Progress Presentation

May 12th - May 22nd

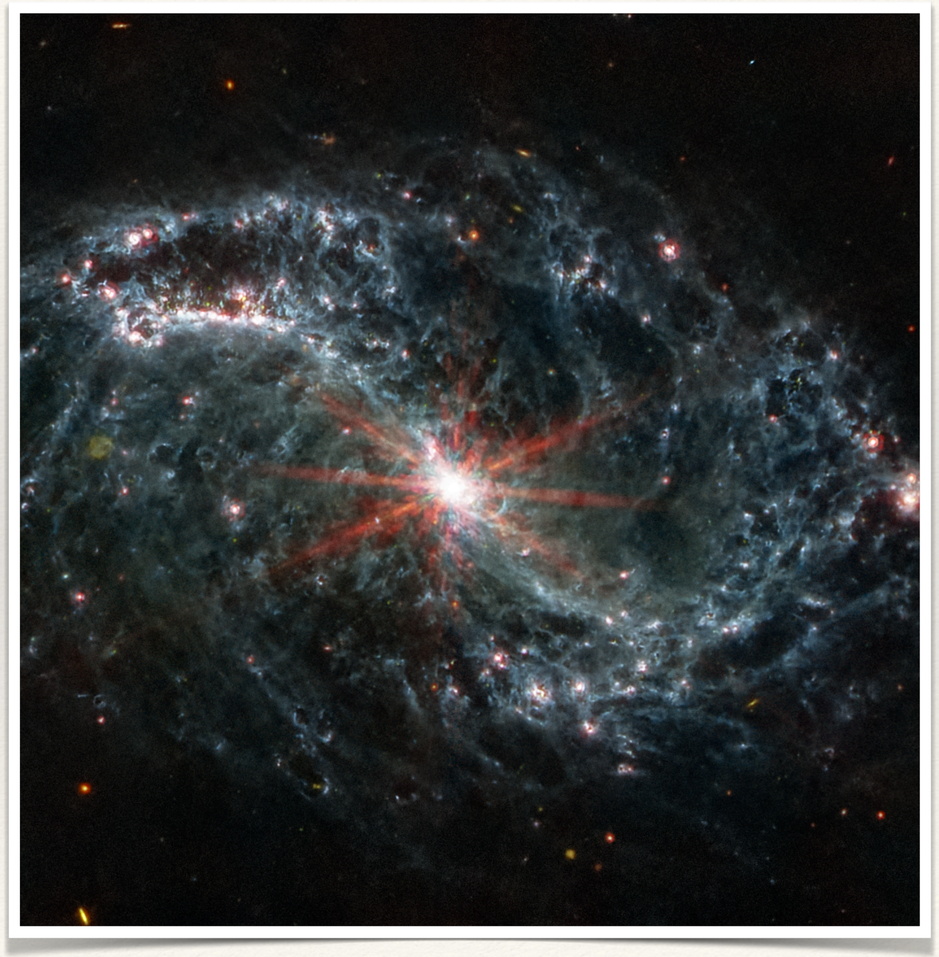
Agenda

Recap of new algorithm and some small tweaks

Issues with belief update

Some preliminary results with two agents

What kind of strategies do civilisations in the universe employ to ensure their survival, and how do these strategies change over time and space?



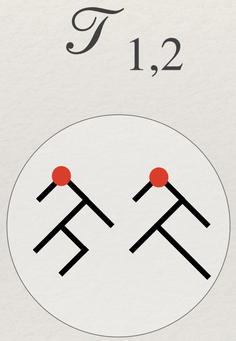
The galaxy NGC 7496 as captured by JWST

Recap: new algorithm

the same hierarchical tree structure as I-NTMCP

$\Rightarrow \sum_{d=0}^{L} (n-1)^d = \mathcal{O}(n^L)$ trees for n agents at max. level L

Level 2



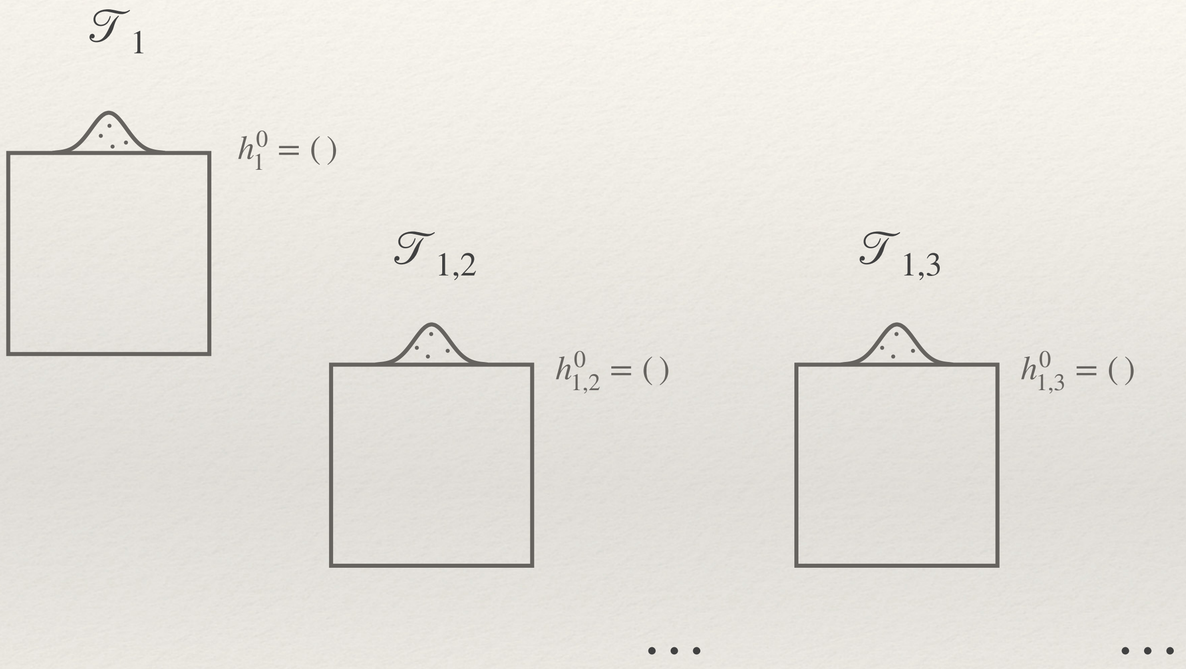
Level 1

Represents 1's beliefs about 3's beliefs about 2's beliefs

Level 0

● = root node

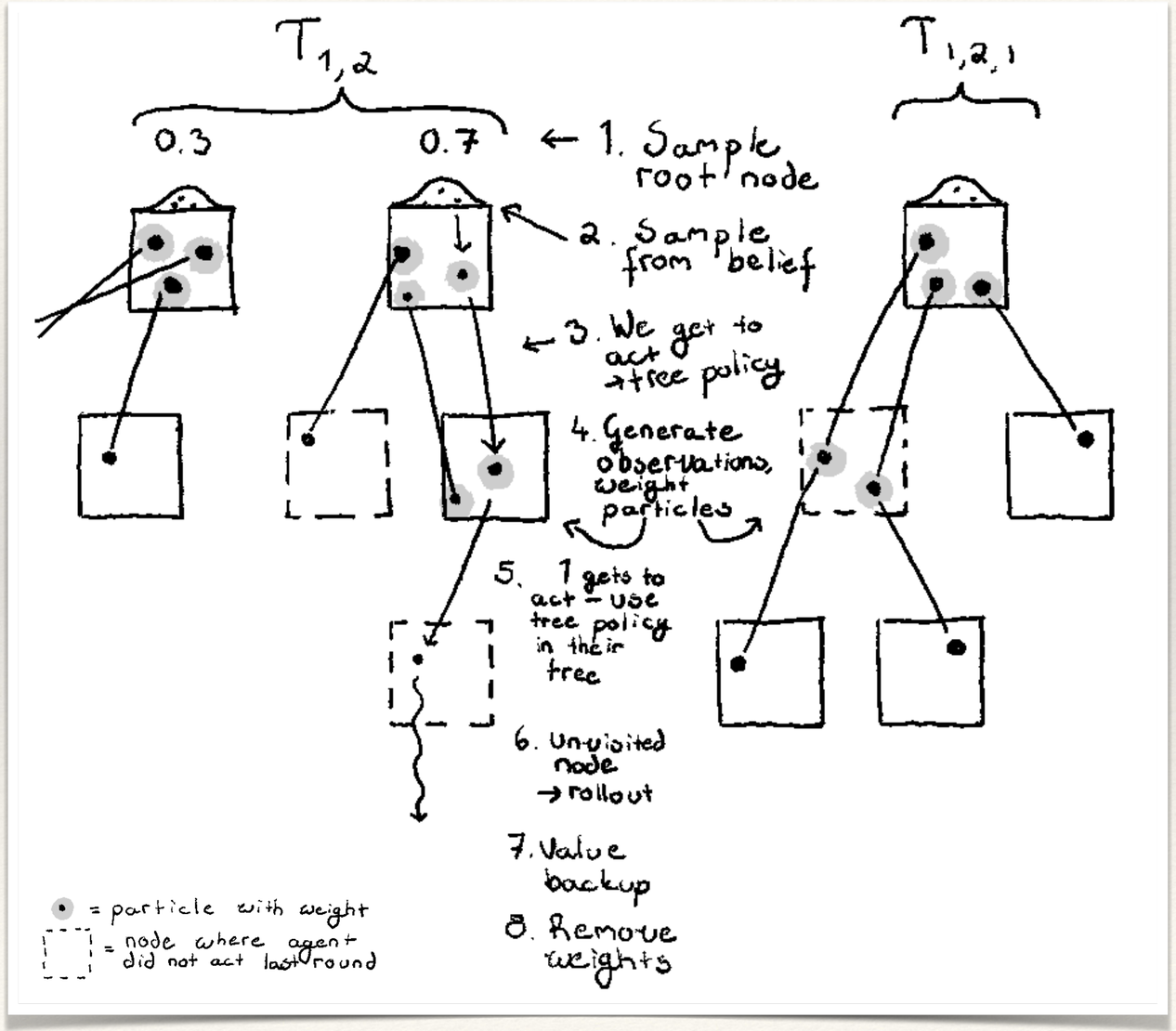
Recap: initialisation



- Each node in a tree corresponds to an agent action history
- A node contains a set of particles
  - particles are **history states**: they contain an environment state and a joint action history
- In addition, root nodes contain a separate set of "belief particles" which are history states as well, but have weights
- Initially all trees have a single root node (empty agent action history), but in general the lower-level trees can have multiple root nodes
  - top level tree (here $\mathcal{T}_1$) always has a unique root node

Recap: planning

- planning consists of expanding each tree a given numbers of times
- bottom up: first we plan at level 0, then level 1, and so on



Recap: tree policy

- The tree policy resembles MCTS's UCT method, but has to be adapted slightly
- When the particles p in a node N all have weights $b(p)$, the particles represent a belief b. Calculate:

  $N_+(b)$ = number of particles with weight > 0 under b

  $W(b) = \sum_{p \in N} b(p)$ = total weight of particles

  $W(b,a) = \sum_{p \in N, p.a = a} b(p)$ = total weight of particles that were next propagated with a

  $\tilde{W}(b,a) = \frac{W(b,a)}{W(b)} N_+(b)$

  $Q(b,a) = \frac{1}{W(b,a)} \sum_{p \in N : p.a = a} b(p) V(p)$ = weighted average of particle values

- Then the chosen action is one that maximises $Q(b,a) + c\sqrt{\frac{\log N_+(b)}{\tilde{W}(b,a)}}$ (c is exploration constant)

Tree policy for opponents' trees

- A softargmax policy is used for sampling actions from the opponents' trees when expanding a tree
  - the opponents' trees (one level lower) only approximate the optimal actions, and the optimal action given by the tree might be different from what the agent actually does
- The probability of choosing action a under belief b in the lower-level tree is

$$\mathbb{P}(a \mid b) \propto \exp\left(\frac{\tilde{W}(b,a)}{\sqrt{N_+(b)}}\right) = \exp\left(\frac{W(b,a)}{W(b)}\sqrt{N_+(b)}\right)$$

- For example: if proportions of action weights are (0.7, 0.2, 0.1), the action probabilities are (0.74, 0.15, 0.11) if $N_+(b) = 10$ and (0.99, 0.01, 0.00) if $N_+(b) = 100$

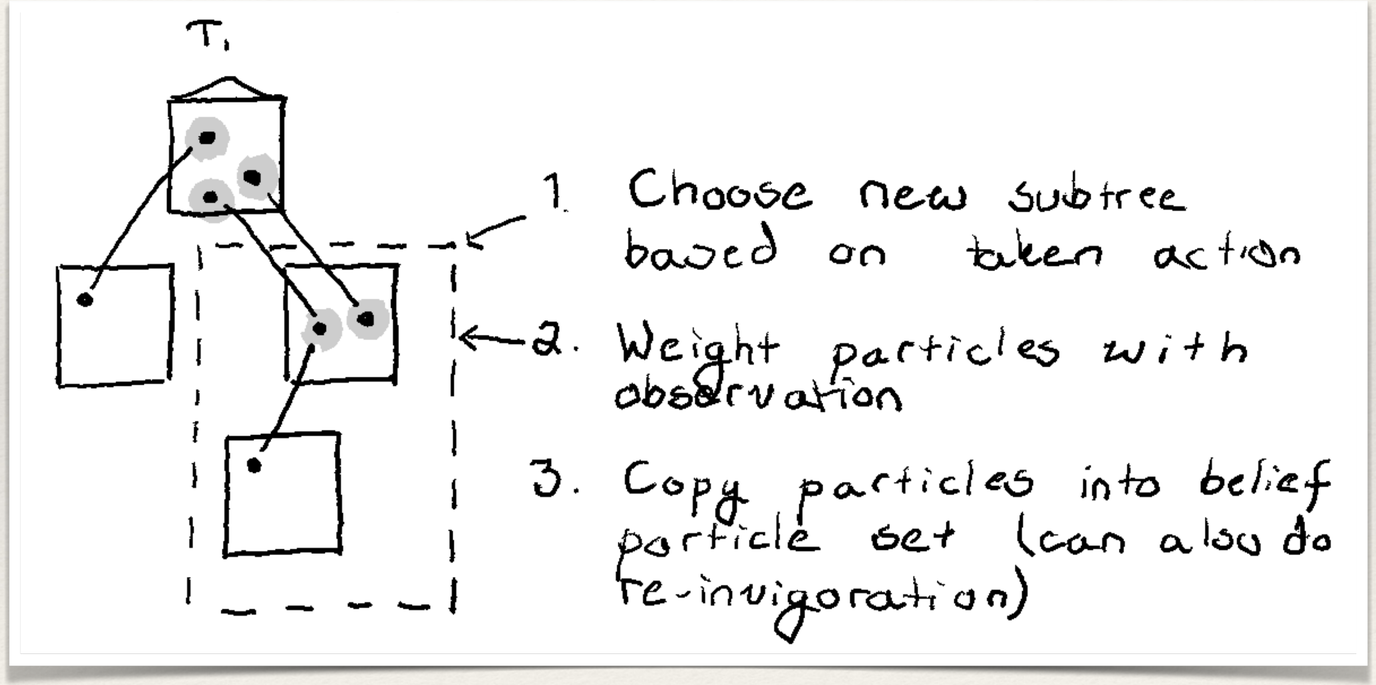
Recap: belief update

- Agent takes an action and receives an observation
- Belief update happens top down: first the top-level tree, then the trees below them, etc.
- Need to:

i. find weights for new root nodes

ii. create beliefs at each new root node

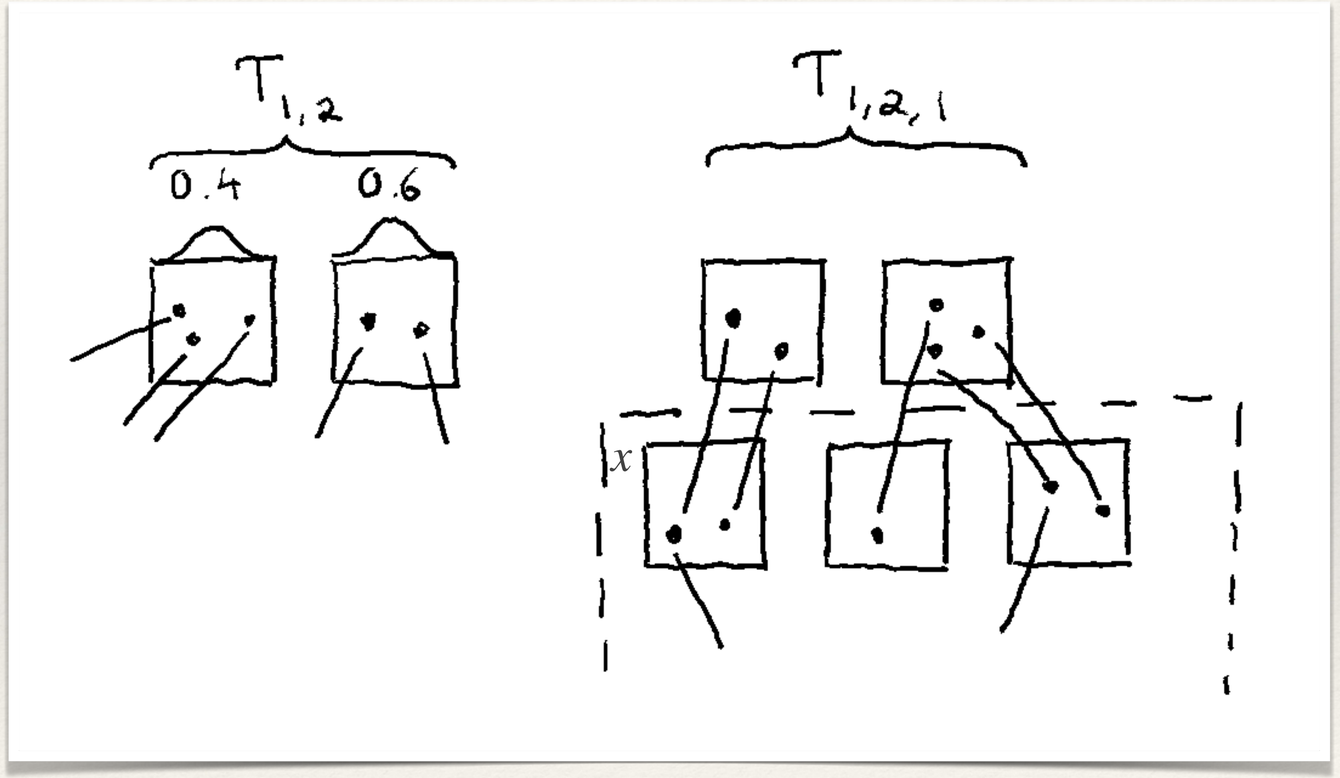
Belief update for the top-level tree



Recap: belief update (lower-level trees)

i. to find a weight for a new root node *x*:

1. calculate ~~proportion~~ sum of weights of particles in each parent tree root node where joint history matches agent history of *x*
2. weight ~~proportions~~ sums of weights by parent tree root node weights and sum



Recap: belief update (lower-level trees)

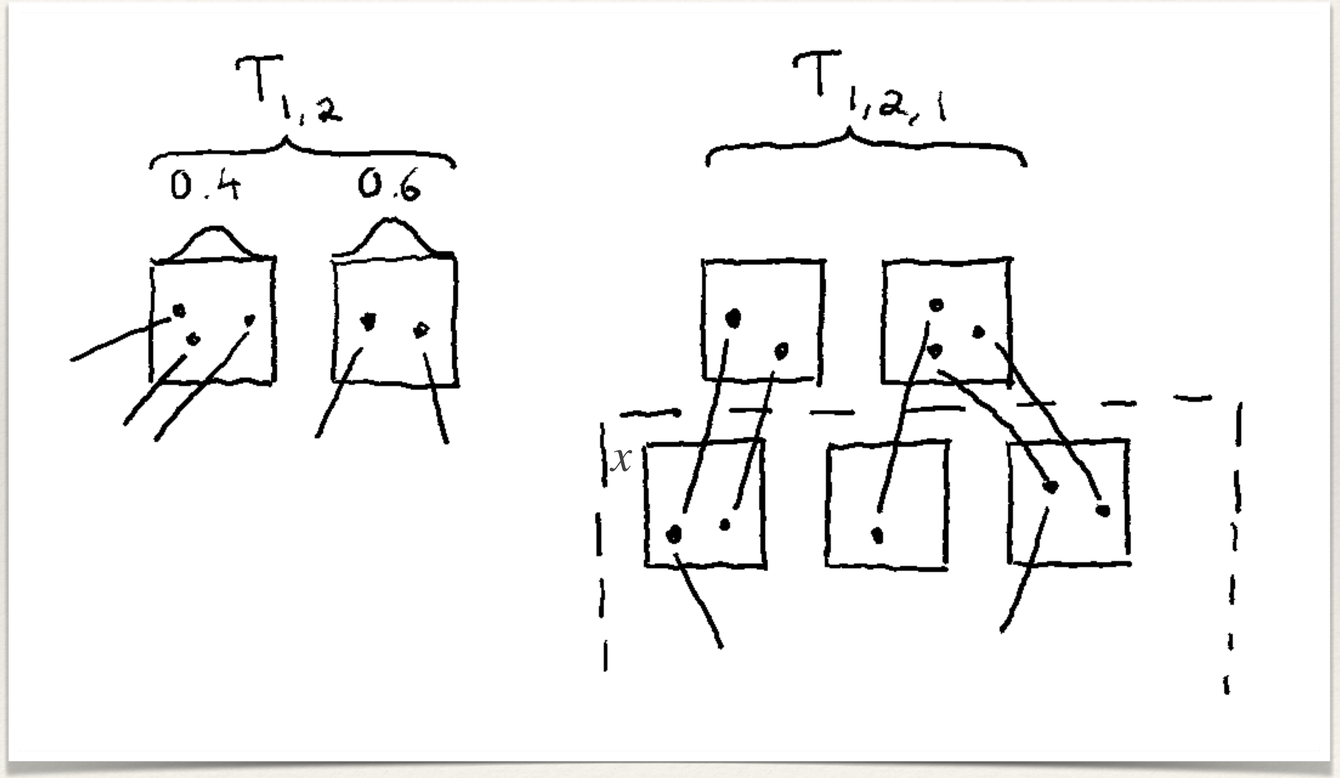
To create a belief for new root node x, first find all matching particles in all parent root nodes and give them weights according to

matching particle weight = particle weight × parent root node weight

Then repeat n times:

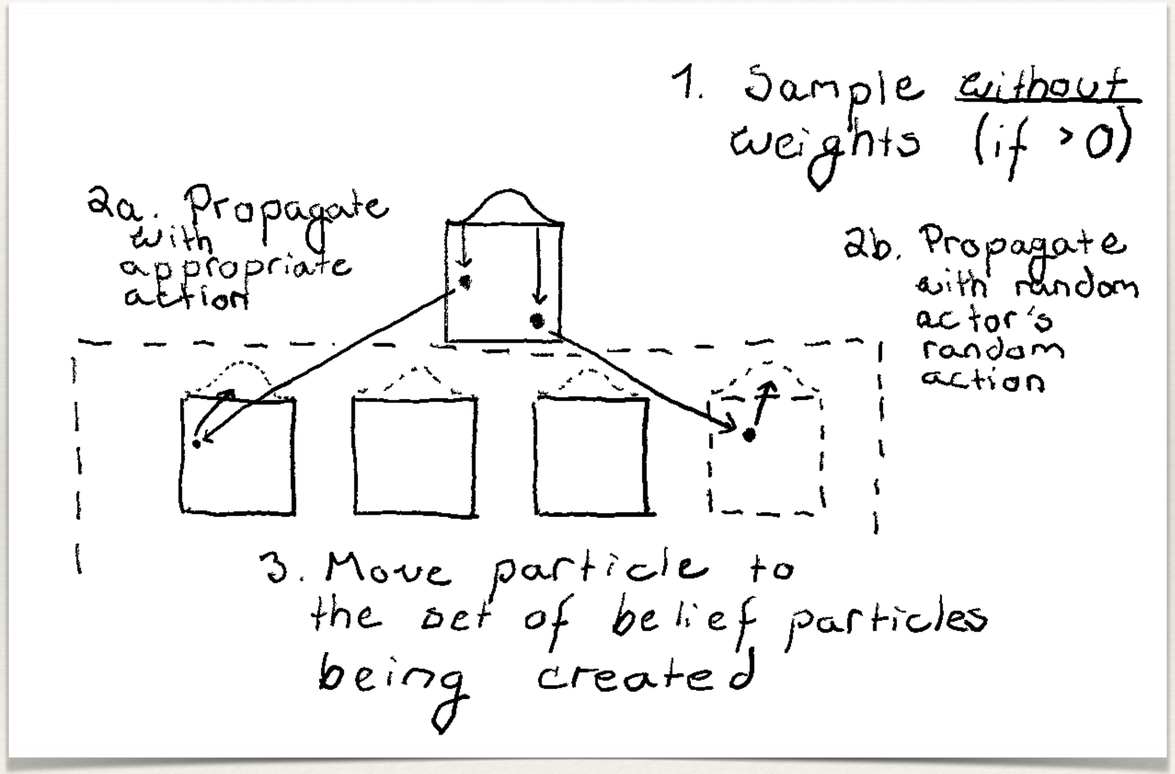
- ~~1. sample parent tree root node (with weights)~~
- ~~2. from the chosen node, sample (with weights) a particle from among the particles matching x~~
- 1. Sample (with weights) a matching particle
- 2. generate observation and weight particles in x
- 3. sample (with weights) a particle from x and increase its count by 1

- finally, divide counts by n to get approximate weights for particles
- root belief particle set can then be created as before



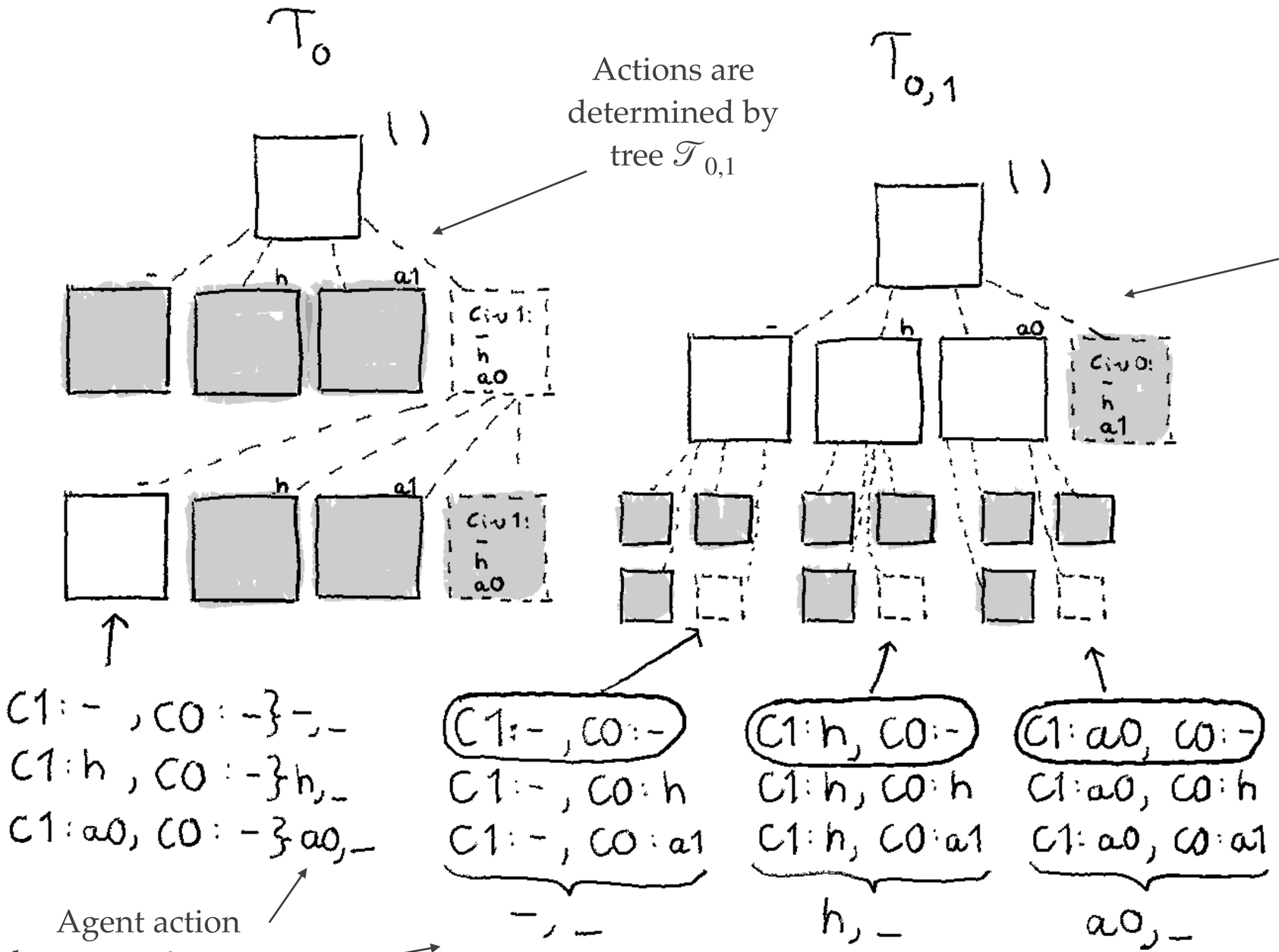
Reinvigoration method

- After the weights $w(h_k)$ of a trees root nodes h_k are determined, $w(h_k)N_{\text{reinvig}}$ particles are created for each one
- In total, N_{reinvig} particles are created per tree



Issue

- **All root nodes get pruned in belief update**
- We are updating the tree $\mathcal{T}_{0,1}$. There are two ways a node with agent action history h_1 can get pruned:
  - there are no particles in the parent tree's ($\mathcal{T}_0$) root nodes that have a joint action history $h = (\cdot, h_1)$
  - more rarely: there are compatible particles, but when weighting particles in one of $\mathcal{T}_{0,1}$'s root nodes, all get weight 0
    - this can happen e.g. if in all compatible particles agent 1 is attacked but not in any of the particles to be weighted
- So the fundamental issue is that **the actions of agent 1 sampled in $\mathcal{T}_0$ are not the same that are sampled in $\mathcal{T}_{0,1}$**


T_0
$T_{0,1}$
Actions are determined by tree $\mathcal{T}_{0,1}$
Actions are determined by level 0 default policy (currently random)
Agent action histories of agent 1

Results: attacking costs

n_agents: 2

agent_growth: sigmoid

agent_growth_params: {speed_range: (0.3, 1), takeoff_time_range: (10, 100)}

rewards: {destroyed: -1, hide: -0.01, attack: -0.1}

n_root_belief_samples: 1000

n_tree_simulations: 200

n_belief_update_samples: 100

n_reinvigoration_particles': 100

obs_noise_sd: 0.1

reasoning_level: 2

action_dist_0: random

discount_factor: 0.9

discount_epsilon: 0.05

exploration_coef: 1

visibility_multiplier: 0.5

decision_making: ipomdp

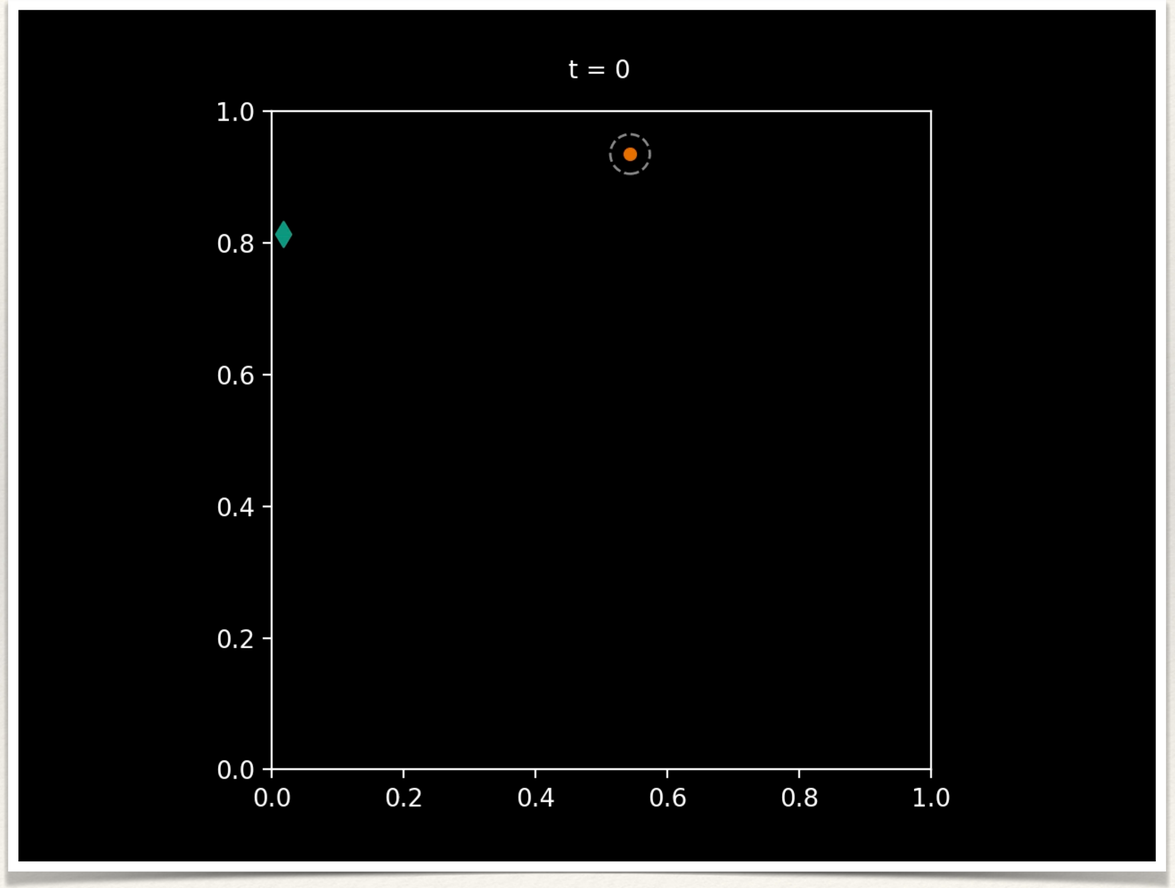
init_age_belief_range: (10, 100)

init_age_range: (10, 100)

init_visibility_belief_range: (1, 1)

init_visibility_range: (1, 1)

n_steps = 100



Results: attacking costs

n_agents: 2

agent_growth: sigmoid

agent_growth_params: {speed_range: (0.3, 1), takeoff_time_range: (10, 100)}

rewards: {destroyed: -1, hide: -0.01, attack: -0.1}

n_root_belief_samples: 1000

n_tree_simulations: 200

n_belief_update_samples: 100

n_reinvigoration_particles': 100

obs_noise_sd: 0.1

reasoning_level: 2

action_dist_0: random

discount_factor: 0.9

discount_epsilon: 0.05

exploration_coef: 1

visibility_multiplier: 0.5

decision_making: ipomdp

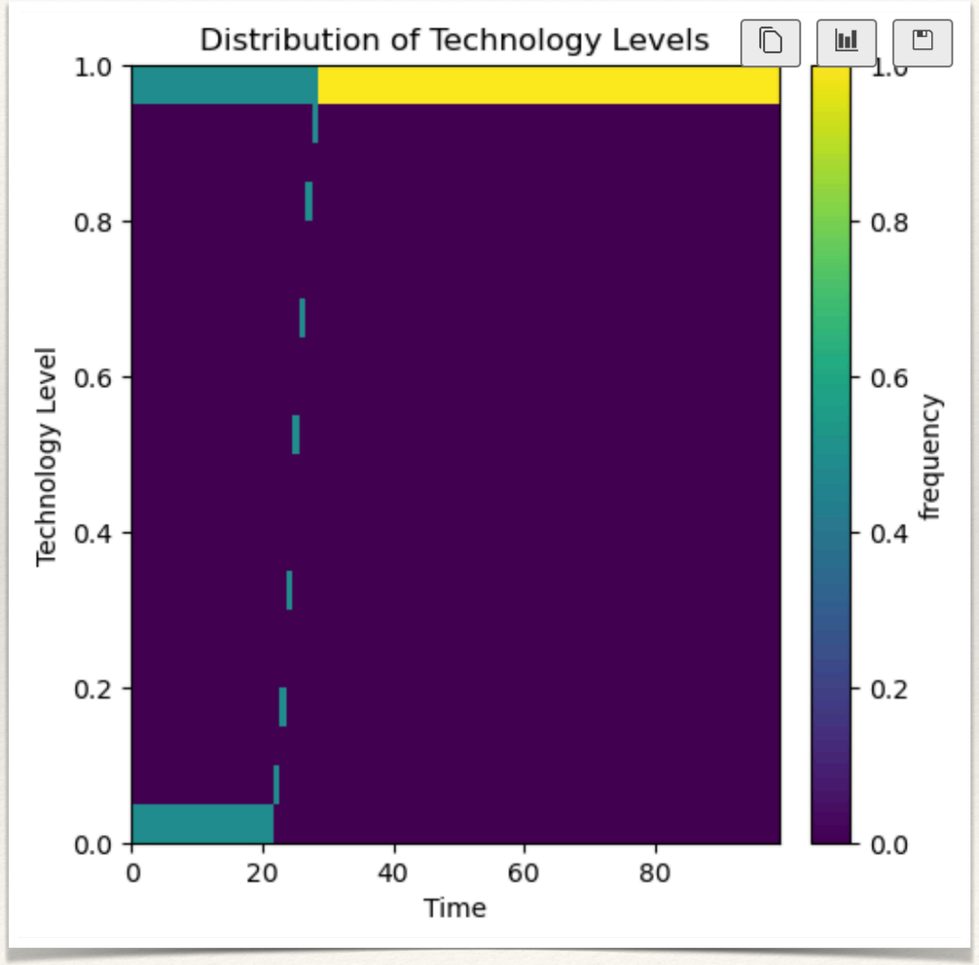
init_age_belief_range: (10, 100)

init_age_range: (10, 100)

init_visibility_belief_range: (1, 1)

init_visibility_range: (1, 1)

n_steps = 100



Results: attacking is free

n_agents: 2

agent_growth: sigmoid

agent_growth_params: {speed_range: (0.3, 1), takeoff_time_range: (10, 100)}

rewards: {destroyed: -1, hide: -0.01, attack: 0}

n_root_belief_samples: 1000

n_tree_simulations: 200

n_belief_update_samples: 100

n_reinvigoration_particles': 100

obs_noise_sd: 0.1

reasoning_level: 2

action_dist_0: random

discount_factor: 0.9

discount_epsilon: 0.05

exploration_coef: 1

visibility_multiplier: 0.5

decision_making: ipomdp

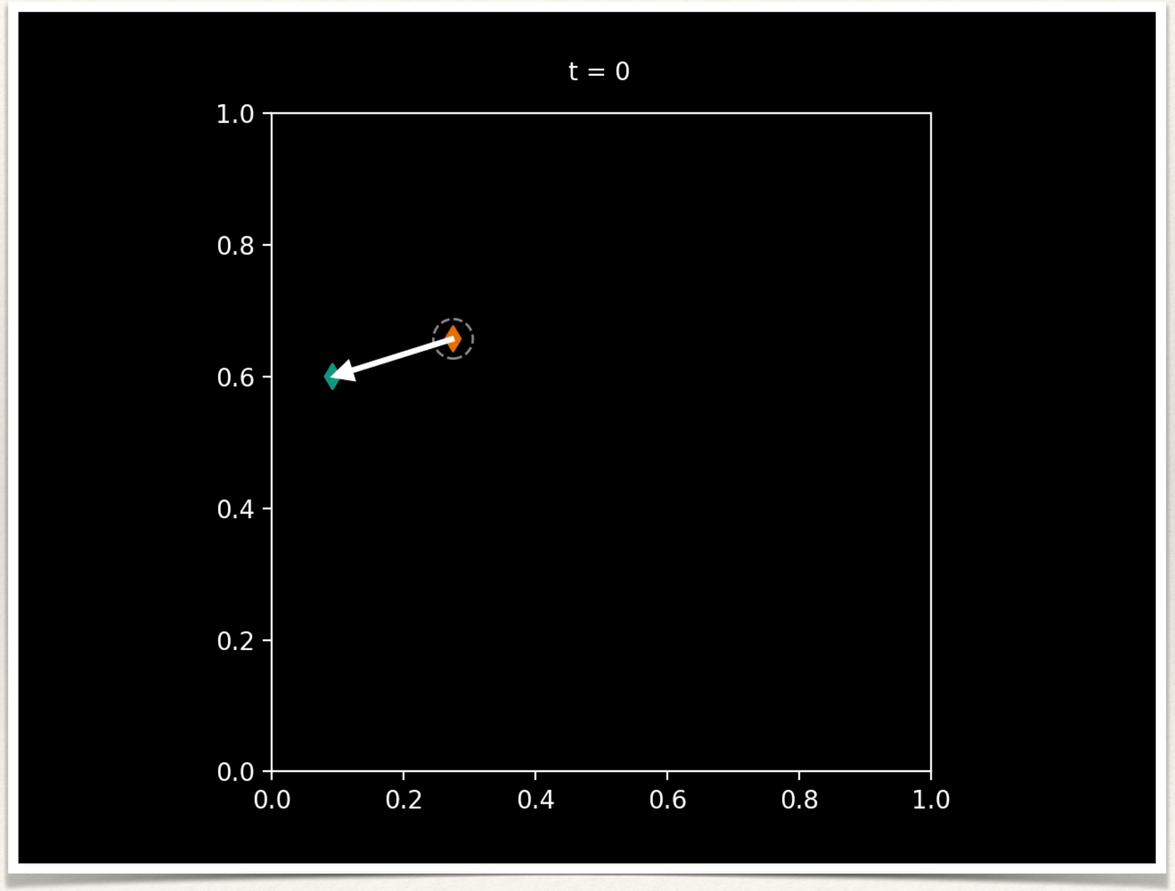
init_age_belief_range: (10, 100)

init_age_range: (10, 100)

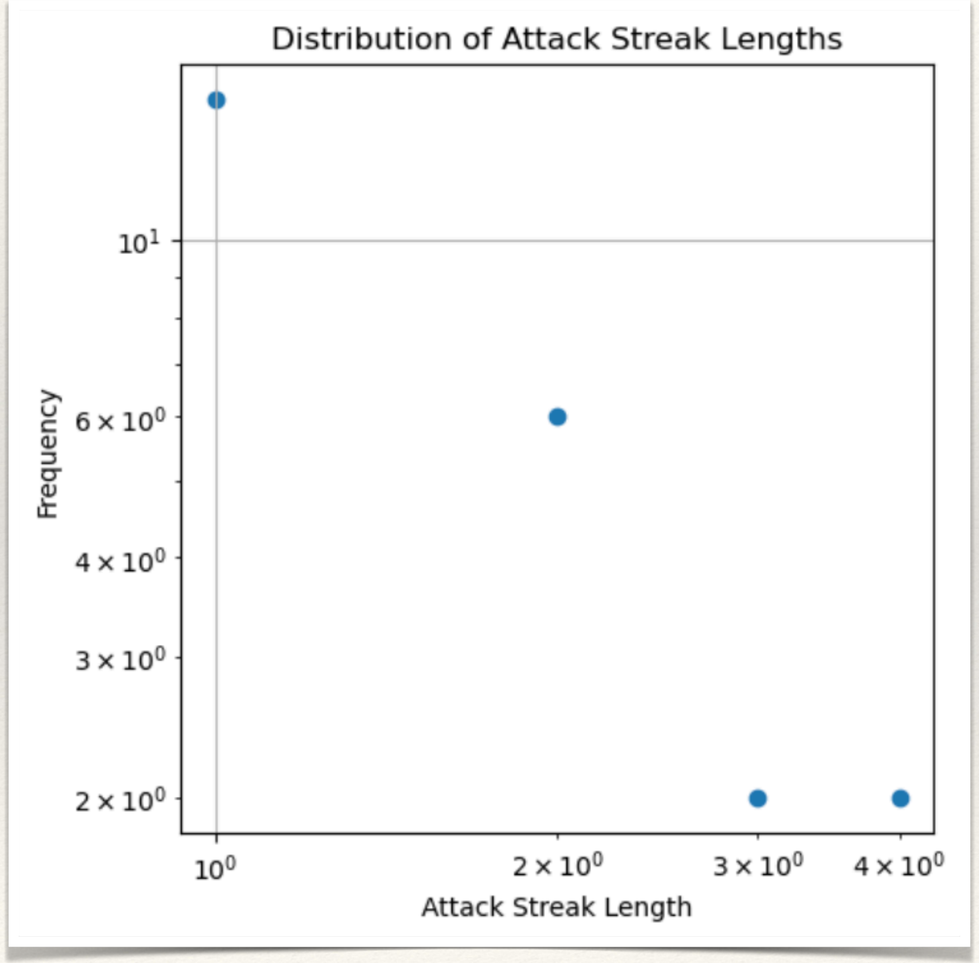
init_visibility_belief_range: (1, 1)

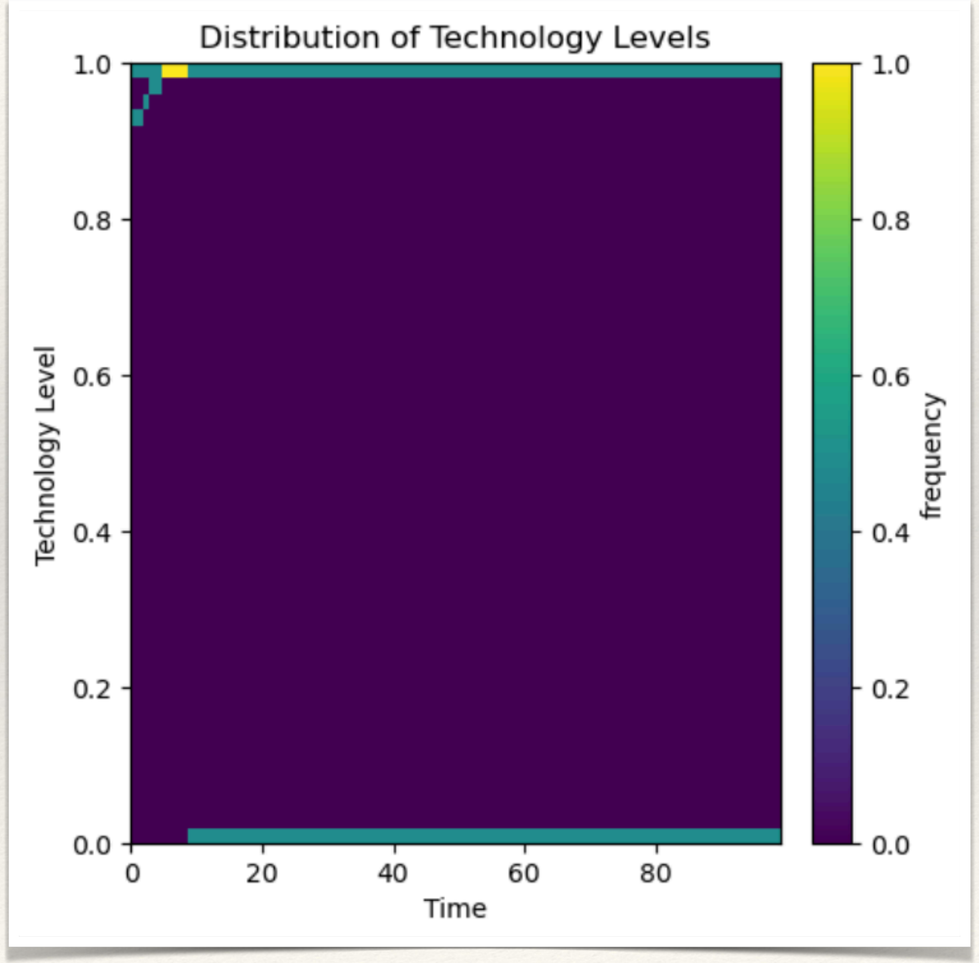
init_visibility_range: (1, 1)

n_steps = 100



Results: attacking is free





Updates to observations