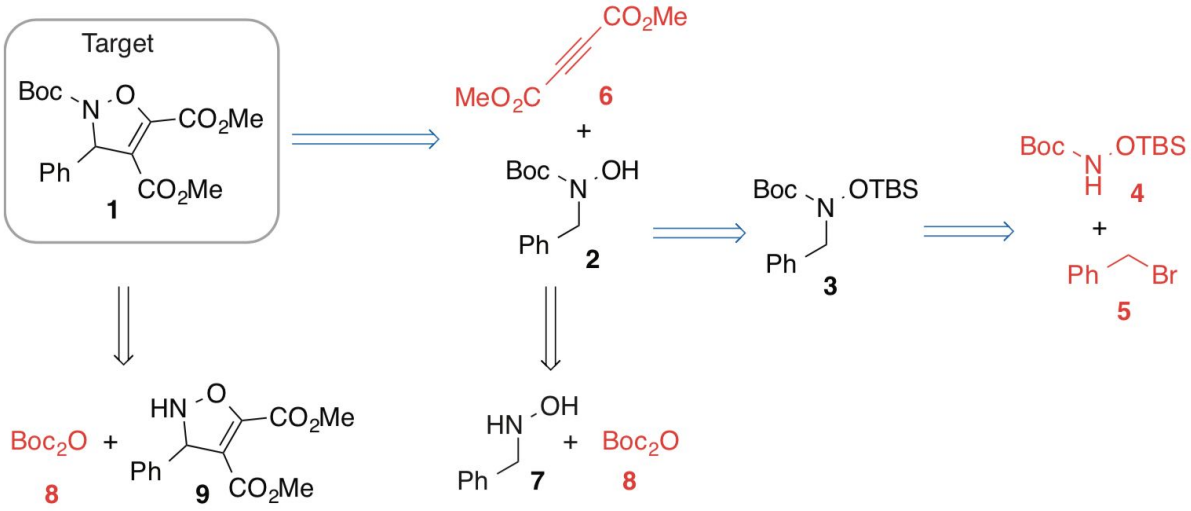
Learning To Plan Chemical Syntheses

Yunhao (Jack) Ji
Yizhan (Ethan) Jiang
Shuja (Shuja) Khalid
Lipai (Jim) Xu

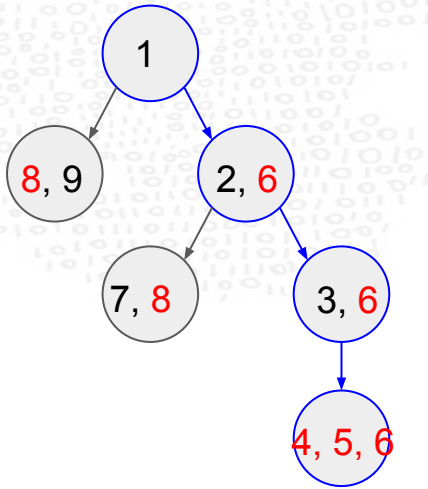
Introduction

- Retrosynthesis

a) Chemical Representation of the Synthesis Plan

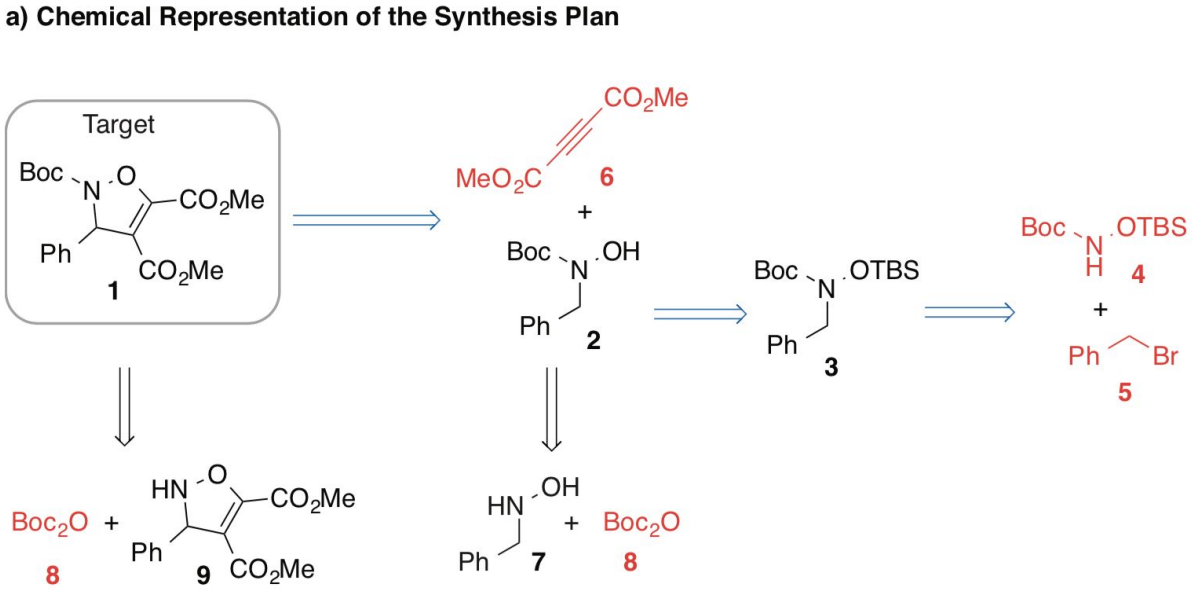


A Search Tree Representation



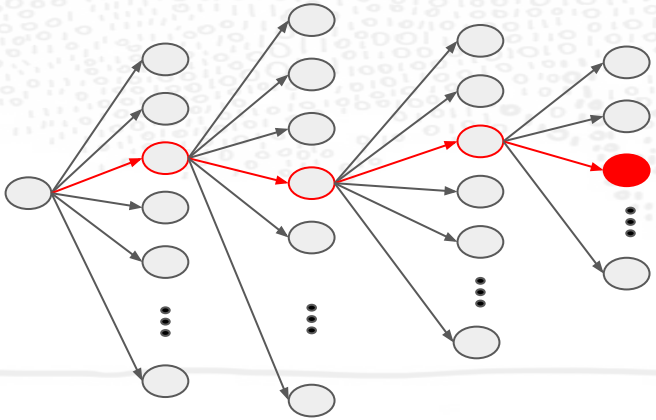
Motivation and Related Work

- Manual constructing a valid tree can be hard



Motivation and Related Work

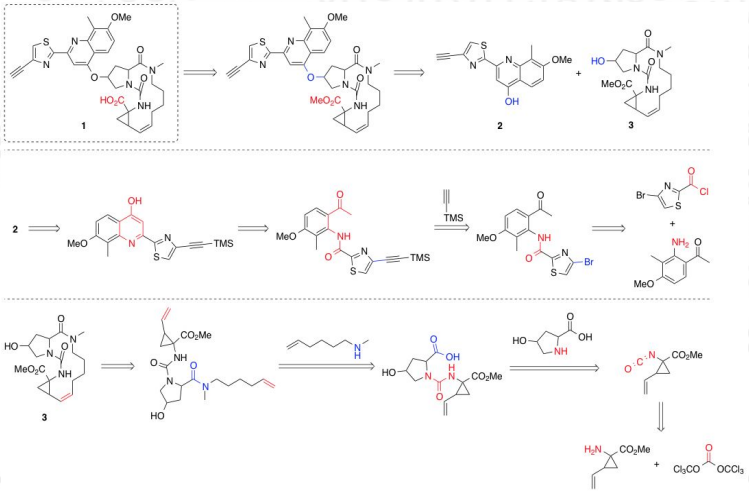
- Computer-assisted synthesis planning (CASP) can automatically extract the transformations
- The generated tree has short depth but large branching factors and hard to define heuristics.



An illustration of an example search tree to a synthesis planning

Neural Networks

Learn Chemical Reaction Rules

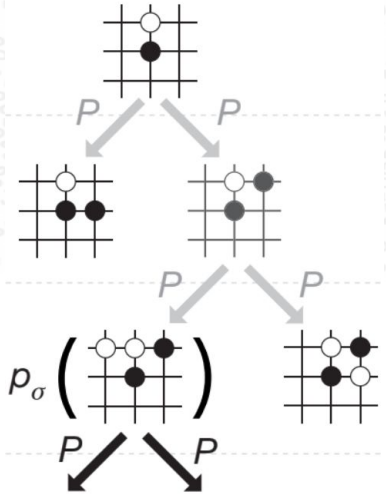


12.4 million reactions from Reaxys database as dataset

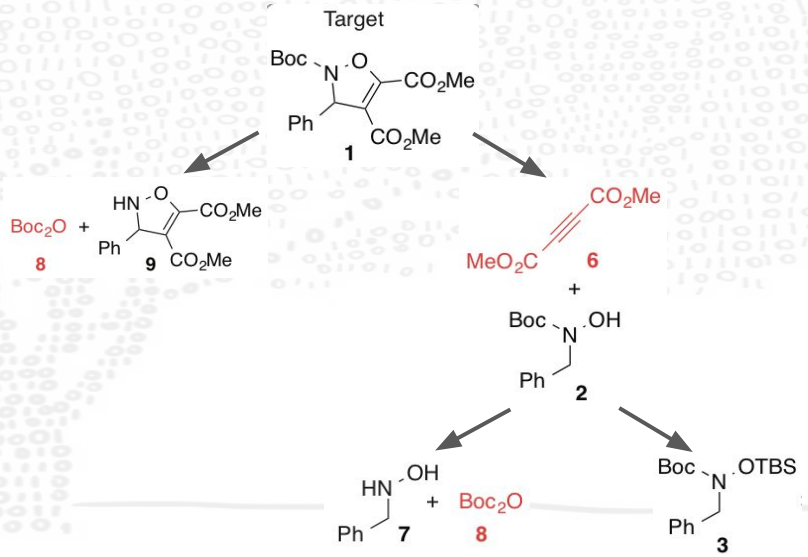
Reaxys®

Neural Networks for Action Selection

Actions in AlphaGo

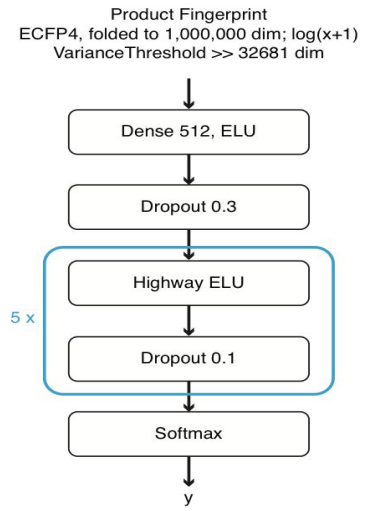


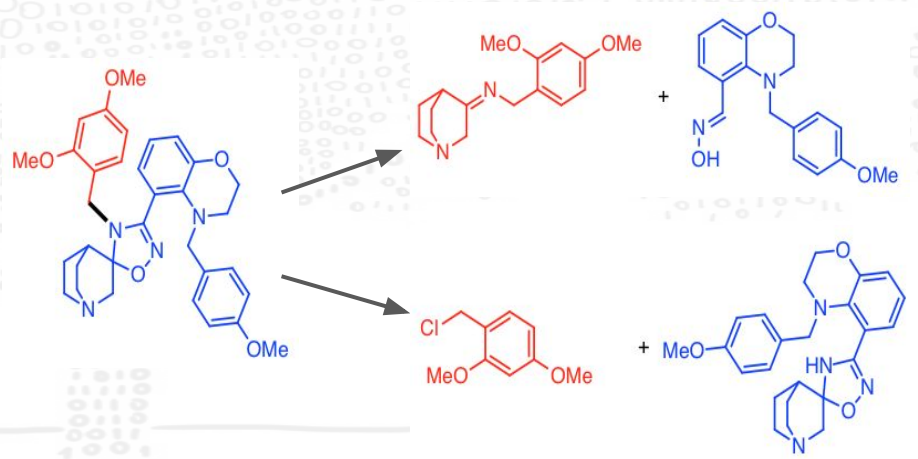
Actions in Chemical Synthesis



Neural Networks for Action Selection (1/2)

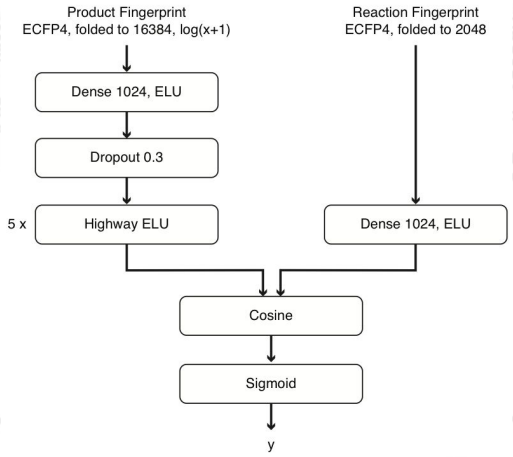
- Expansion Policy Neural Network
  - Find K most possible molecular transformations

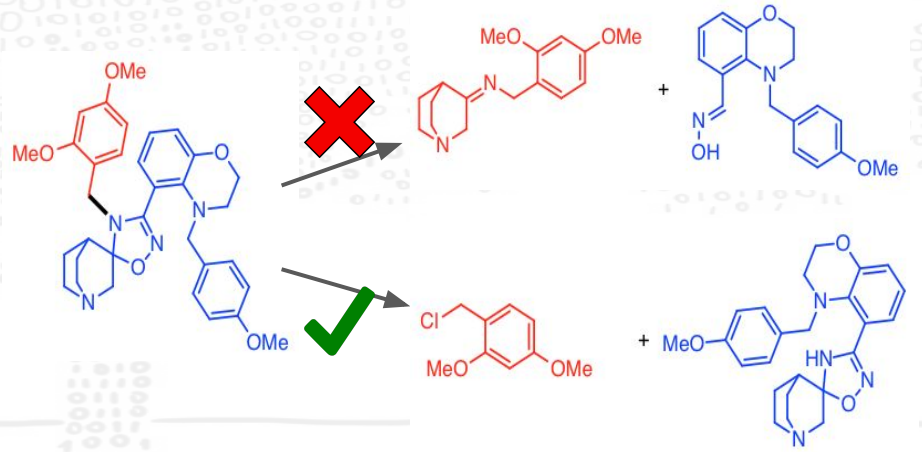




Neural Networks for Action Selection (2/2)

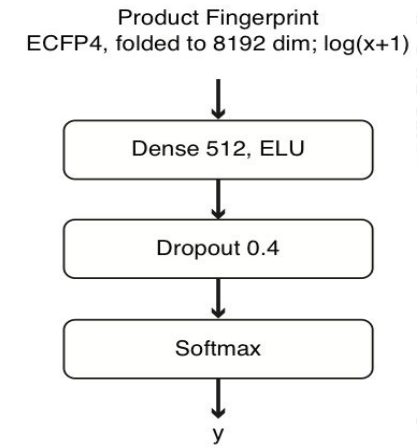
- In-Scope Filter Neural Network
  - Filter out infeasible transformations

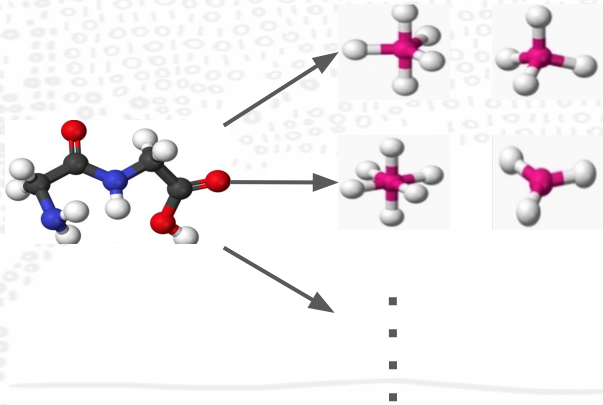




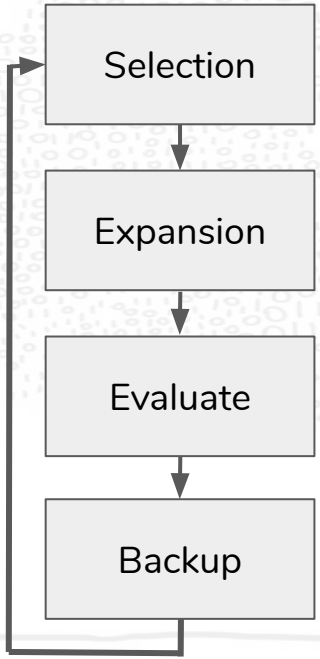
Neural Network for Rollout

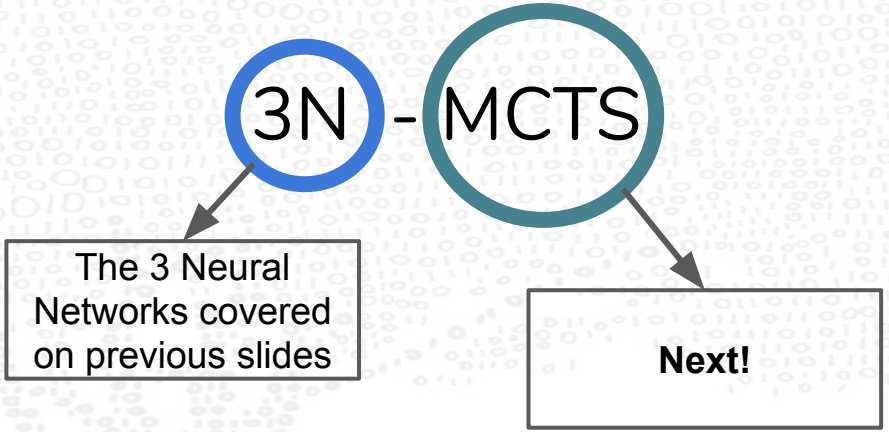
- Rollout Policy Neural Network
  - Select 10 most possible transformations
  - Only three layers for creating fast rollout policy



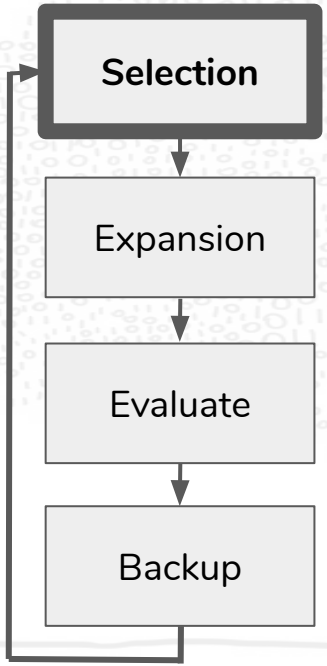


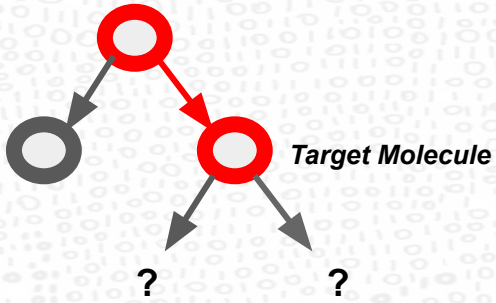
Synthesis Planning with 3N-MCTS





Synthesis Planning with 3N-MCTS

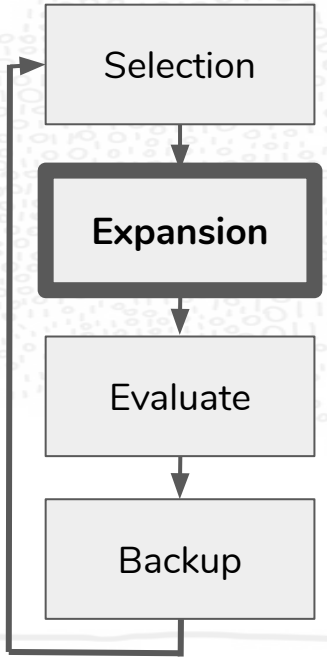


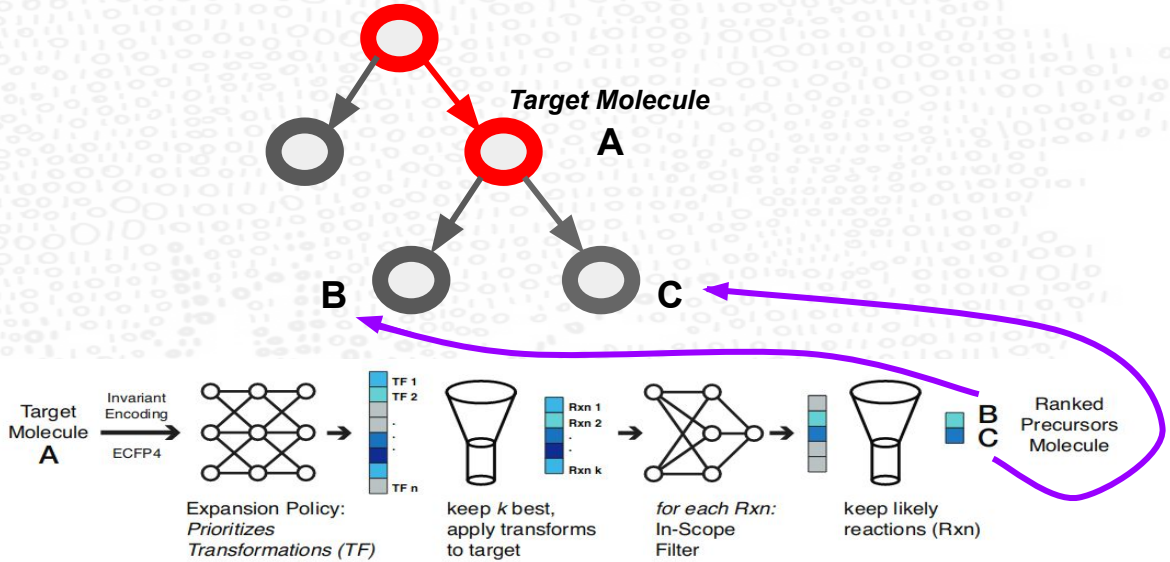


$$a_t = \underset{a \in \mathcal{A}(s_t)}{\arg\max} \left(\frac{Q(s_t, a)}{N(s_t, a)} + cP(s_t, a) \frac{\sqrt{N(s_{t-1}, a_{t-1})}}{1 + N(s_t, a)} \right)$$

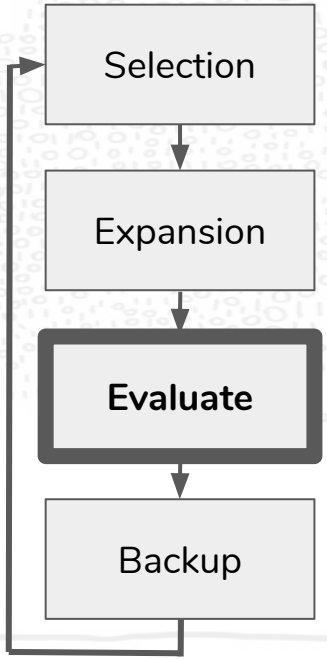
$N(s, a)$: *Visit count of state-action pair*
$P(s, a)$: *Prior probability of visiting state-action pair*
$Q(s, a)$: *Scalar value of state-action pair*
c : *Exploration constant*

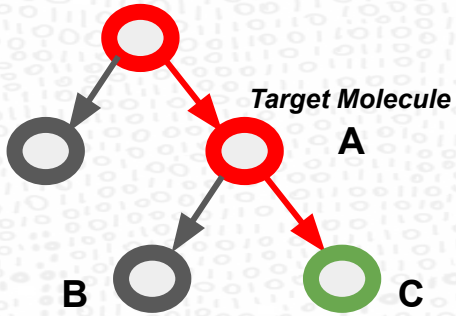
Synthesis Planning with 3N-MCTS





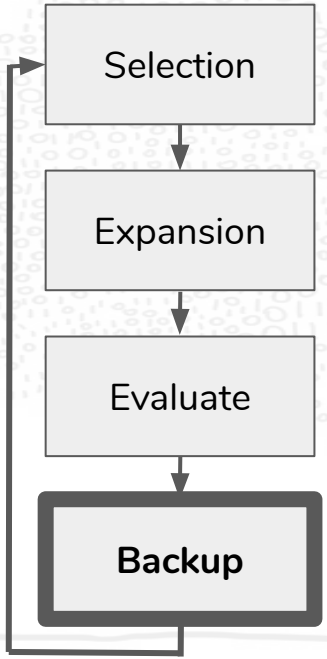
Synthesis Planning with 3N-MCTS

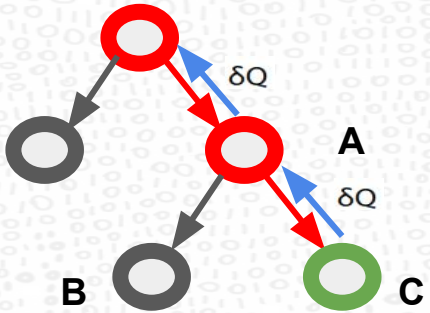




- Check if state is terminal
  - **Terminal** → evaluate with the reward function
  - **Non-terminal** → begin rollout/evaluation step
- Recursively sample actions from rollout policy until termination condition is met

Synthesis Planning with 3N-MCTS





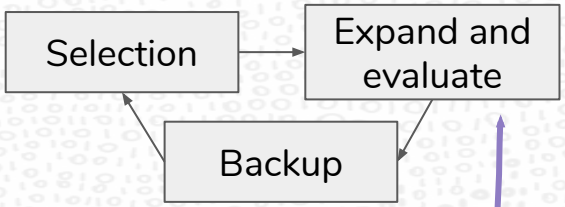
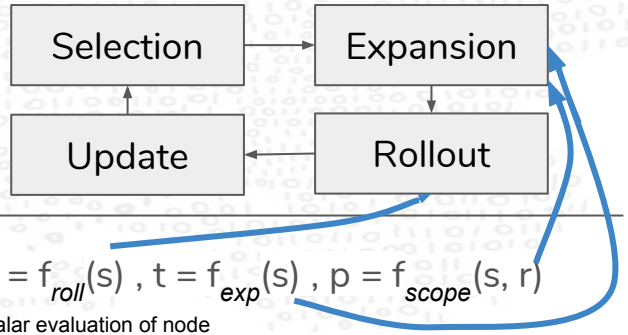
$$Q(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{n} \mathbb{I}_i(s,a) z_i W(b_i)$$

$N(s,a)$: Visit count of state-action pair

$W(b_i)$: Custom objective function

z_i : Reward ∈ [-1,0,1]

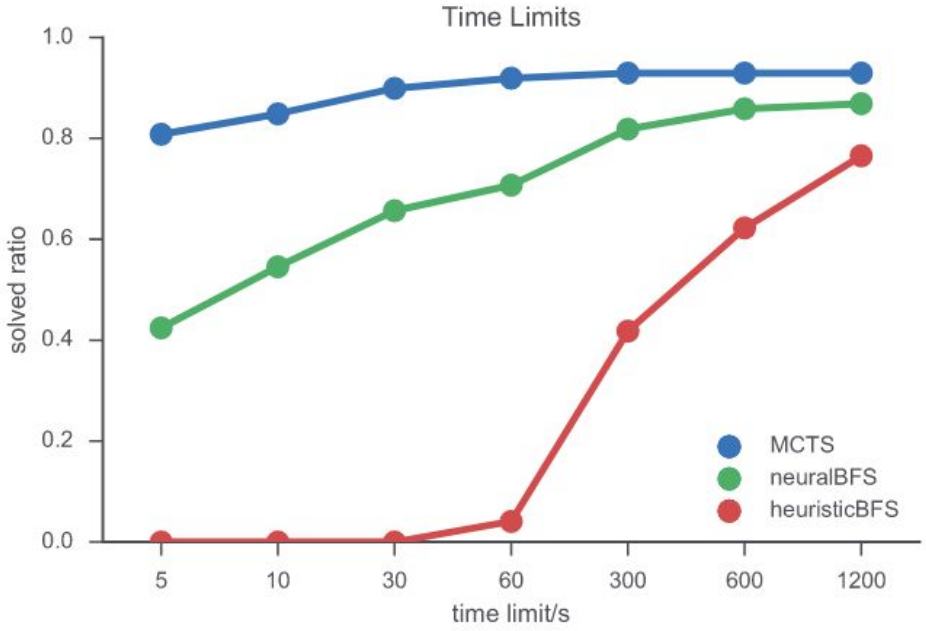
Synthesis Planning with MCTS

	AlphaGo Zero	3N-MCTS
Algorithm		
Neural Nets	$(p, v) = f(s)$ **p:** probability of selecting each move from a list of action probabilities **v:** scalar evaluation that estimates the probability of the current player winning from position s	$q = f_{roll}(s)$, $t = f_{exp}(s)$, $p = f_{scope}(s, r)$ **q:** scalar evaluation of node **r:** reactions between molecules **t:** possible transformations **p:** probability of the molecules reacting
Goal	Select the set of actions (from a fixed set of actions) that will lead to victory! Take that Lee Sedol!	Selecting the set of transformations (from a fixed set of transformations) that will help us find new drugs to cure diseases! Take that cancer!

Results & Discussion

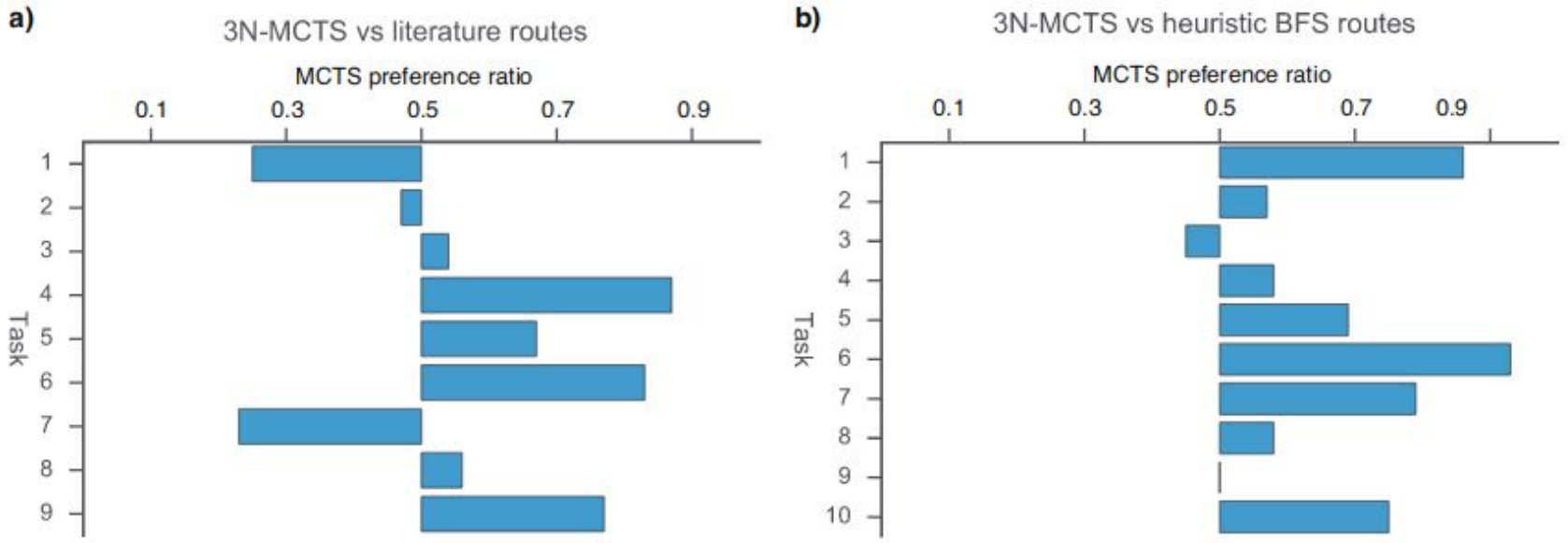
- Comparison with related methods
- Preference of chemical experts
- Limitations

Comparison with related methods



Method / Time lim	3N-MCTS	nBFS	hBFS
5 sec	**80%**	40%	0%
60 sec	**92%**	71%	4%
1200 sec	**~93%**	~80%	~75%

Preference of Chemical Experts



Limitations

- Not enough train data for some tasks
- Stereochemistry
- Not totally admitted by the industry

Thank You

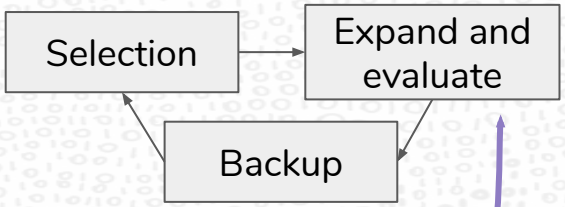
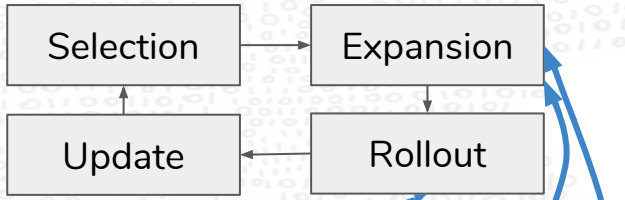
References

Background image: http://turnoff.us/geek/binary-tree (with changes)

Alpha Go content:
http://discovery.ucl.ac.uk/10045895/1/agz_unformatted_nature.pdf

Learning to Plan Chemical Synthesis content:
https://arxiv.org/pdf/1708.04202.pdf

Synthesis Planning with MCTS

	Alpha Go Zero	3N-MCTS
Algorithm		
Neural Nets	$(p, v) = f(s)$ **p:** probability of selecting each move from a list of action probabilities **v:** scalar evaluation that estimates the probability of the current player winning from position s	$q = f_{roll}(s)$, $t = f_{exp}(s)$, $p = f_{scope}(s, r)$ **q:** scalar evaluation of node ; **r:** reactions between molecules **t:** possible transformations ; **p:** probability of the molecules reacting
Objective	Maximise an upper confidence bound on Q(s,a) + U(s,a) where, U(s,a) ∝ P(s,a)/(1+N(s,a)) **Q(s,a)**: action-value ; **N(s,a)**: count visit ; **P(s,a)**: prior probability	Maximise the Q function which includes an adjustable objective $W(b_i)$ $Q(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{n} \mathbb{I}_i(s,a) z_i W(b_i)$ where, **N(s,a)**: count visit ; z_i: reward received during rollout