# Planning and Control using Model-Based Reinforcement Learning

Pavel Czempin, Vincent Friedrich,
Ruotong Liao, Jannik Nettelnstroth
Mentors: Mathias Sundholm,
Hamdi Belhassen
Co-Mentor: Michael Rauchensteiner

24. February 2021

PreciBake

# Introduction

# Problem description

*Challenge:* **Optimal production decisions** in a bakery

PreciBake®

*Approach:* **Model-Based Reinforcement Learning**

*Goals:*

1. **Maximize sales**
2. **Minimize waste**
3. **Minimize shelf-time**

# Our environment


**Producer Model**

- models an oven
- up to 30 products
- Agent can interact with this part

**Inventory**

- which products are ready for sale?

**Consumer Model**

- generates orders
- stochastic part of the environment

Image Source: freepik.com

PreciBake

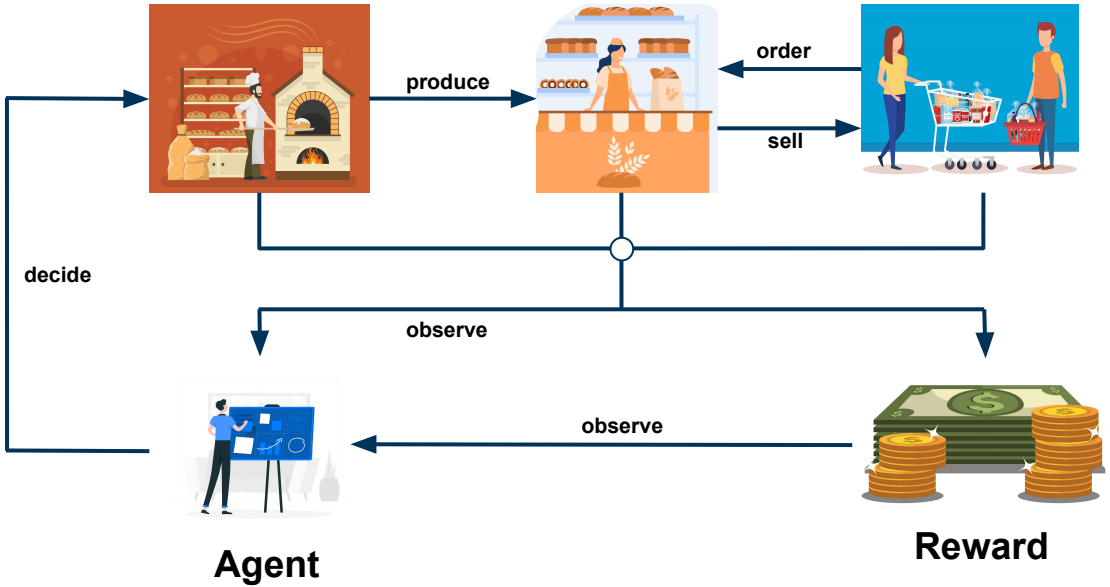
# Complete setting

**Agent**

- decides when and which products to bake
- tries to maximize the reward

**Reward**

- measures performance of the agent with regard to the three goals
- is also used as metric to evaluate agent






PreciBake

Image Source: freepik.com

# Consumer models

**Poisson model**

➔ **samples orders** from a **poisson distribution** with predefined mean

**Real Data**

➔ **customer data** for **10 days in August** and **15 days in December** of the same year

PreciBake®

# Reward

| | |
|---|---|
| fulfilled-orders-ratio | $m_s = \frac{\#\text{sold products}}{\#\text{ordered products}}$ |
| product-waste-ratio | $m_w = 1 - \frac{\#\text{wasted products}}{\#\text{produced products}}$ |
| average_freshness | $m_f = \frac{1}{\#\text{sold products}} \sum_{\text{sold product } i} I[\text{age}_i \leq \text{freshness\_time}_i]$ |
| overall reward | $m = \frac{4m_s + 4m_w + m_f}{9}$ |

# Model-Based Reinforcement Learning

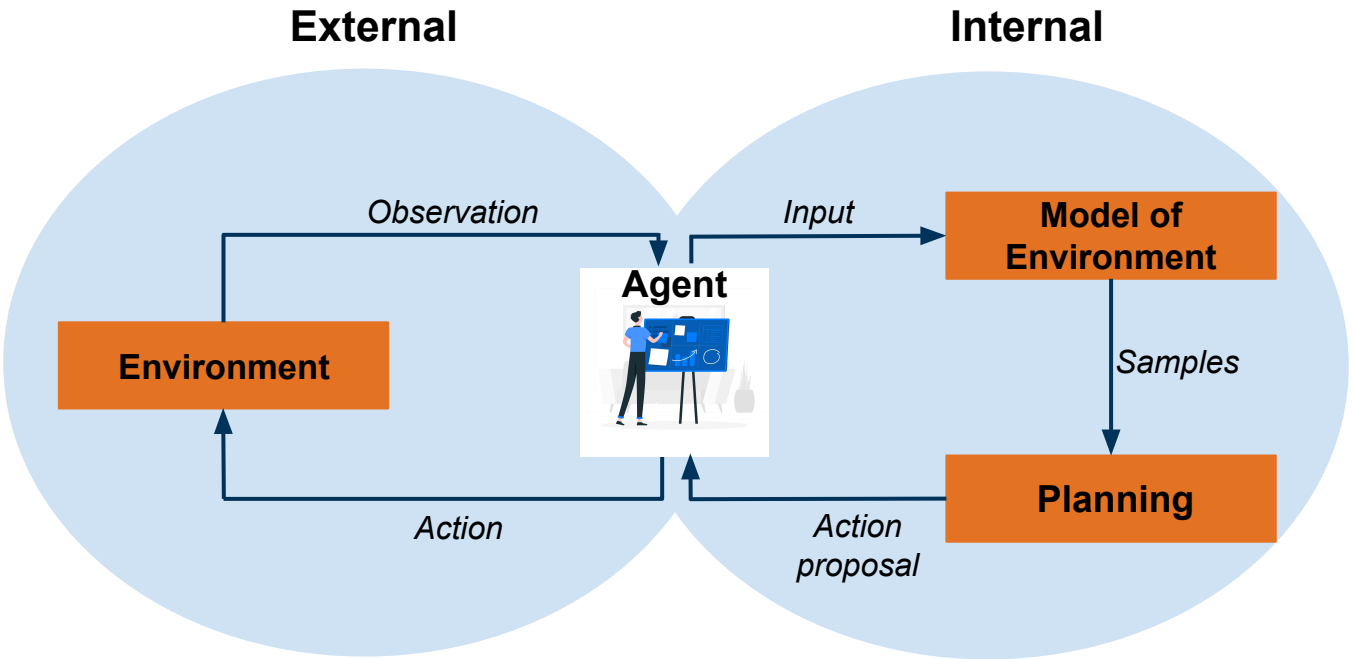



Image Source: freepik.com

PreciBake

# Hawkes Predictions

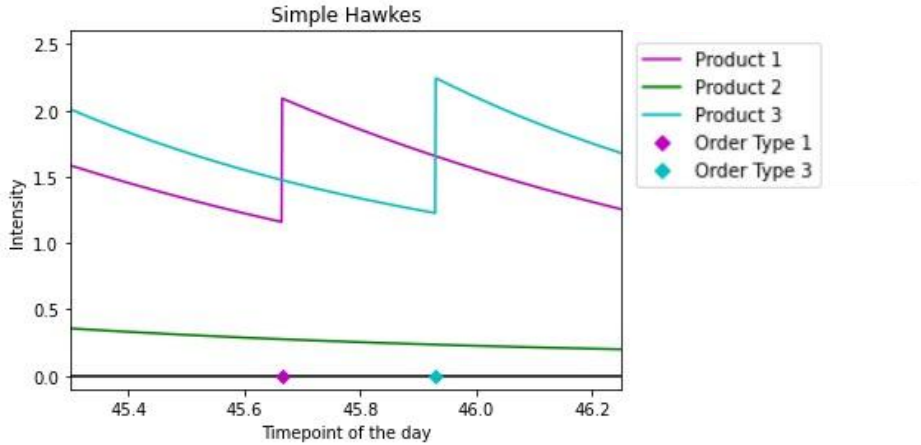
# Simple Hawkes Model

- specified by **Intensity function**
  - **Base rate** $\mu$
  - **Excitement parameter** $\alpha$
  - **Decay parameter** $\omega$
  - **past orders** *H(t)*

**Intensity function**

$$\lambda(t) = \mu + \alpha \sum_{q \in H(t)} e^{-\omega(t-q)}$$

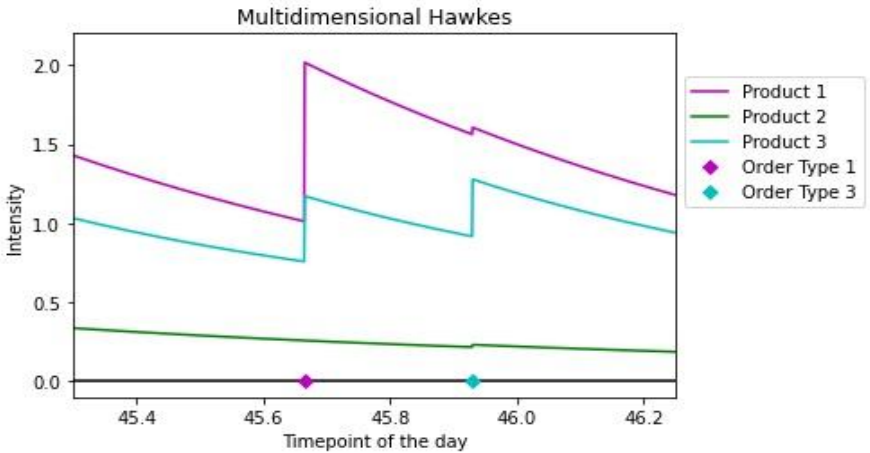
- **Orders** of different products are **independent**




PreciBake

# Multidimensional Hawkes Model

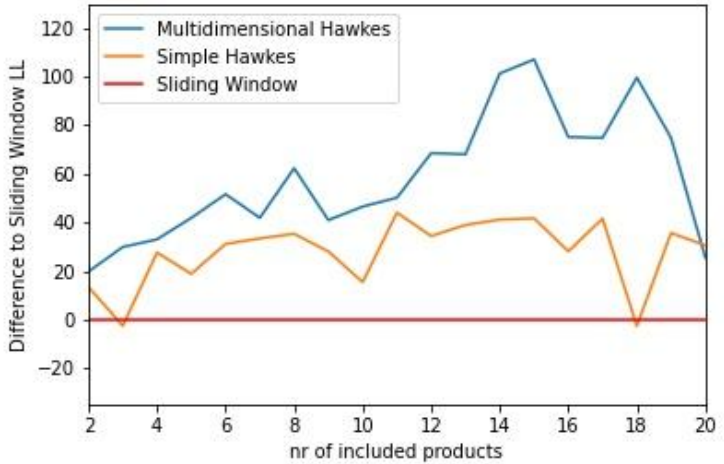
- **Extension** of the Simple Hawkes Model
- Intensity captures **interdependencies** between different products



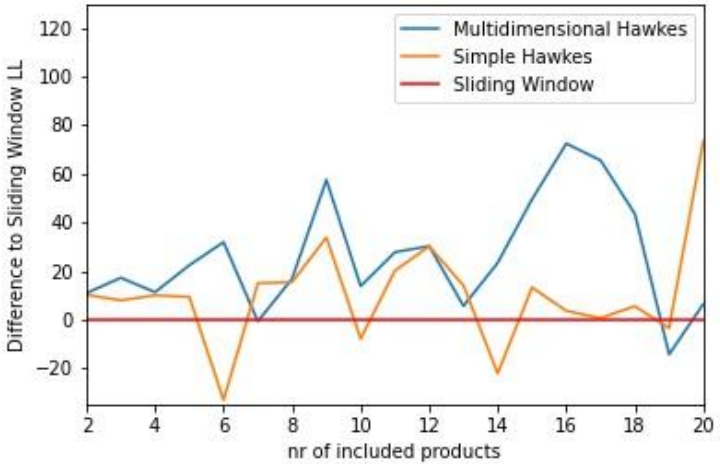

PreciBake®

# Hawkes Experiments on Real Data

**Train set (August Data)**




**Test set (December Data)**




- Comparison of **Multidimensional Hawkes**, **Simple Hawkes** and **Sliding Window** (baseline model)
- Measure: **relative average Log-Likelihood** compared to **baseline model**

PreciBake

# Monte Carlo Tree Search Planning

# Monte Carlo Tree Search

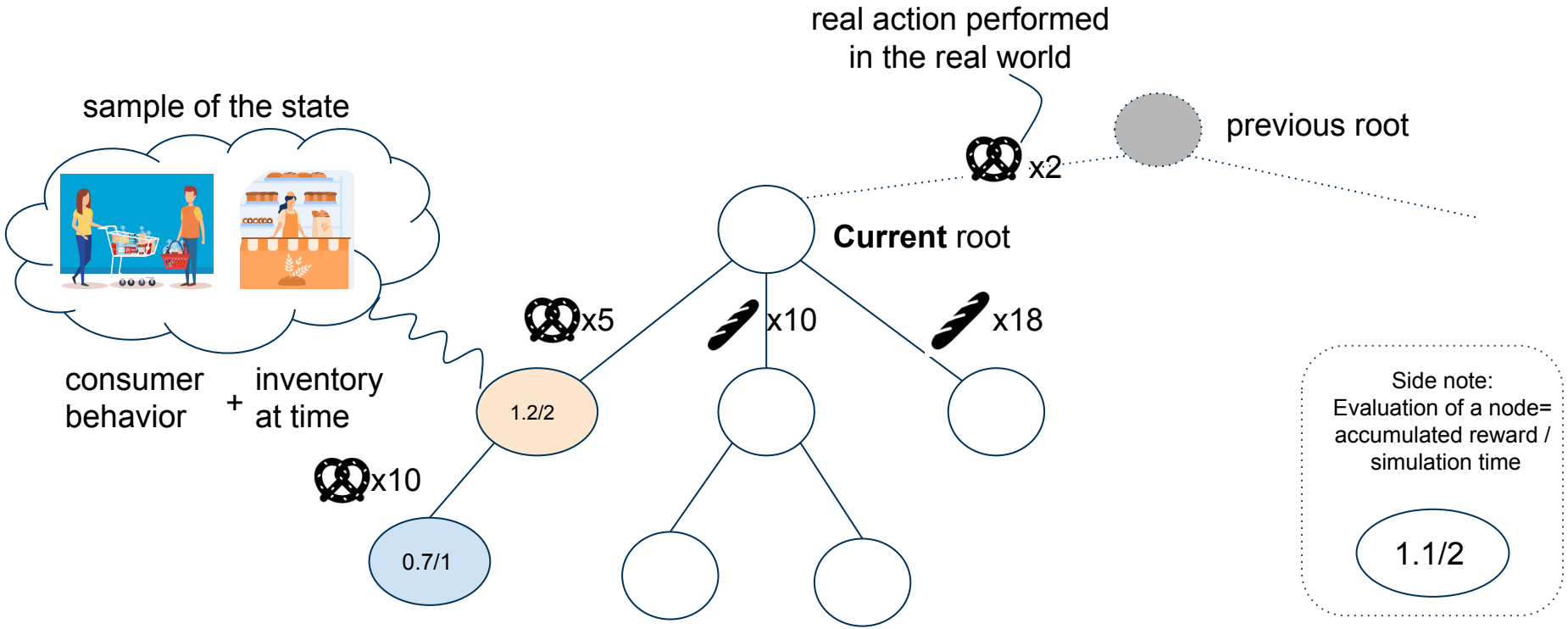
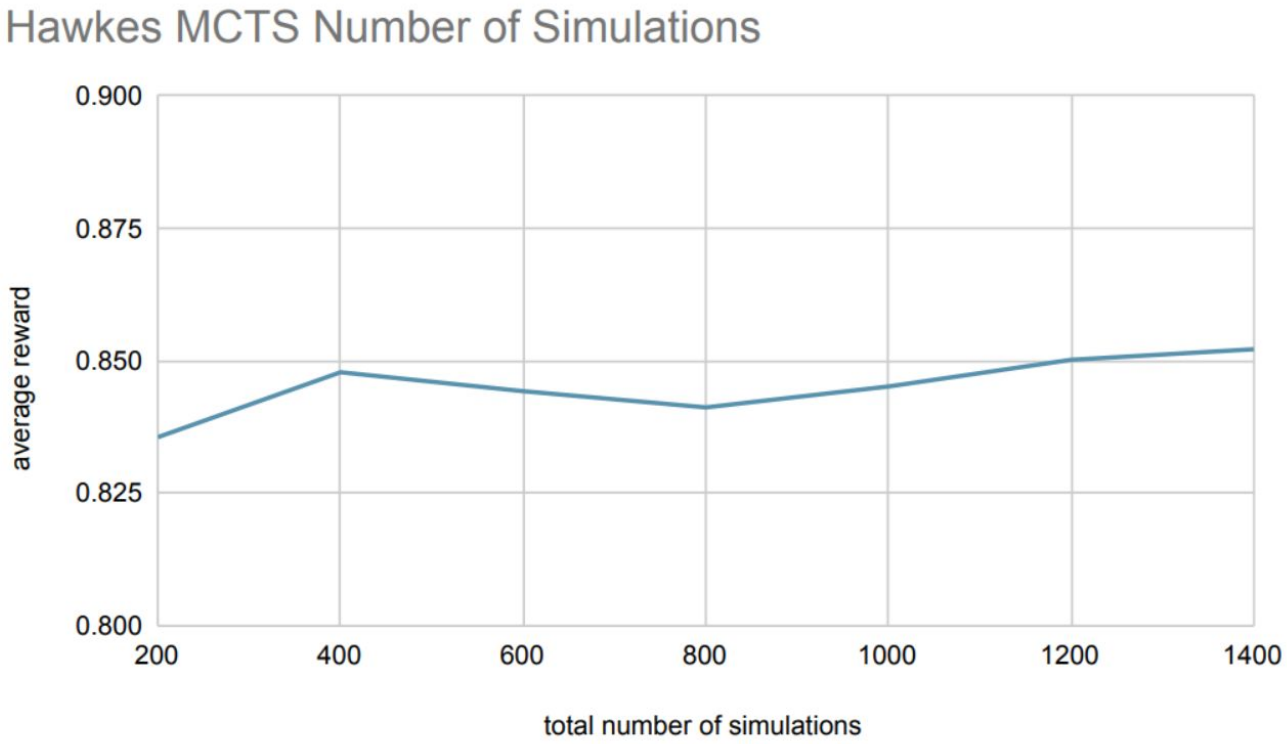



Image Source: freepik.com

PreciBake

# Monte Carlo Tree Search

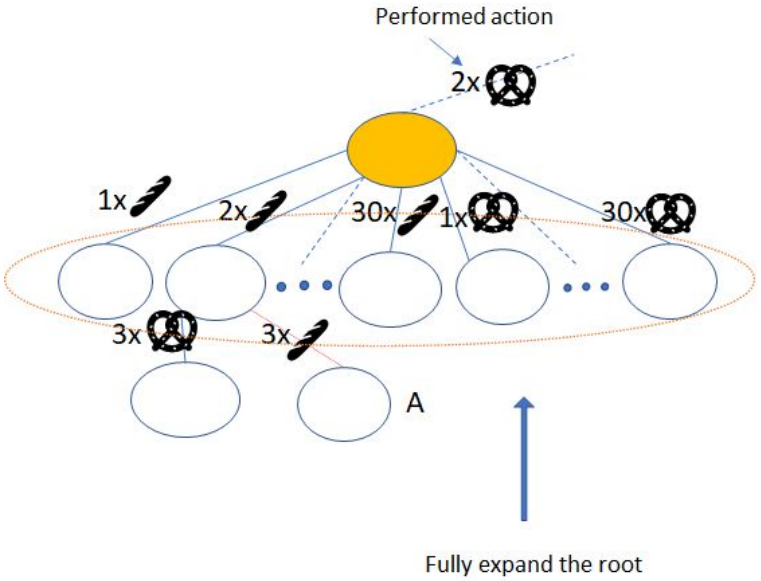
**Heuristic idea**: Simulation-based search algorithm relies on simulation budget



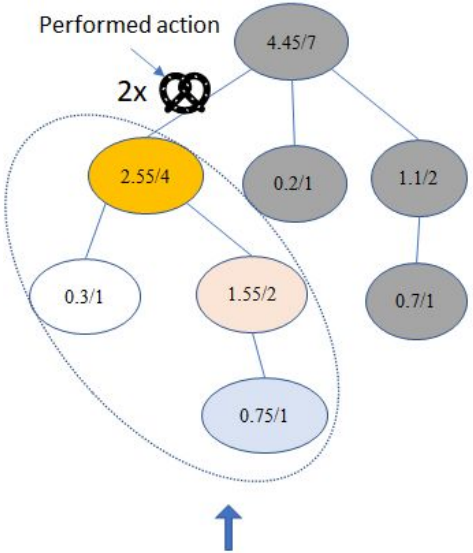

PreciBake®

# Monte Carlo Tree Search

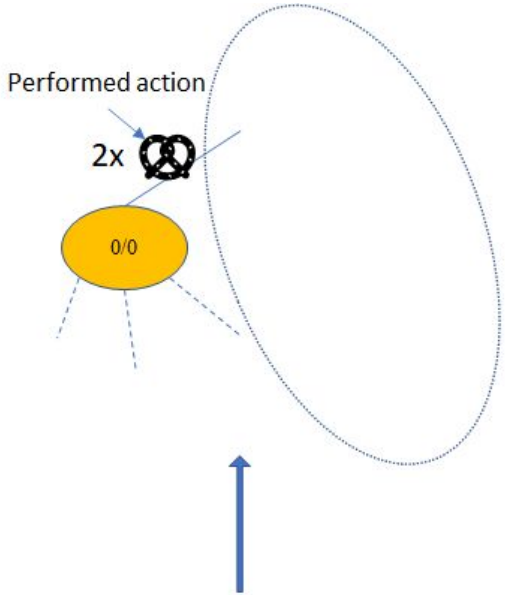
Tryouts to improve performance of the Monte Carlo Tree




Fully expand the root




Multiply with **discount factor** representing unreliability

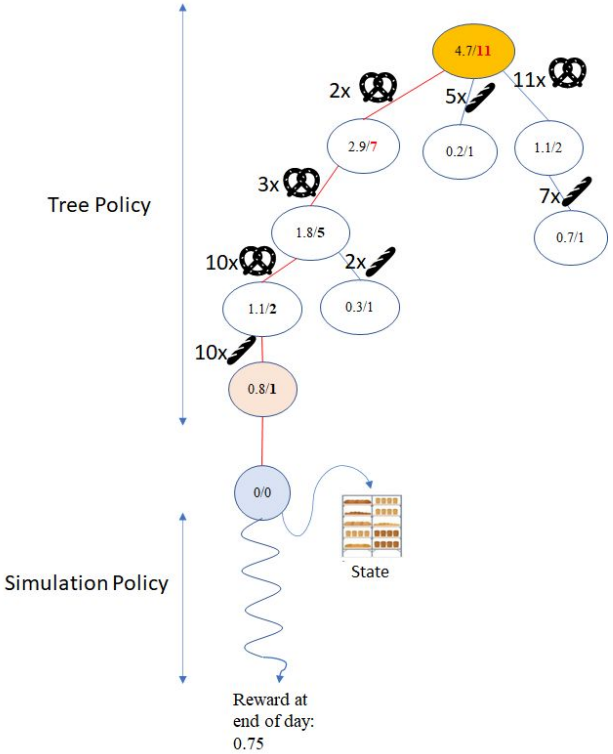



Rebuild a new tree

PreciBake

# Monte Carlo Tree Search

Problem: The tree tends to explore deep leaf nodes and produces **skewed** tree




Side note:
Evaluation of a node= accumulated reward / simulation time

1.1/2

Image Source: freepik.com

PreciBake

# Monte Carlo Tree Search

Problem: The tree tends to explore deep leaf nodes and produces skewed tree

Solution: Introduce **depth limitation to tree policy**

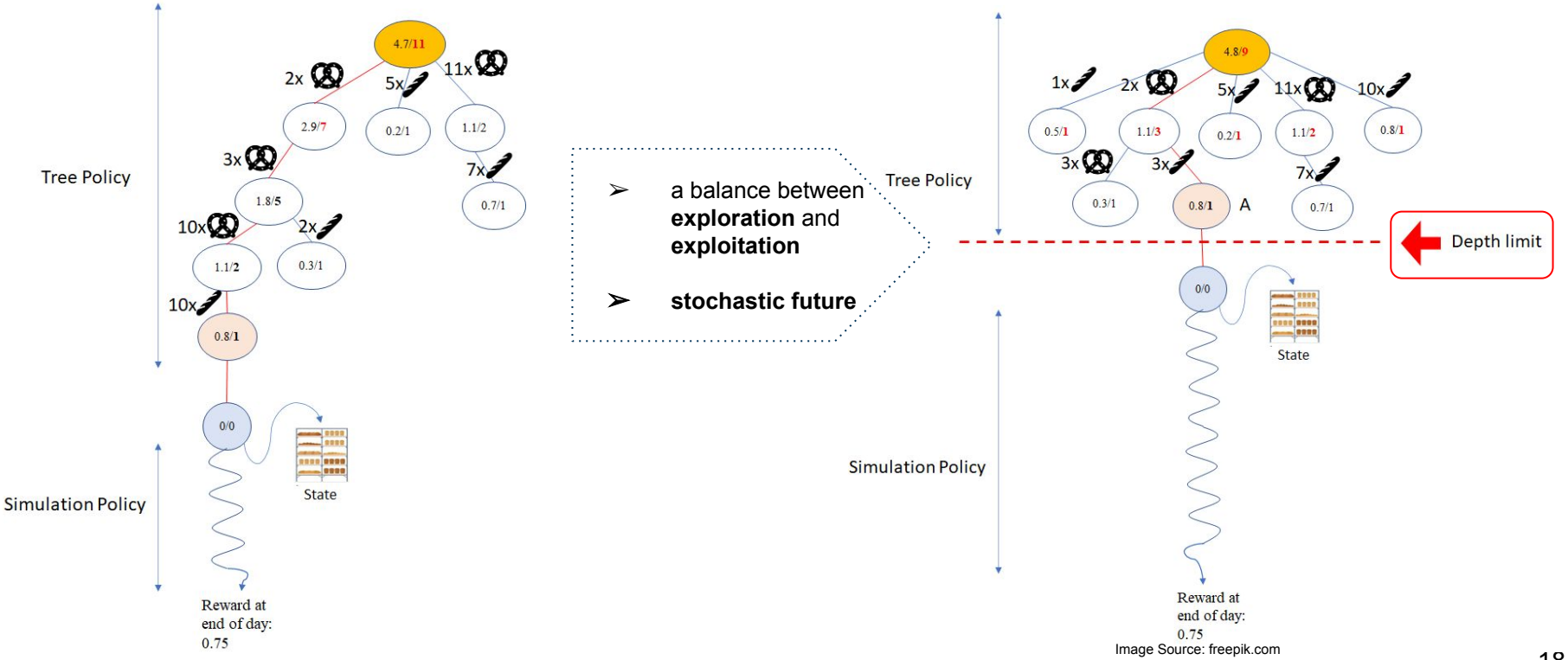
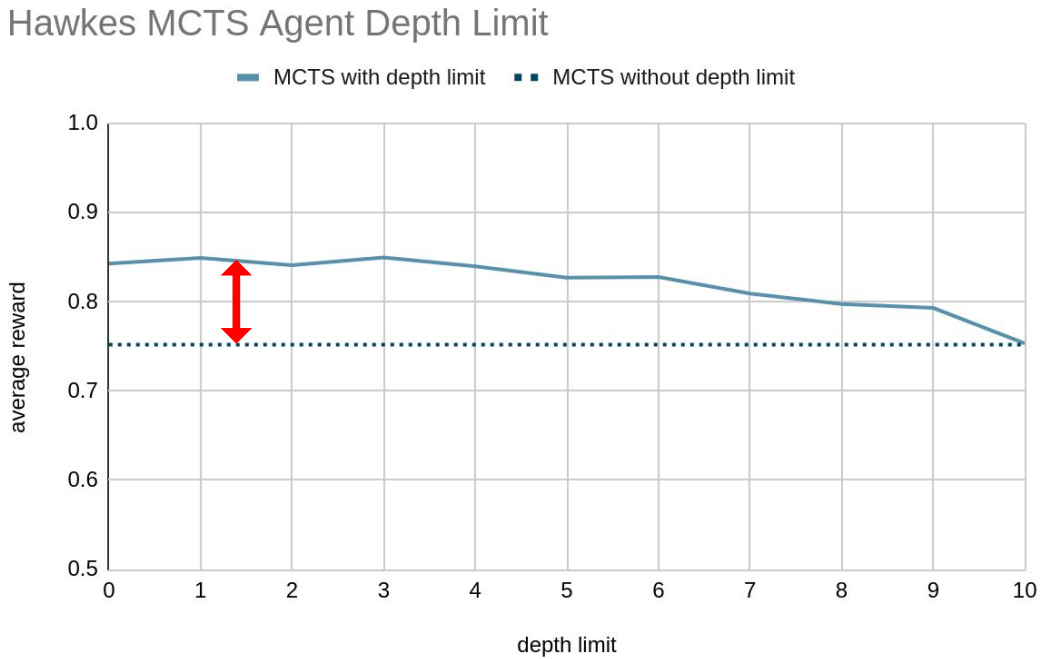
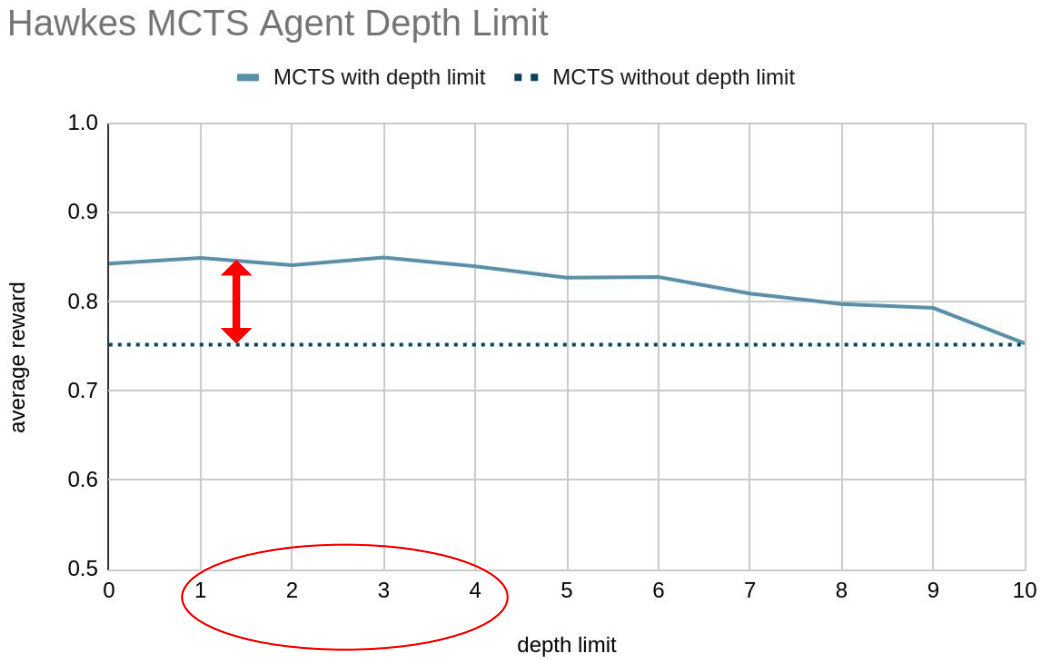



Image Source: freepik.com

# Monte Carlo Tree Search



# Monte Carlo Tree Search



# Monte Carlo Tree Search

Problem: Random simulation policy might not provide accurate evaluation

Solution: Introduce RL simulation policy - **Policy Gradients (PG)**

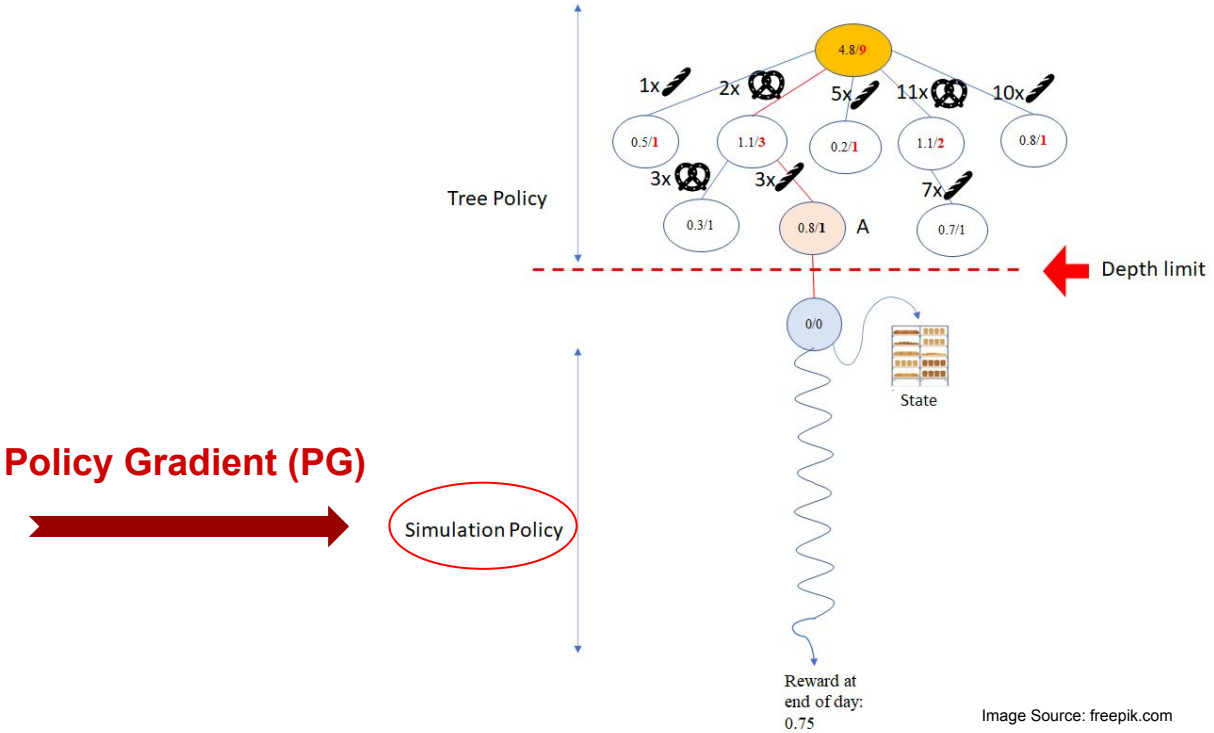



Image Source: freepik.com

PreciBake

# Policy Gradient Method

- **Policy function:** $a \sim \pi(a|s)$
  - Sample actions for given state
- Directly optimize **function approximator** for likelihood of a sample
- Push up probabilities of actions that lead to higher return
- Proximal Policy Optimization (**PPO**)
  - **Better convergence** by clipping gradients

# Final Experiments

# Experiment Setup - Agents

Threshold Agent

Monte Carlo Agent

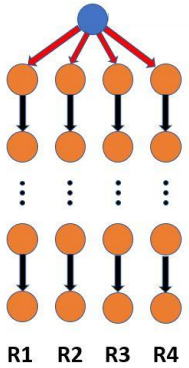


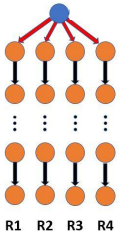

Image Source: freepik.com

PreciBake

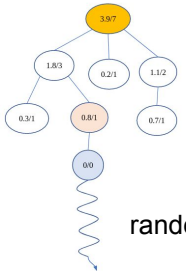
# Experiment Setup - Agents

## Threshold Agent

## Monte Carlo Agent

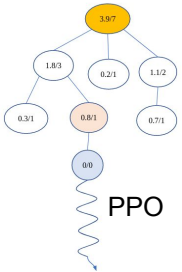



## Monte Carlo Tree Search Agent




random

## MCTS+PG Agent




PPO

Image Source: freepik.com

# Experiment Setup - Environment

Setting 1

Training

Testing

Setting 2

Training

Testing

Setting 3

Training

August

Testing

December

Image Source: freepik.com

# Experiment Setup

Setting 1 Setting 2 Setting 3

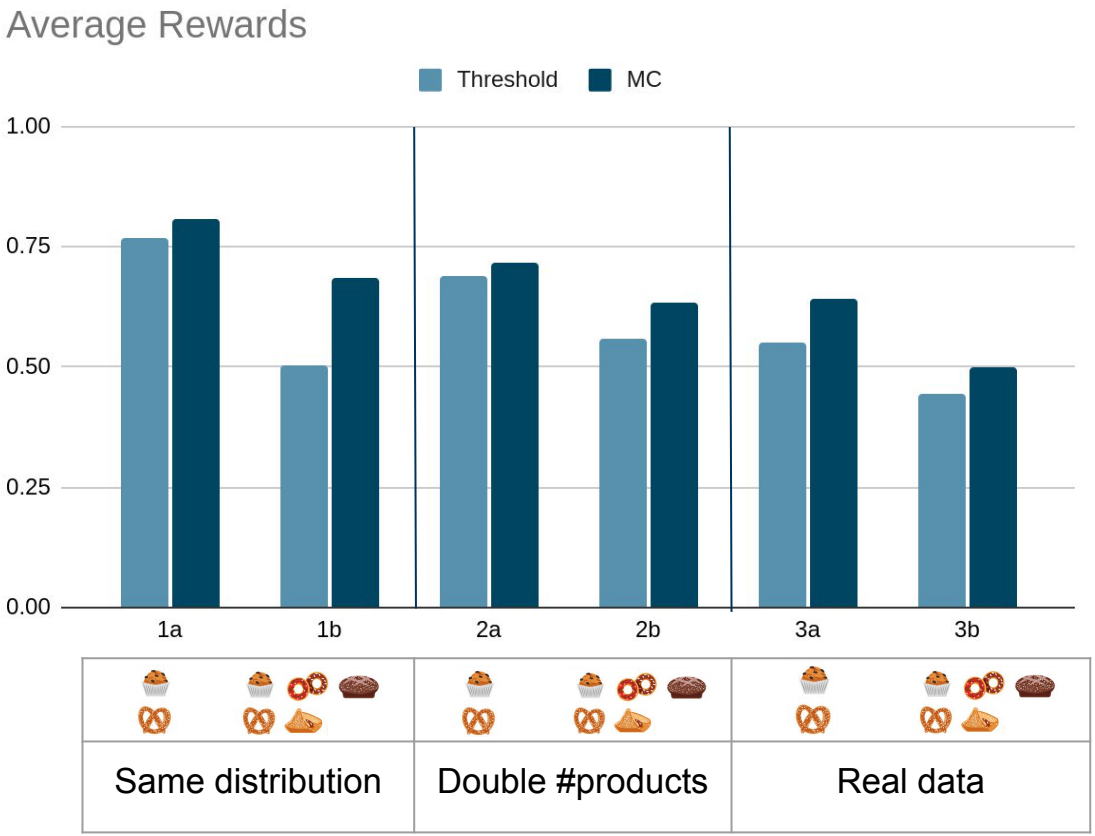
a

b

Image Source: freepik.com

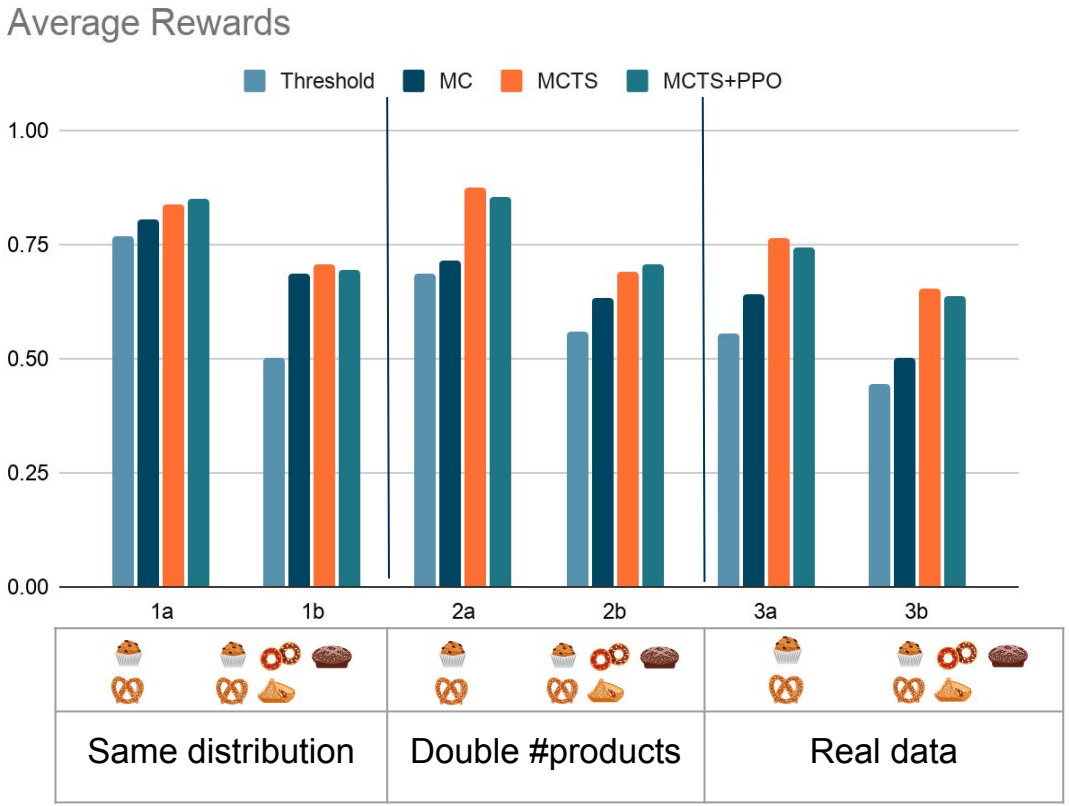
PreciBake

# Results



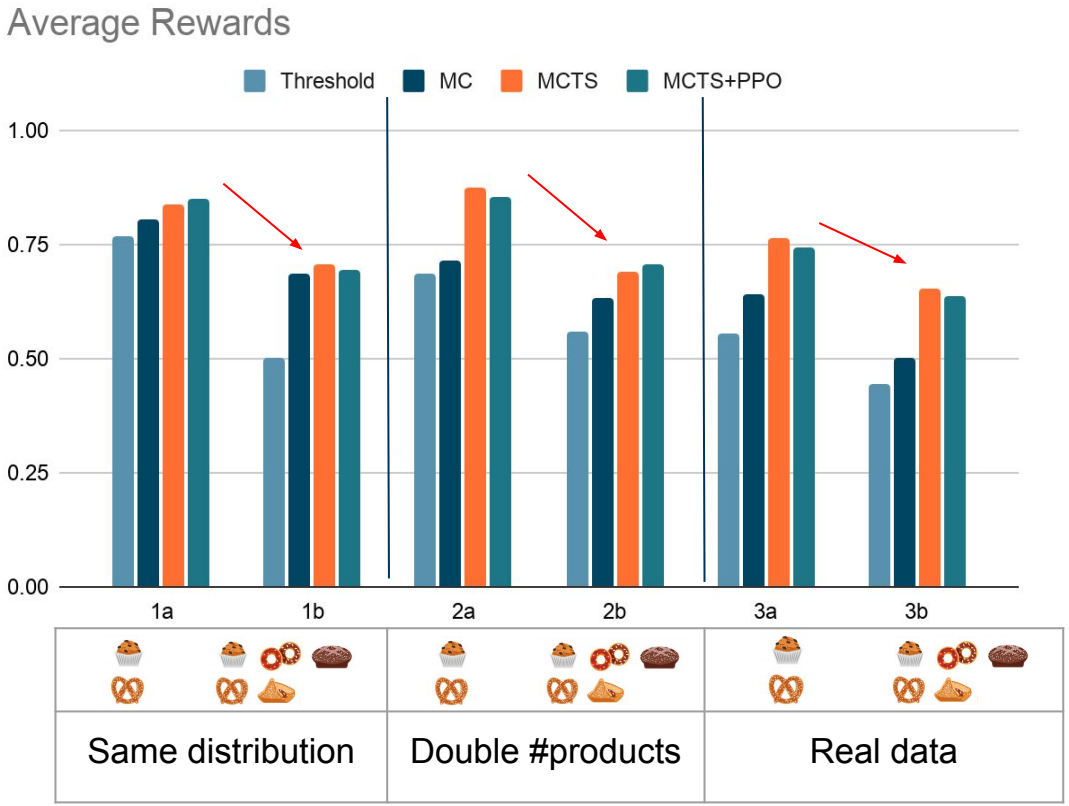

PreciBake

# Results



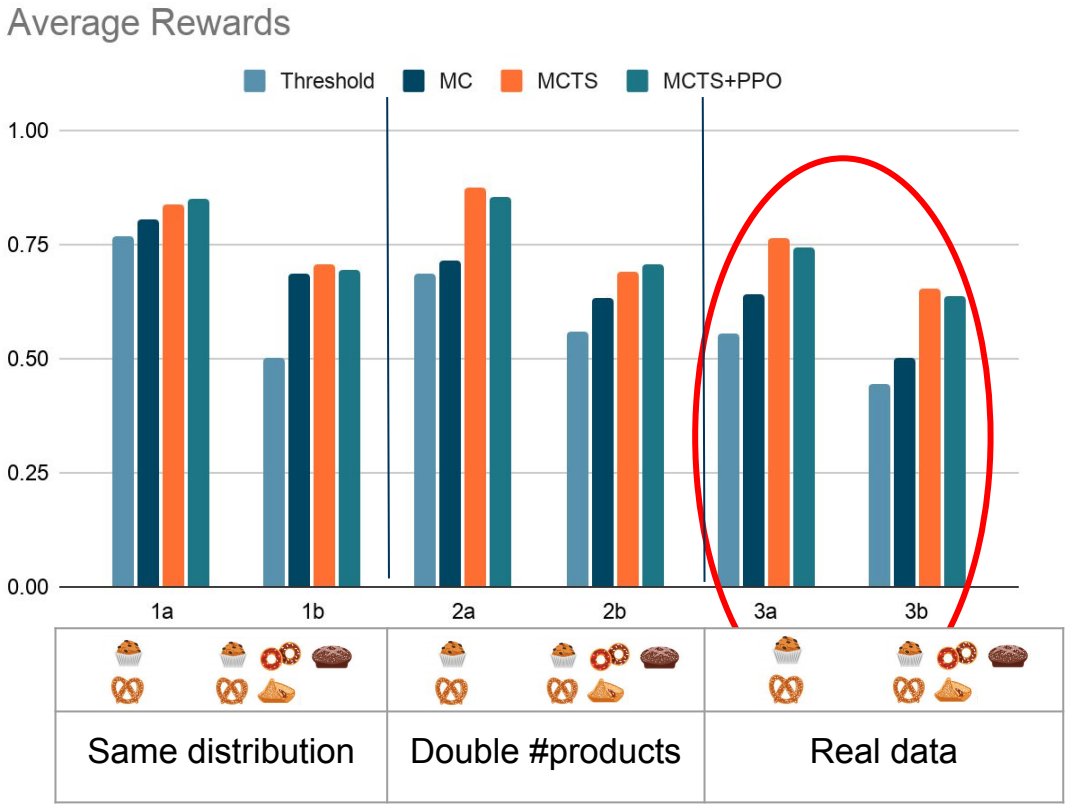

PreciBake

# Results



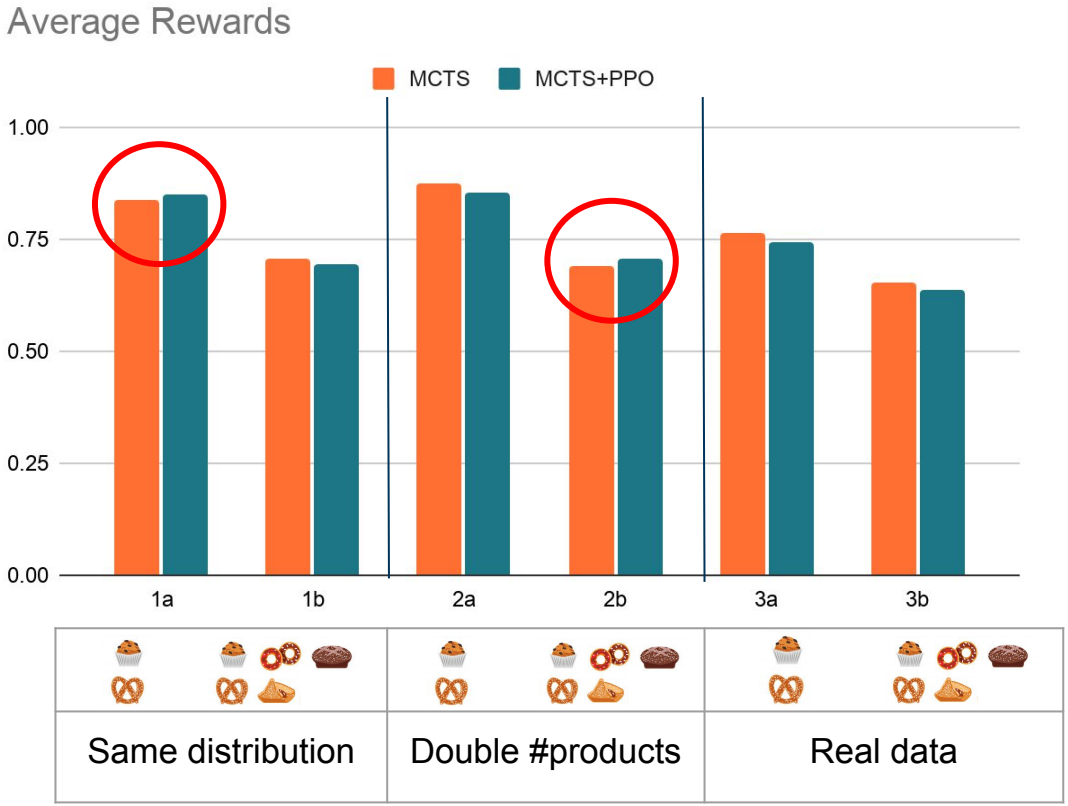

PreciBake

# Results




PreciBake

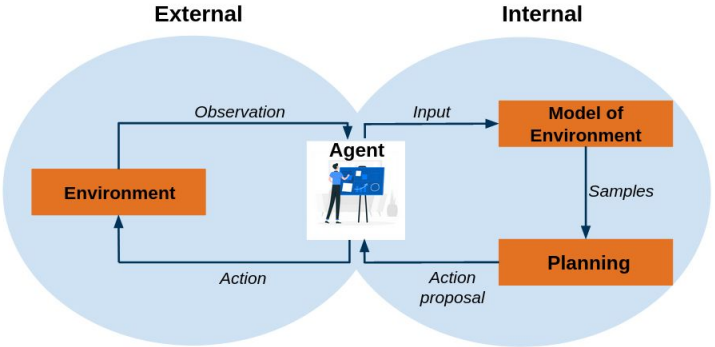
# Results




PreciBake

# Possible Future Improvements

- Improve **internal model**
- Expand usage of **RL** in MCTS
- Learn environment dynamics
- Model-based RL **without tree search**



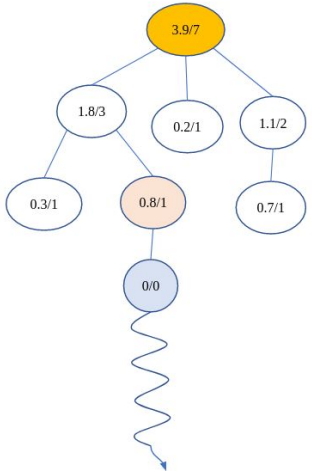



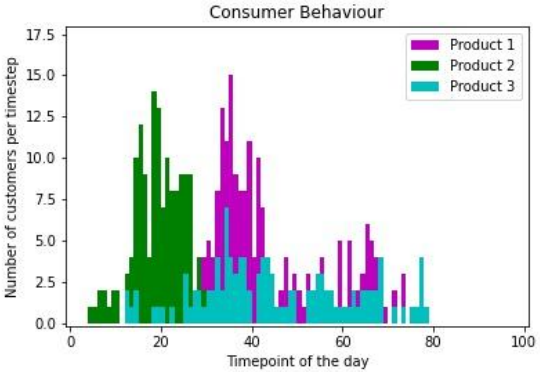

Image Source: freepik.com

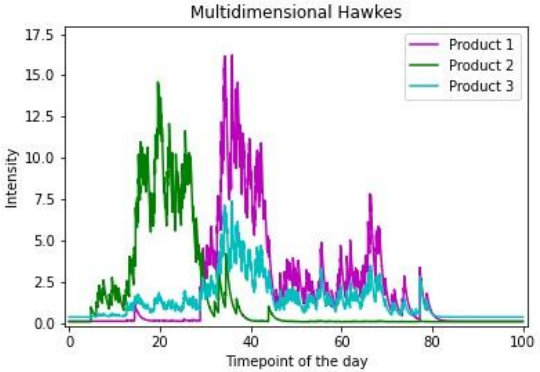
# Thank you for your attention!

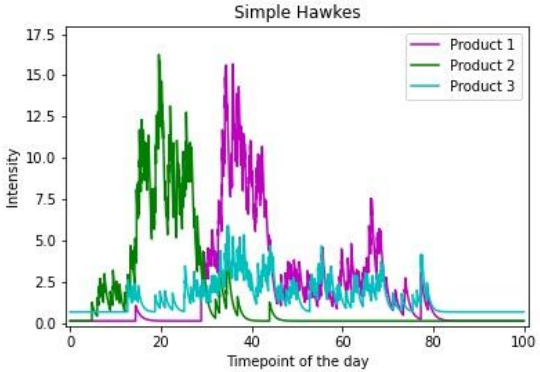
# Backup slides

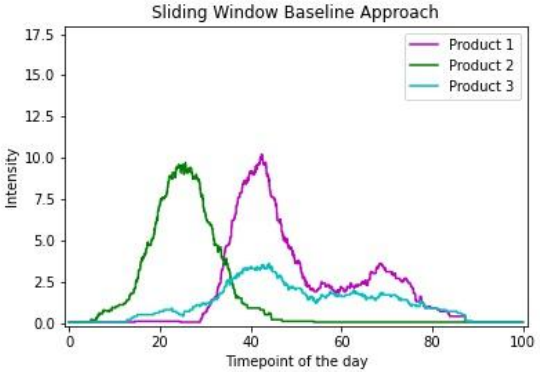
# Hawkes Experiments on Real Data

**Real consumer behaviour** and **Intensity functions** (example day, 3 products)



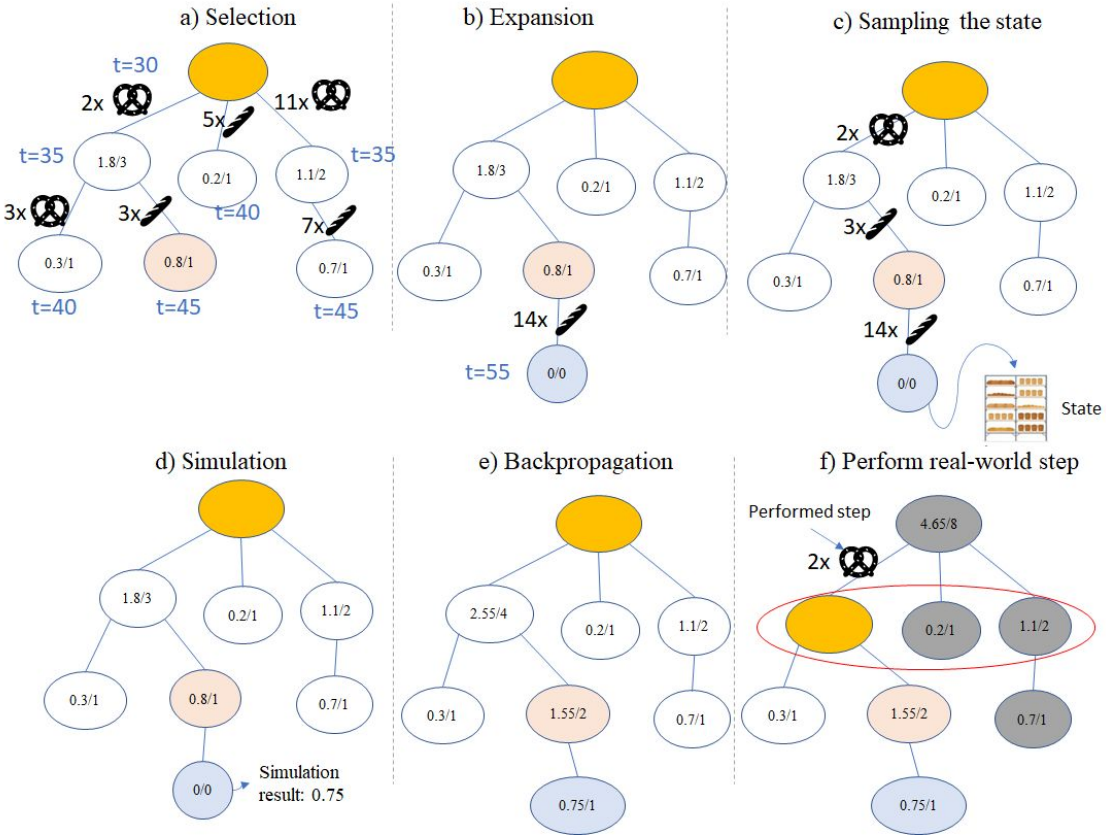




PreciBake

# Monte Carlo Tree Search



# Sampling from Hawkes

**Sampling via Thinning**

**Input**: time $t_i$, history $\mathcal{H}(t_i)$ and end of episode $t_{max}$
**Output**: tuple $(t_{i+1}, j)$ of next order time $t_{i+1}$ and product $j$ or *None*

1. **Set** $t = t_i$
2. **Choose** $\mu_0 \in \mathbb{R}^+$ such that $\mu_0 \geq \sum_{k=1}^{N} \lambda_k\big(t' | \mathcal{H}(t_i)\big) \quad \forall t' \geq t$
3. **Sample** inter-event time $\tau \sim \text{Exp}(1/\mu_0)$
4. **Set** $t = t + \tau$
5. **If** $t > t_{\text{max}}$:
   return *None*
6. **Sample** $j \sim \text{Categorical}\Big(\frac{\lambda_1(t|\mathcal{H}(t_i))}{\mu_0}, ..., \frac{\lambda_N(t|\mathcal{H}(t_i))}{\mu_0}, 1 - \frac{\sum_{k=1}^{N} \lambda_k(t|\mathcal{H}(t_i))}{\mu_0}\Big)$
   **If** $j \in \{1, ..., N\}$:
   return $(t, j)$
   **Else**:
   Go to 3.