlo

**MCMC**
- No “setup” cost.
- Samples are not independent
- Sampling is costly
  - Mixing time can scale poorly
  - Mode mixing

**GFlowNets**
- Needs to be trained.
- Samples are independent.
- Sampling is efficient: do ancestral sampling
Generative Modeling

Generative Modeling

- Trained on data, to maximize its likelihood.
- Prone to overfitting.
- Improves with more data.

GFlowNets

- Trained to match a reward function.
- Prone to underfitting.
- Improves with more trajectories.
Reinforcement Learning

See section 7.2 in *GFlowNet Foundations*
Applications
Combinatorial Optimization Problems

For instance, finding the **largest clique** in a graph.

- **States**: sets of fully connected nodes.
- **Reward**: size of the set.

Scientific Discovery

For instance, constructing molecules.

- **States**: sets of partially constructed molecules.
- **Reward**: a property of the molecule.

GFlowNet EM
Structure Learning
Extensions

- Continuous GFlowNets
- Conditional GFlowNets
References

Generative Flow Networks and Bayesian Structure Learning
Structure Learning

Credit to Tristan Deleu
Structure Learning

- Graphical representation of the **conditional independences** in a distribution, represented as a **Directed Acyclic Graph (DAG)**.

- The **joint distribution** is decomposed as:

\[
P(X_1, \ldots, X_d) = \prod_{k=1}^{d} P(X_k \mid Pa_G(X_k))
\]

- **Structure learning**: Given a dataset of observations \(D\), find the graph structure \(G\).
Markov Equivalence

- Recall: A Directed Graphical Model encodes the Conditional Independence of a distribution.
- Multiple DAGs may encode the same Conditional Independence statements.

![Diagram of Markov Equivalence]

- Two DAGs encoding the same Conditional Independence statements are called **Markov Equivalent**.
Theorem (Verma & Pearl, 1991)

Two DAGs $G_1$ and $G_2$ are **Markov Equivalent** if and only if they have the same skeleton and the same v-structures.

- Markov Equivalence Classes can be represented as a **Completed Partially Directed Acyclic Graph** (CPDAG).
A & B are d-separated by C in $\mathcal{G}$

$X_A \perp X_B \mid X_C$

Exercise: Violation of Faithfulness

$X := N_X$
$Y := X + N_Y$
$Z := X - Y + N_Z$

with $N_X, N_Y, N_Z \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

$\rho(X, Y, Z)$ is a Multivariate Normal distribution, where the only conditional independence statements are: $X \perp Z$ and $X \not\perp Z \mid Y$. 
Structure Identifiability

**Theorem**

If \( p \) is faithful wrt. \( \mathcal{G}^0 \), then the Markov Equivalence class of \( \mathcal{G}^0 \) is **identifiable** from \( p \).

- Only the Markov Equivalence class is identifiable from observations, **not an individual graph**. Two Markov Equivalent graphs may lead to different causal conclusions!

  ![Diagram](image)

- Under different assumptions, an individual DAG may be identifiable
  - Additive Noise Model (ANM): \( X_j := f_j(X_{Pa_j}) + N_j, N_j \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \), where \( f_j \) are nonlinear.
  - Using **interventional data** (i.e. data resulting from controlled experiments).
Constraint-based methods

**Step 1: Identify the skeleton**

For each pair of nodes $X$ & $Y$, and $A \subseteq V \setminus \{X, Y\}$, test if $X \perp_D Y \mid A$.

If there is no set $A$ s.t. $X \perp_D Y \mid A$, then add an edge $X \rightarrow Y$. 

![Graph Diagram]

IC Algorithm
Constraint-based methods

**Step 1: Identify the skeleton**
For each pair of nodes $X$ & $Y$, and $\mathbf{A} \subseteq V \setminus \{X, Y\}$, test if $X \perp_{D} Y \mid \mathbf{A}$.

If there is no set $\mathbf{A}$ s.t. $X \perp_{D} Y \mid \mathbf{A}$, then add an edge $X \leftarrow Y$.

**Step 2: Identify the v-structures**
For each structure $X \leftarrow Z \rightarrow Y$ with no edge between $X$ & $Y$, orient $X \rightarrow Z \leftarrow Y$ iff $Z \notin \mathbf{A}$, where $\mathbf{A}$ is such that $X \perp_{D} Y \mid \mathbf{A}$.
Constraint-based methods

Step 2’: Additional orientations

Use Meek’s orientation rules to orient some of the remaining edges.

IC Algorithm
Score-based methods

• Treat the problem of learning the structure of the DAG as a **model selection problem**

\[
\max_{\mathcal{G} \in \text{DAG}} \text{score}(\mathcal{G} \mid \mathcal{D})
\]

**Choice of scores**

• **Likelihood score:**
  \[
  \text{score}_{L}(\mathcal{G} \mid \mathcal{D}) = \log p(\mathcal{D} \mid \hat{\theta}_{g}^{\text{MLE}}, \mathcal{G})
  \]

• **Bayesian score:**
  \[
  \text{score}_{B}(\mathcal{G} \mid \mathcal{D}) = \log p(\mathcal{D} \mid \mathcal{G}) + \log p(\mathcal{G})
  \]

• **Bayesian Information Criterion (BIC):**
  \[
  \text{score}_{\text{BIC}}(\mathcal{G} \mid \mathcal{D}) = \log p(\mathcal{D} \mid \hat{\theta}_{g}^{\text{MLE}}, \mathcal{G}) - \frac{\log N}{2} \text{Dim}[\mathcal{G}]
  \]
Score-based methods

\[
\max_{\mathcal{G} \in \text{DAG}} \text{score}(\mathcal{G} \mid \mathcal{D})
\]

• How to search over the space of DAGs?

• The number of DAGs over \(n\) nodes is super-exponential in \(n\): \(2^{\Theta(n^2)}\)

**Theorem**

Let \(G_{\leq d} = \{\mathcal{G} \text{ a } \text{DAG} \mid \text{every node has at most } d \text{ parents}\}\). Finding a DAG in \(G_{\leq d}\) that maximizes a score is \(\text{NP-hard}\) for \(d \geq 2\).

• Heuristic solutions:
  
  • Greedy algorithms: Hill climbing, GES
  
  • Genetic algorithms
  
  • Constrained continuous optimization: NOTEARS, Gran-DAG, DCDI, etc…
Bayesian Structure Learning

- When the **dataset is small**, we want to take into account the **epistemic uncertainty** over the graph structures of the Bayesian Network.

- **Markov Equivalence:** There may be multiple graphs encoding the **same conditional independences**.

  \[
  X \indep Z \quad \text{and} \quad X \indep Z \mid Y
  \]

  \[
  X \perp Z \quad \text{and} \quad X \perp Z \mid Y
  \]

- From the point of view of observations, Markov equivalent graphs **fit the data equally well**.

\[
P(G \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid G)P(G)}{P(\mathcal{D})}
\]
Bayesian Structure Learning

Instead of finding a single graph from observations, characterize the whole posterior distribution over graphs:

$$P(G \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid G)P(G)}{P(\mathcal{D})}$$

Graphs are **discrete** and **composite** objects. The number of DAGs is **super-exponential** in the number of nodes (e.g., there are $10^{72}$ DAGs over 15 nodes).

The marginal likelihood is in general **intractable**:

$$P(\mathcal{D} \mid G) = \int_{\Theta} P(\mathcal{D} \mid \theta, G)P(\theta \mid G)d\theta$$

We will choose models so that this can be computed efficiently in **closed form**.
Markov Chain Monte Carlo

- Approximate the posterior distribution using Markov Chain Monte Carlo (MCMC).
- Build a Markov chain by adding, removing, or reversing edges uniformly at random.
- Issue: Highly multimodal distribution (Markov equivalence), leading to poor mixing of the Markov chain.

DAG-GFlowNet
GFlowNet over DAGs

- DAGs are constructed sequentially **one edge at a time**, starting from the **empty graph**.
- All the states of the GFlowNet are **valid DAGs**, meaning that **all the states are terminating**.
- A new edge to be added to a DAG:
  - must not **already be present**;
  - must not **introduce a cycle**.
- We can filter out invalid actions using a **mask**, that can also be **updated online**.
Detailed balance condition

**Flow matching condition** (Bengio et al., 2021)

\[
\sum_{s \in Pa(s')} F_\theta(s \rightarrow s') - \sum_{s'' \in Ch(s')} F_\theta(s' \rightarrow s'') = R(s')
\]

**Detailed balance condition** (Ours)

\[
R(s')P_B(s \mid s')P_\theta(s_f \mid s) = R(s)P_\theta(s' \mid s)P_\theta(s_f \mid s')
\]

- Valid when **all the states** of the GFlowNet are **terminating**
- Induces a **distribution** \( P(s) \propto R(s) \)
- It **does not depend on flows** anymore (flow-matching or detailed balance conditions).
- It **does not depend on the total flow** \( Z \) (trajectory balance condition).

Fixed **backward** transition probability

*e.g. Uniform distribution*

Learned **forward** transition probability
Hierarchical model for the forward transition probabilities: \( P_\theta(G' \mid G) = (1 - P_\theta(s_f \mid G)) P_\theta(G' \mid G, \neg s_f) \)
Application to Bayesian Structure Learning

**Bayesian Structure Learning**
Characterize the **posterior distribution** over DAGs

\[
P(G \mid D) = \frac{P(D \mid G)P(G)}{P(D)}
\]

**GFlowNet**
A GFlowNet induces a **distribution**

\[
P(s) \propto R(s)
\]

**DAG-GFlowNet**

\[
R(G) = P(D \mid G)P(G)
\]
The GFlowNet is trained **off-policy**.

We use a **replay buffer** to store transitions over the course of training, and **sample transitions** from the replay buffer.

\[
\mathcal{L}(\theta) = \mathbb{E}_\pi \left[ \log \frac{R(G') P_B(G \mid G') P_\theta(s_f \mid G)}{R(G) P_\theta(G' \mid G) P_\theta(s_f \mid G')} \right]^2
\]
Experimental results
Experimental results – Accurate approximation

Comparison with the exact posterior distribution $P(G \mid \mathcal{D})$ on graphs with $d = 5$ nodes, computed by enumerating the 29,281 possible DAGs.
Experimental results – Simulated data

E-SHD

Lower is better

MC\textsuperscript{3}  Gadget  B-PC  B-GES  DiBS  BCD  GFN

AUROC

Higher is better

MC\textsuperscript{3}  Gadget  B-PC  B-GES  DiBS  BCD  GFN

log $P(G, D' \mid D)$

Higher is better

MC\textsuperscript{3}  Gadget  B-PC  B-GES  DiBS  BCD  GFN

MCMC

Bootstrapping

Variational Inference
Experimental results – Flow cytometry data

- Real-world **flow cytometry data**, to learn protein signaling pathways.

- Data: continuous measurements of **11 phosphoproteins**. There are **853 observations**.

- The *ground truth* graph contains **11 nodes** and **17 edges**.

- The consensus graph may not represent a realistic description of the system.
Experimental results – Flow cytometry data

- **BGe score**
- **log-marginal probability of MEC**

**DiBS** samples many low-scoring graphs

**DiBS** samples many unique MECs

**Low diversity** of sampled graphs within a MEC

**BCD Nets** samples only 2 distinct graphs
Experimental results – Interventional data

- The real world flow cytometry data also contains interventional data, based on experimentations where some phosphoproteins are inhibited.

- We model these as perfect interventions, even though it may not be the case in practice.

- We know the intervention targets.

- We can adapt the reward function (computation of the marginal likelihood) to handle a mixture of observational & interventional data.

- This is a first step toward causal discovery.

<table>
<thead>
<tr>
<th></th>
<th>E-# Edges</th>
<th>E-SHD</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact posterior*</td>
<td>–</td>
<td>–</td>
<td>0.816</td>
</tr>
<tr>
<td>MC$^3$</td>
<td>25.97 ± 0.01</td>
<td>25.08 ± 0.02</td>
<td>0.665</td>
</tr>
<tr>
<td>DAG-GFlowNet</td>
<td>30.66 ± 0.04</td>
<td>27.77 ± 0.03</td>
<td>0.700</td>
</tr>
</tbody>
</table>
Thank you

github.com/tristandeleu/jax-dag-gflownet