Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, Yoshua Bengio

McGill Mila Université de Montréal

Summary Slide

Hypothesis: good idea to generate x with probability proportional to >0 rewards

$$\pi(x) \approx \frac{R(x)}{Z} = \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}$$

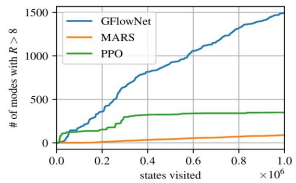
no MCMC-like iterative process; x built sequentially (e.g. a graph, node by node)

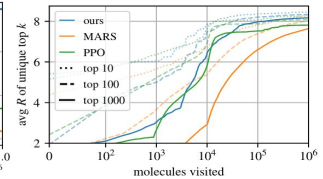
Treat MDP like a flow network (water going through pipes), to get π, learn *F* that satisfies:

$$\sum_{s,a:T(s,a)=s'} F(s,a) = R(s') + \sum_{a' \in \mathcal{A}(s')} F(s',a')$$

in-flow of s’ out-flow of s’

Use: $\pi(a|s) = \frac{F(s,a)}{F(s)} = \frac{F(s,a)}{\sum_{a'} F(s,a')}$





Creates a diverse generative model!

Mila

Motivation: Small-molecule Drug Discovery

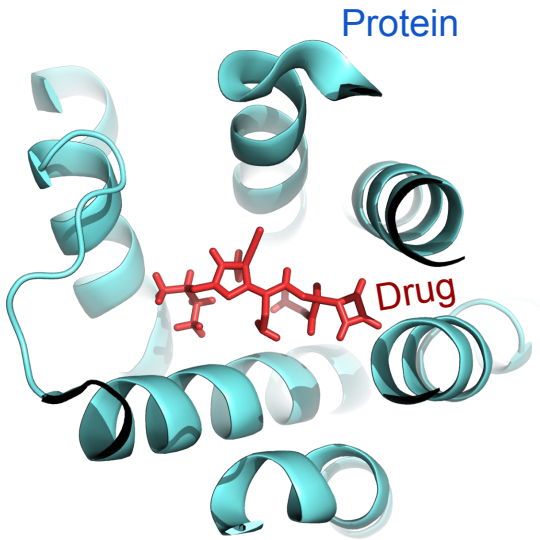
find drugs that bind to protein(s)

$>10^{16\sim20}$ space (simplified + for *one* protein)

most molecules are *bad*:

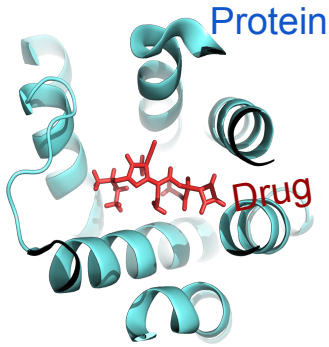
- not chemically feasible
- not binders
- toxic

Needles in a haystack!

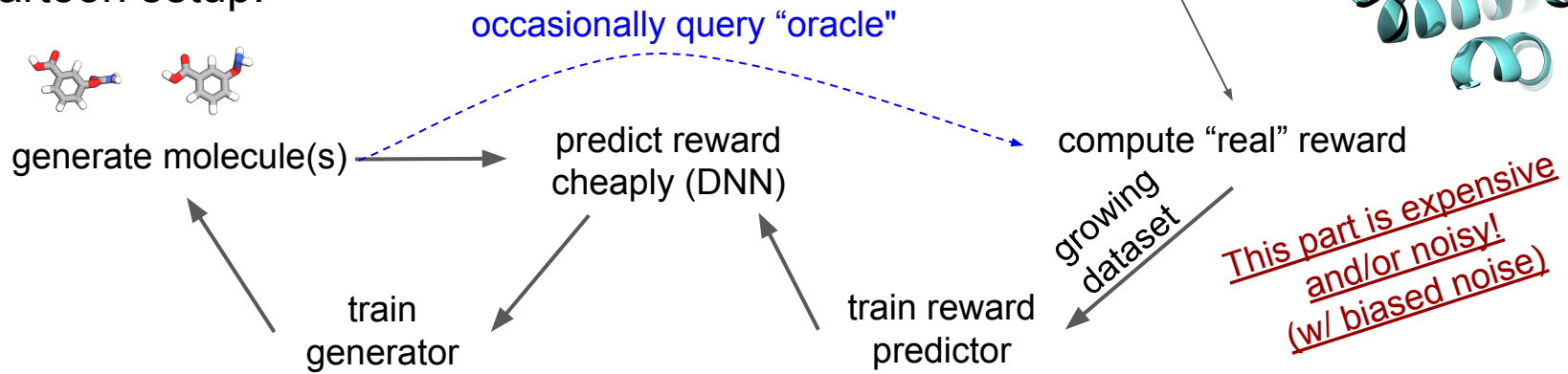


Mila

Motivation: Drug Discovery



Cartoon setup:



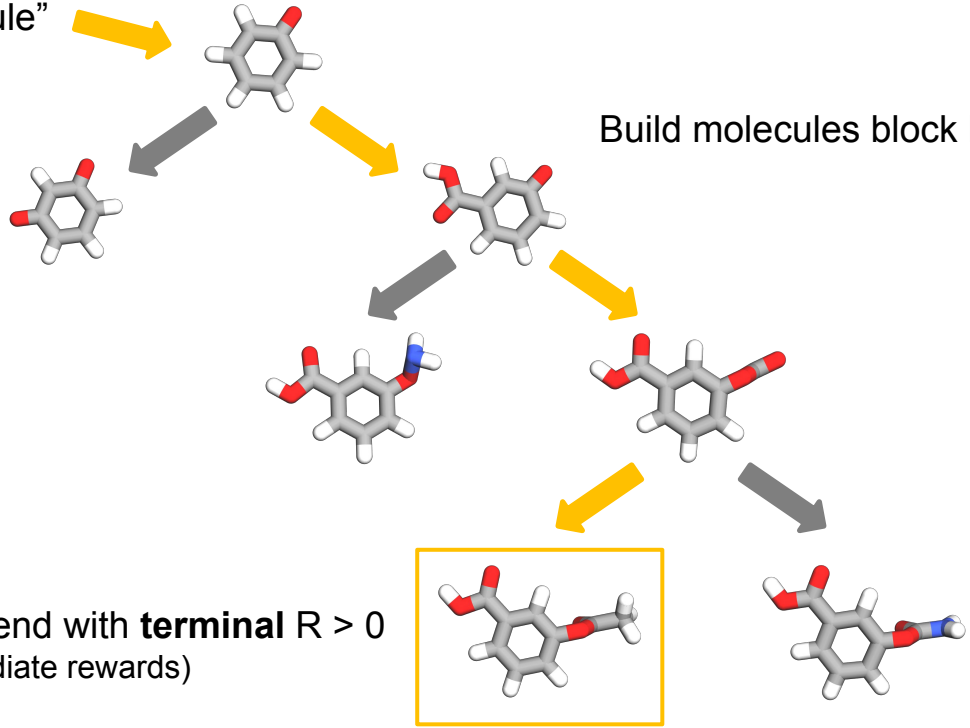
Oracle?

- Ideal: send **diverse** batches (10-100k) of candidates to a lab, O(weeks)
- For now: use noisy physics simulator, O(15 CPUs)/molecule

Mila

Motivation: Drug Discovery

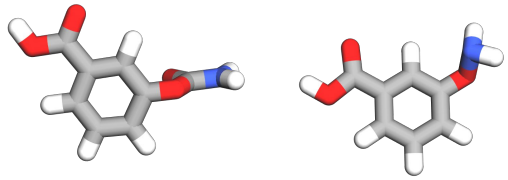
“empty molecule”



Build molecules block by block

Episodes end with **terminal** R > 0
(no intermediate rewards)

Motivation: Drug Discovery



Use noisy physics simulator, O(15 CPUs)/molecule

- Noise = lots of candidates “look good”
- Not quite a needle anymore, many modes!
- Best according to simulator → send to wet lab (one day), may fail there
- What to send?

Diverse modes! Generate all the high reward molecules that look good

Mila

Just apply Reinforcement Learning?

- We have an environment (actions = build molecule)
- We have a reward
- RL! (Segler et al., 2017; De Cao & Kipf, 2018; Popova et al., 2019; Gottipati et al., 2020; Angermueller et al., 2020)

RL is fast. But RL is greedily looking for one mode, even MaxEnt.
Finding **a few** good/optimal modes makes PPO/SAC/SoftQL/&co happy.

→ Not great for *diverse* batch oracle queries

Just apply MCMC?

- We have a Markov Chain (actions = edit molecule)
- We have a reward = unnormalized probability, want to sample from it
- MCMC! (Seff et al., 2019; Xie et al., 2020)

But MCMC is slow, gets stuck in modes easily (lack of diversity!), requires mode-mixing for any new sample

→ Not great for *diverse* batch oracle queries

Mila

What about the usual generative models?

- Trained from positive samples only (e.g. existing drugs)
  But we have a more informative (non-binary) signal! (reward)
- We don’t just want high reward, we want to avoid low reward (and have the data)
- Still possible to do great: Jin et al., 2018; Shi et al., 2020; Luo et al., 2021

Mila

Flow Networks as Generative Models

Hypothesis: a good objective is to generate x with probability proportional to R(x)

We want

$$\pi(x) \approx \frac{R(x)}{Z} = \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}$$

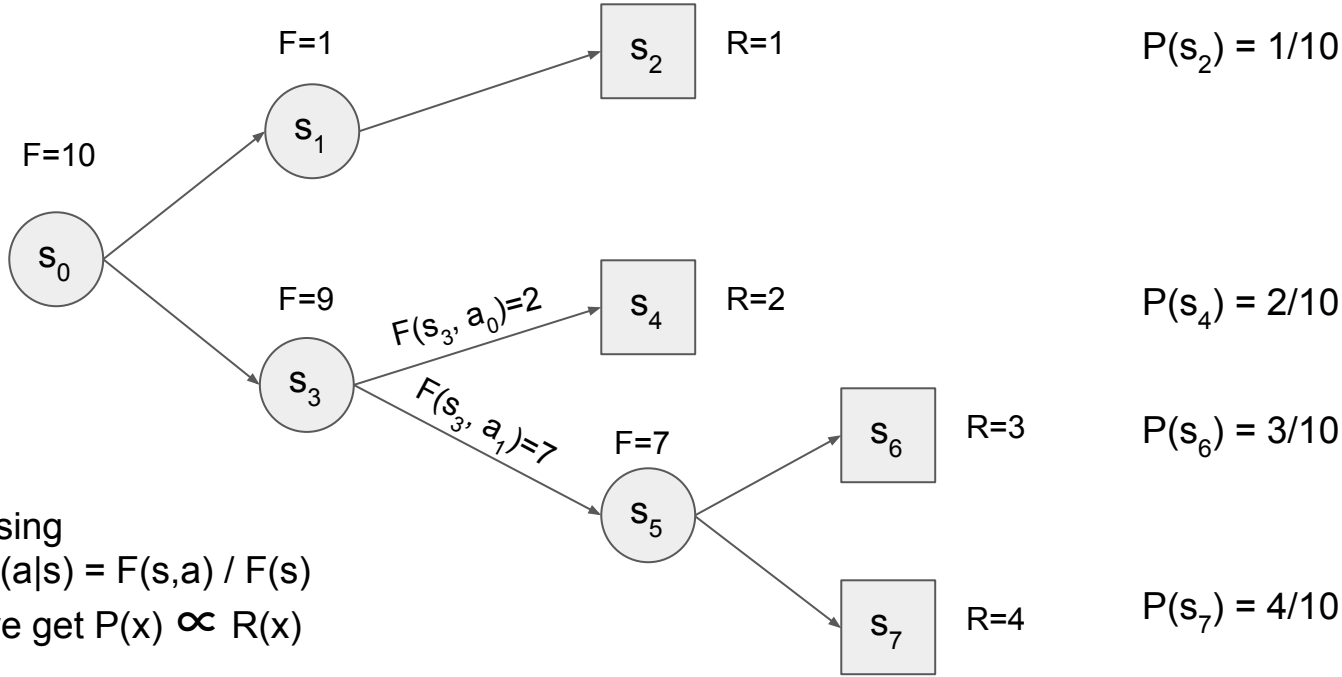
without an MCMC-like iterative process.

Insight from SumTrees led to this, let's start there.

Mila

SumTrees (& the control as inference view: SoftAC/SoftQL)

$\pi(a|s) = Q(s,a) / V(s) = F(s,a) / F(s)$

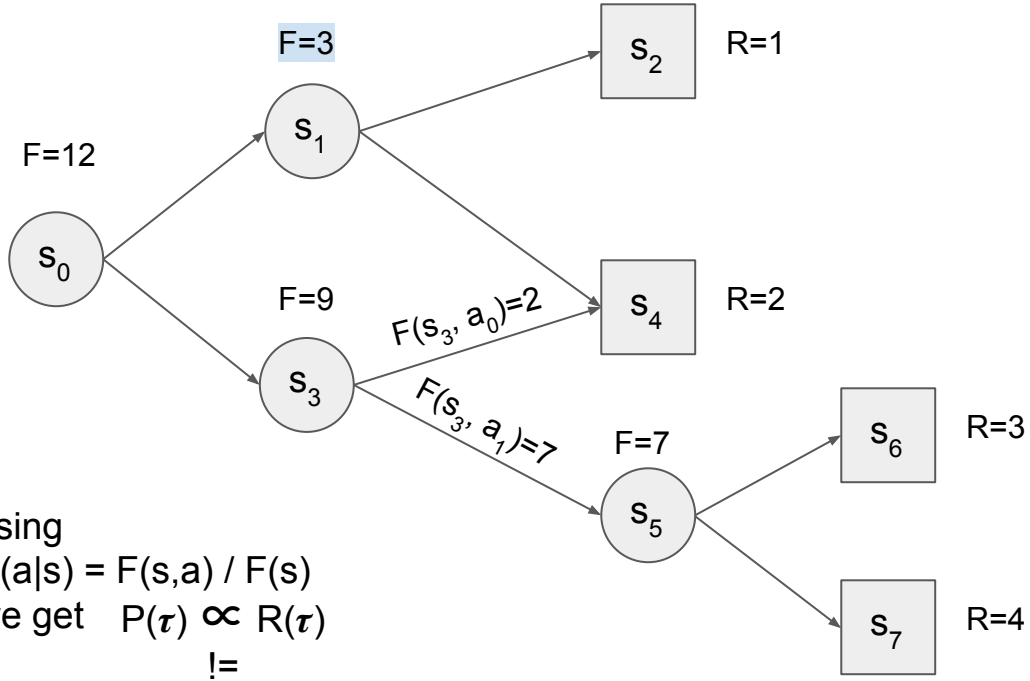


using
$\pi(a|s) = F(s,a) / F(s)$
we get $P(x) \propto R(x)$

Mila

What if it's a DAG?

Naively applying SoftQL/SumTree yields the wrong solution



$P(s_2) = 1/12$

$P(s_4) = 4/12$

$P(s_6) = 3/12$

$P(s_7) = 4/12$

using
$\pi(a|s) = F(s,a) / F(s)$
we get $P(\tau) \propto R(\tau)$

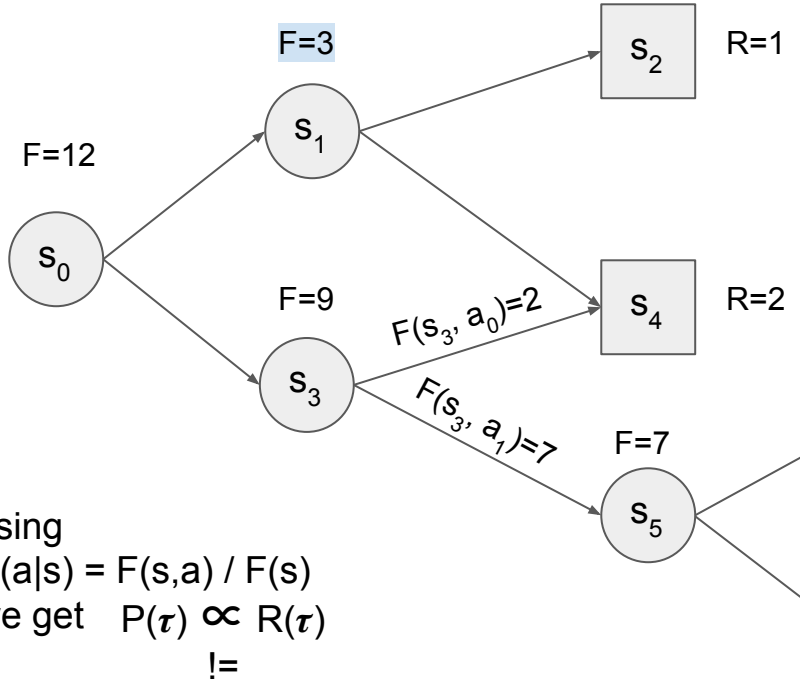
!=

$P(x) \propto R(x)$

Mila

What if it's a DAG?

Naively applying SoftQL/SumTree yields the wrong solution



- $P(\tau) \propto R(\tau)$ is bad if many τ lead to the same X!
- Exponentially bad in graph generation (combinatorial # of paths)

$P(s_4) = 4/12$

$P(s_6) = 3/12$

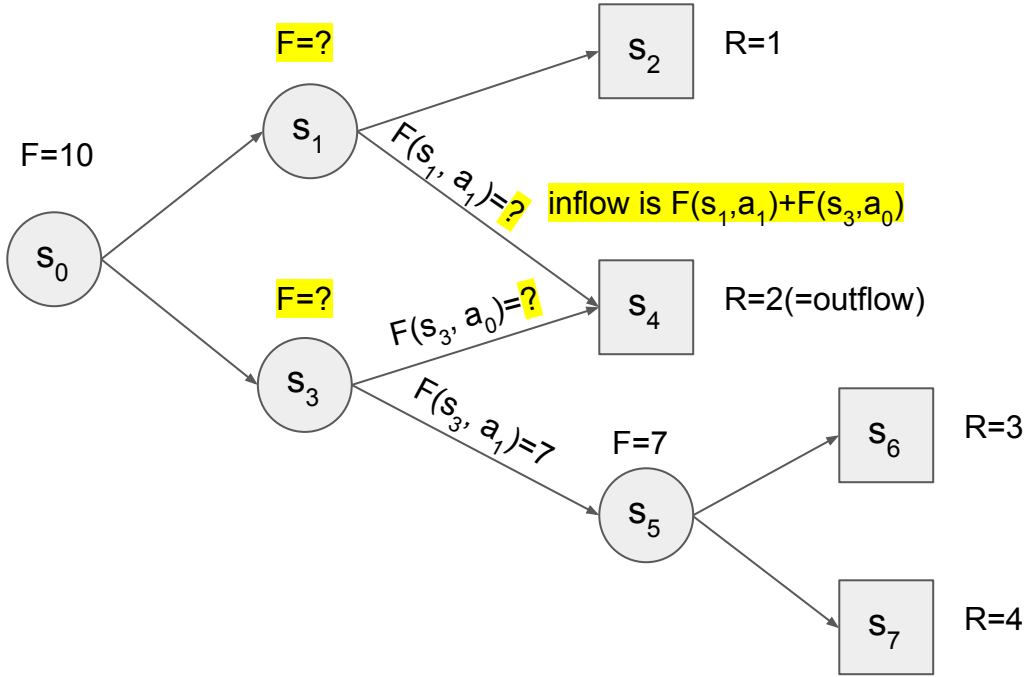
$P(s_7) = 4/12$

using
$\pi(a|s) = F(s,a) / F(s)$
we get $P(\tau) \propto R(\tau)$
!=
$P(x) \propto R(x)$

Mila

Interpreting the DAG as a flow network

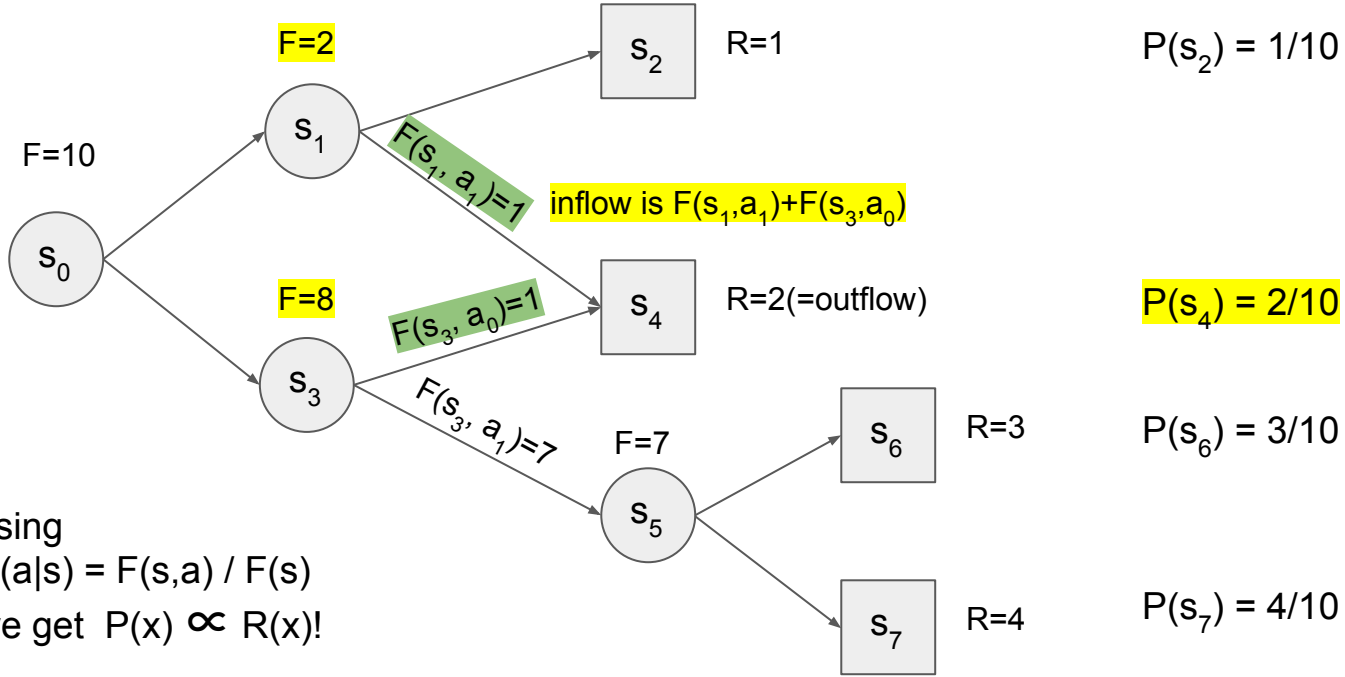
F(s) such that inflow = outflow



Mila

Interpreting the DAG as a flow network

F(s) such that inflow = outflow

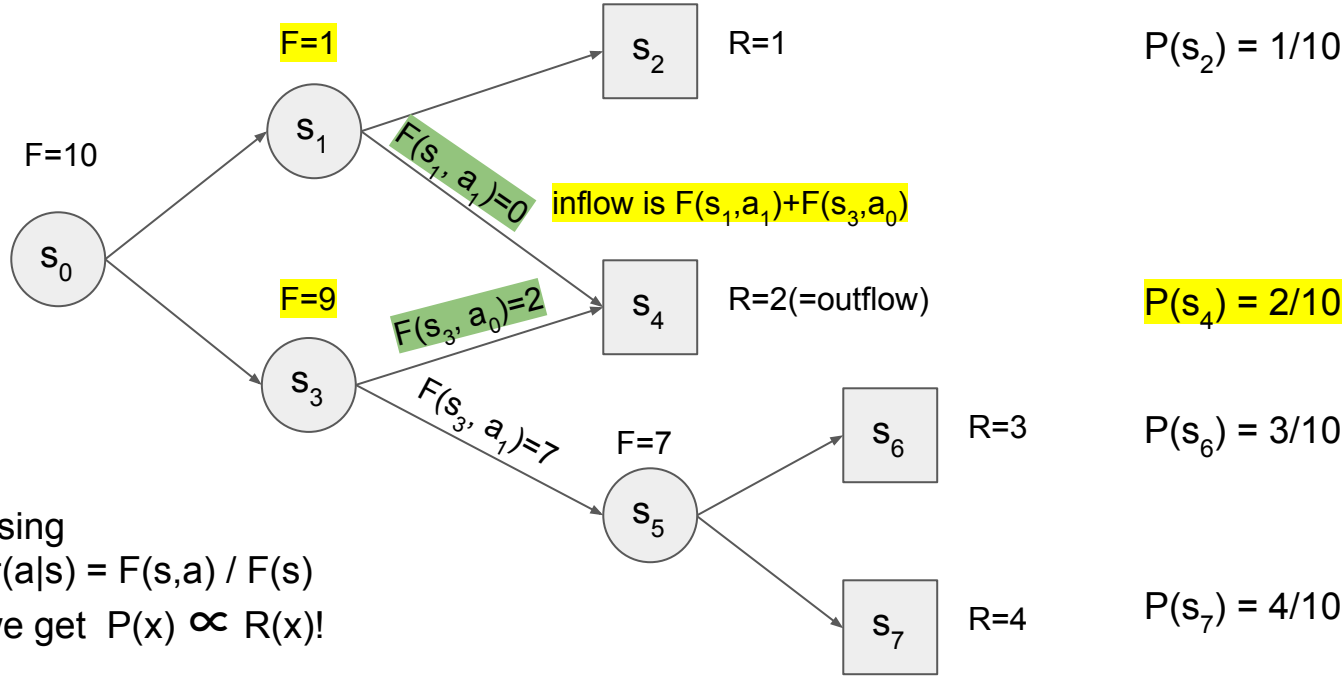


using
$\pi(a|s) = F(s,a) / F(s)$
we get $P(x) \propto R(x)$!

Mila

Interpreting the DAG as a flow network

F(s) such that inflow = outflow



using
$\pi(a|s) = F(s,a) / F(s)$
we get $P(x) \propto R(x)$!

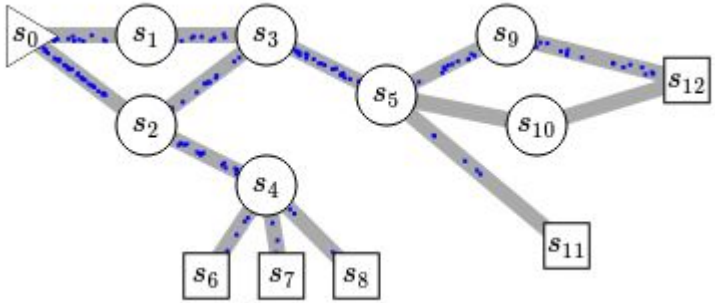
but multiple valid solutions? 🤔

Mila

Interpreting the DAG as a flow network

One more visualization:

flow = #particles moving through pipes



We want a valid flow, given the rewards (“#particles") of the terminal states

Mila

Flow consistency

Satisfy flow conditions, for all s’

$$\sum_{s,a:T(s,a)=s'} F(s,a) = R(s') + \sum_{a' \in \mathcal{A}(s')} F(s',a')$$

in flow of s’ out flow of s’

Your RL senses should be tingling ;)

Mila

Satisfying the flow equations yields what we want

Proposition 2. *Let us define a policy π that generates trajectories starting in state s_0 by sampling actions $a \in \mathcal{A}(s)$ according to*

$$\pi(a|s) = \frac{F(s,a)}{F(s)} \tag{5}$$

where $F(s,a) > 0$ is the flow through allowed edge (s,a), $F(s) = R(s) + \sum_{a \in \mathcal{A}(s)} F(s,a)$ where $R(s) = 0$ for non-terminal nodes s and $F(x) = R(x) > 0$ for terminal nodes x, and the flow equation $\sum_{s,a:T(s,a)=s'} F(s,a) = R(s') + \sum_{a' \in \mathcal{A}(s')} F(s',a')$ is satisfied. Let $\pi(s)$ denote the probability of visiting state s when starting at s_0 and following $\pi(\cdot|\cdot)$. Then

(a) $\pi(s) = \frac{F(s)}{F(s_0)}$

(b) $F(s_0) = \sum_{x \in \mathcal{X}} R(x)$

(c) $\pi(x) = \frac{R(x)}{\sum_{x' \in \mathcal{X}} R(x')}$.

This is what we're after!

Mila

How to train GFlowNet

Take inspiration from Bellman→TD(0) to learn *F*

$$\tilde{\mathcal{L}}_\theta(\tau) = \sum_{s' \in \tau \neq s_0} \left(\sum_{s,a:T(s,a)=s'} F_\theta(s,a) - R(s') - \sum_{a' \in \mathcal{A}(s')} F_\theta(s',a') \right)^2$$

Dangerous objective, $F(s_0,.)$ is going to be huge! $F(s_0) = Z$

Mila

How to train GFlowNet

Instead, learn the log, **and** match flows in log-space

$$\mathcal{L}_{\theta,\epsilon}(\tau) = \sum_{s' \in \tau \neq s_0} \left(\log \left[\epsilon + \sum_{s,a:T(s,a)=s'} \exp F_\theta^{\log}(s,a) \right] - \log \left[\epsilon + R(s') + \sum_{a' \in \mathcal{A}(s')} \exp F_\theta^{\log}(s',a') \right] \right)^2$$

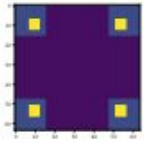
with an epsilon (care less about tiny flows)

Mila

Works well! Hypergrid results

We ran experiments with this reward function:

$$R(x) = R_0 + R_1 \prod_i \mathbb{I}(0.5 < |x_i|) + R_2 \prod_i \mathbb{I}(0.6 < |x_i| < 0.8)$$



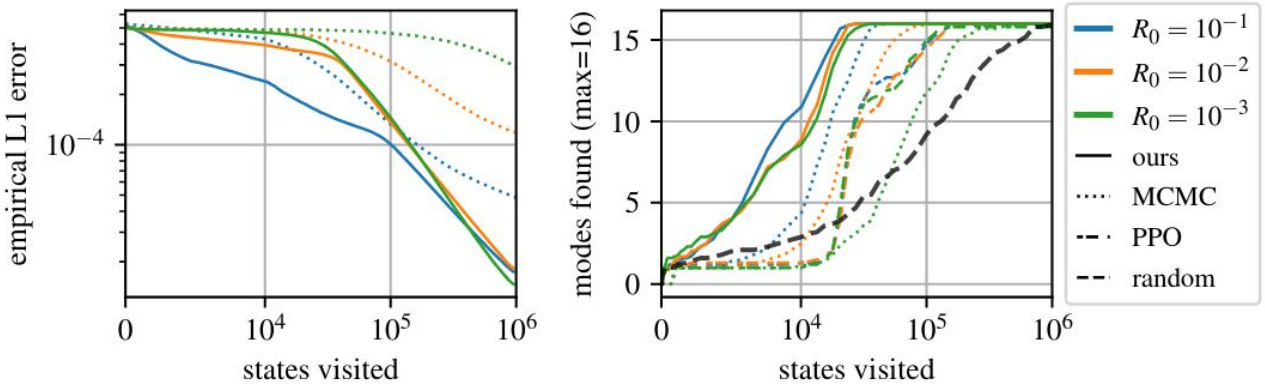
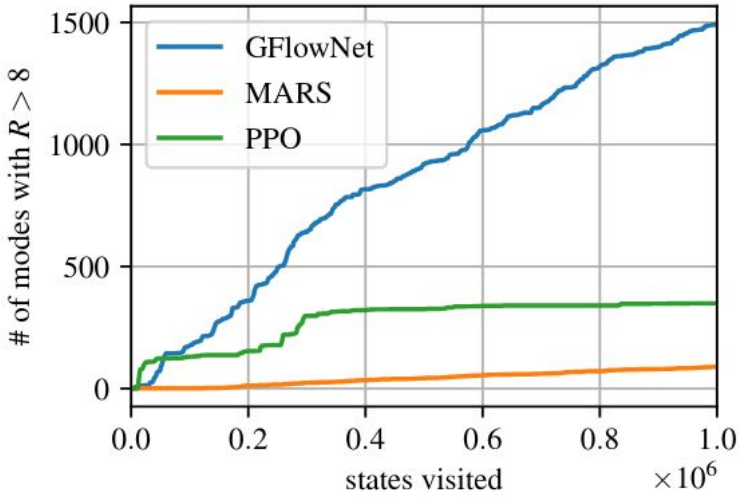


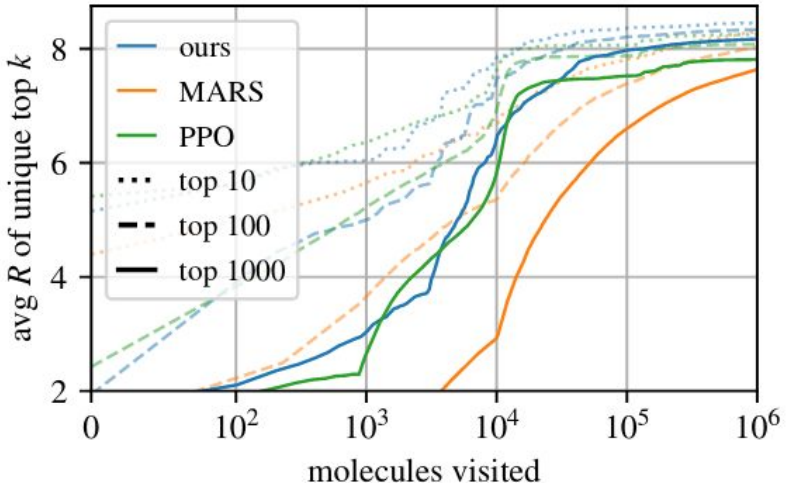
Figure 2: Hypergrid domain. Changing the task difficulty R_0 to illustrate the advantage of GFlowNet over others. We see that as R_0 gets smaller, MCMC struggles to fit the distribution because it struggles to visit all the modes. PPO also struggles to find all the modes, and requires very large entropy regularization, but is robust to the choice of R_0. We plot means over 10 runs for each setting.

Mila

Works well! Molecule results

Pretrain reward function once on 300k molecules (computed on CPU simulator)
Modes are found faster, with better rewards

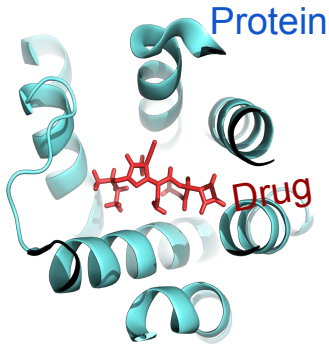




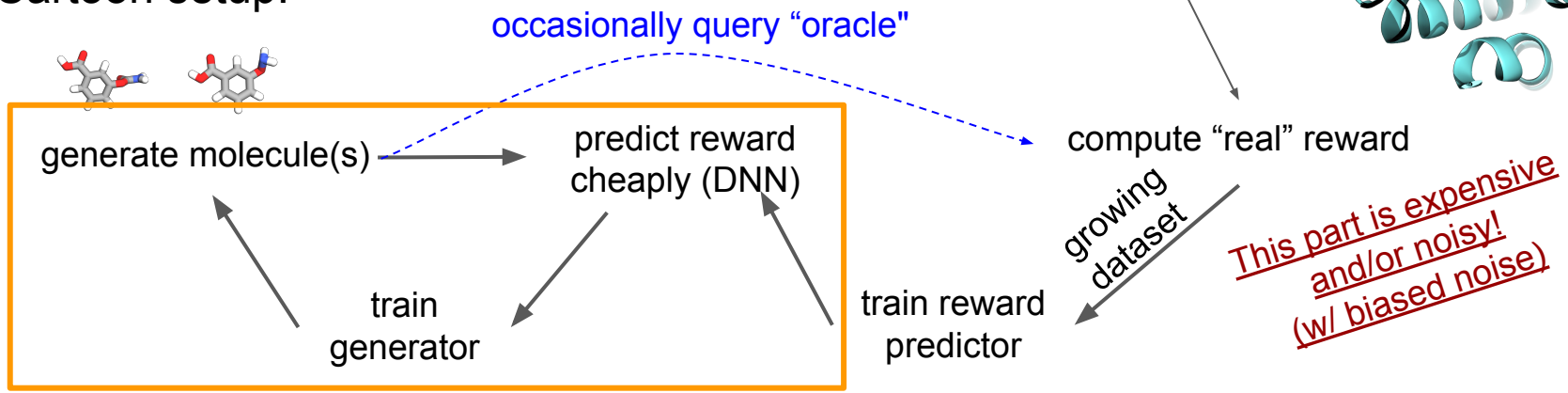
modes = Bemis-Murcko scaffolds

Mila

Motivation: Drug Discovery



Cartoon setup:



Oracle?

- Ideal: send **diverse** batches (10-100k) of candidates to a lab, O(weeks)
- For now: use noisy physics simulator, O(15 CPUs)/molecule

Mila

Works for Batch Active Learning

Use docking as oracle, send only batches:

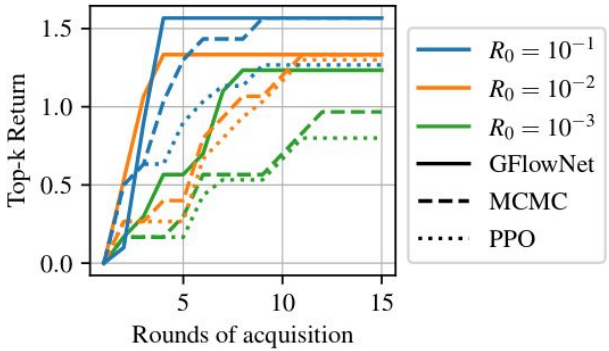


Figure 6: The top-k return (mean over 3 runs) in the 4-D Hyper-grid task with active learning. GFlowNet gets the highest return faster.

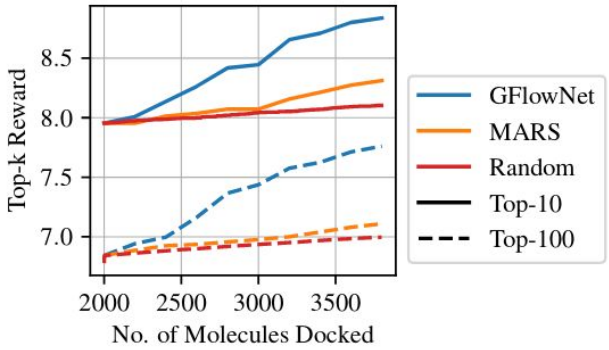


Figure 7: The top-k docking reward (mean over 3 runs) in the molecule task with active learning. GFlowNet consistently generates better samples.

Mila

Other Cool GFlowNet Facts

- This is an off-policy method!
- Converges with enough capacity
- **Constructing a diverse batch is O(n)**
  - Amoritizes the sampling that MCMC does into the training process
  - Has an OOD potential, jumps to novel modes (hard for a MC to do)
  - We get Z (= marginalization) for free
- **Easy to fold epistemic uncertainty in R** (useful for active learning)
- No need for likelihood computations (local “TD” updates)
- Takes advantage of scalar reward (!= pos/negative data in traditional generative models)
- There exists (at least one) equivalences between F and some “real” Q^{π}
  - In fact, with Q^{π} where π is the uniform policy on a canonical tree

Mila

Recap: GFlowNet

- Generative model for discrete objects (e.g. graphs) with $P(x) \propto R(x)$
- Based on a Bellman-like learning objective
- Explores by virtue of $P(x) \propto R(x)$ + not bound by Markov Chain
- Not happy with a single good mode

Cons:

- Some design of R(x) necessary (e.g. $R(x)^{\beta}$)
- Reliance on “TD” could be an issue (Bengio et al., Agarwal et al., 2020)
- Tends to underfit (overestimates small Rs, underestimates large Rs)

Mila

Why this is promising

This is a general setup for solving black box optimization!
→ And provide black-box **exploration**

- Discovering materials
- Discovering antibiotics
- Discovering good controls for a plant, or hyperparameters in ML
- Reasoning tasks: discovering good explanations for data
- Causal discovery: discovering good causal models
- Estimates Z → free energies, empowerment?