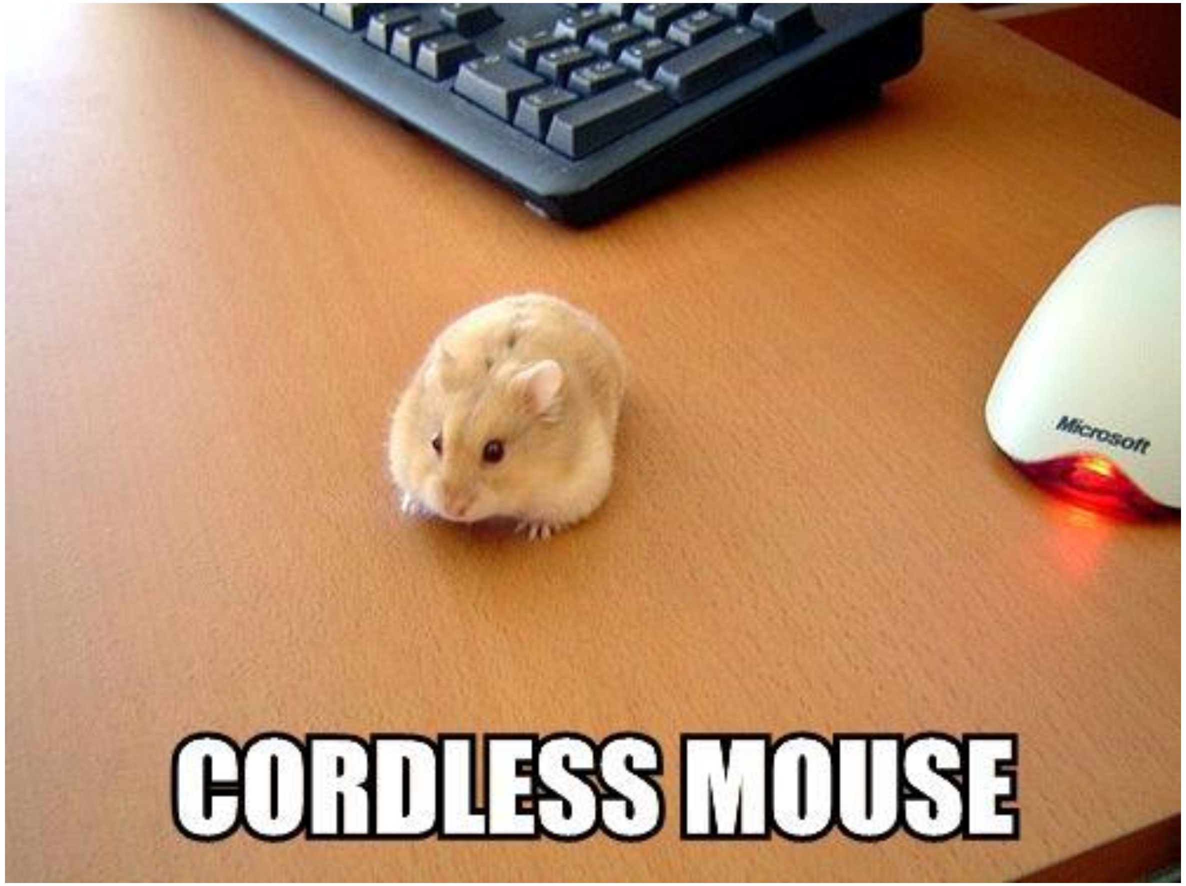
Explaining Animal Learning through Reinforcement Learning, Reward Parameterization, and Evolving World Models

Camila Blank

To begin...



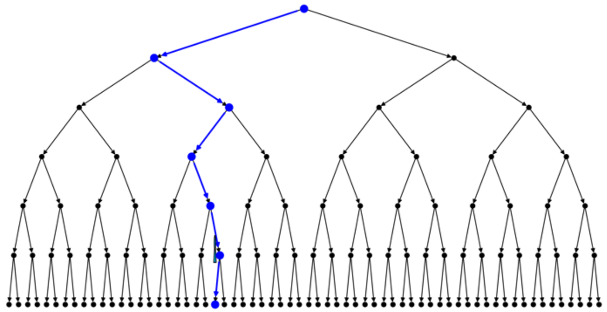
Motivation

- Gain insight on the neural processes underlying a mouse’s decision-making process in curiosity-driven navigation
- Combine reinforcement learning with multiple frameworks for intrinsic rewards
- Quantify contributions of extrinsic and intrinsic rewards, track an evolving world model, and observe effects on cohorts with stimulated neural circuits
- We focus on modeling the learning process itself rather than just learned behavior

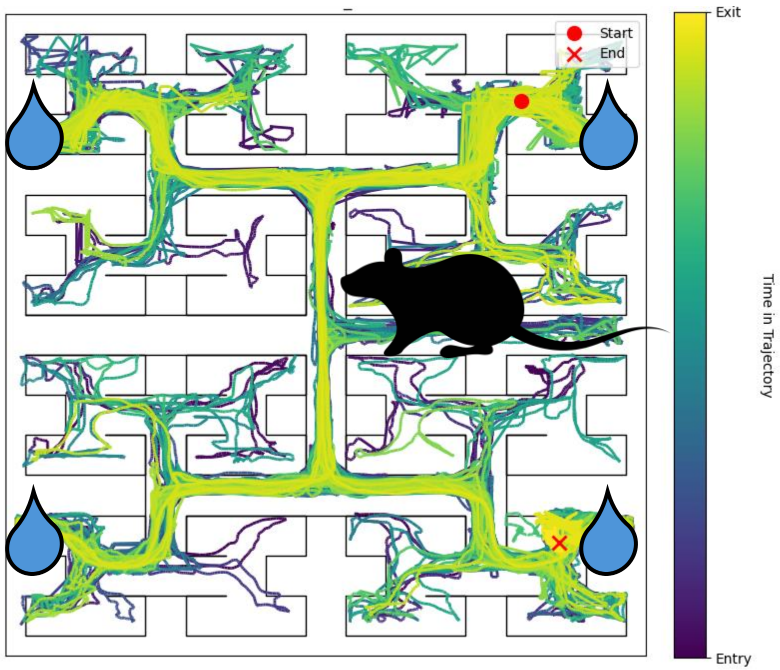
Rosenberg et. al. “Mice in a labyrinth show rapid learning, sudden insight, and efficient exploration”

- Mice in labyrinths make about 2000 decisions per hour
- There is an “underlying search algorithm” that primarily explained by local turning rules, not a global memory of the maze
- Many mice experience sudden improvements, implying moments of insight about their environment

Mouse Maze Dataset



- Water-starved mice
  - Excitatory: C21
  - Control: saline



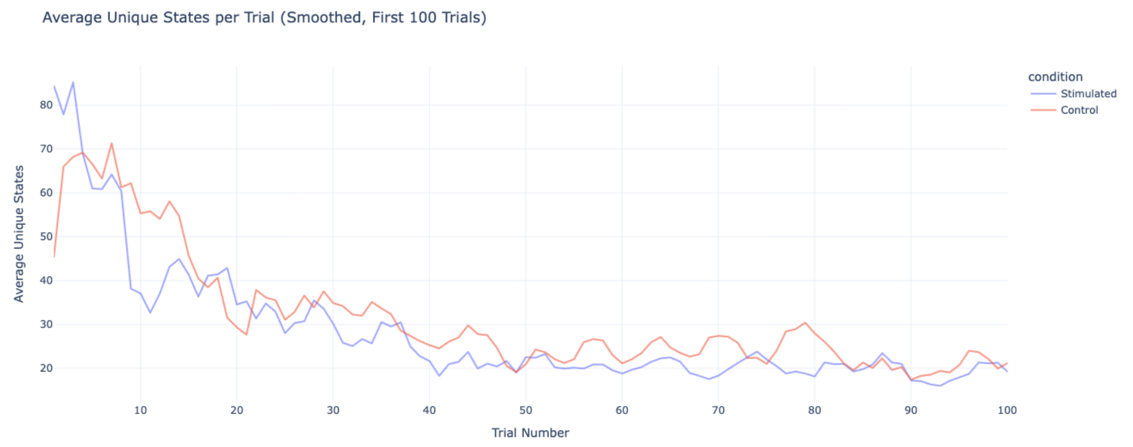
- Maze structure:
  - 127-node binary tree → 3 possible actions
  - Four randomly alternating water ports

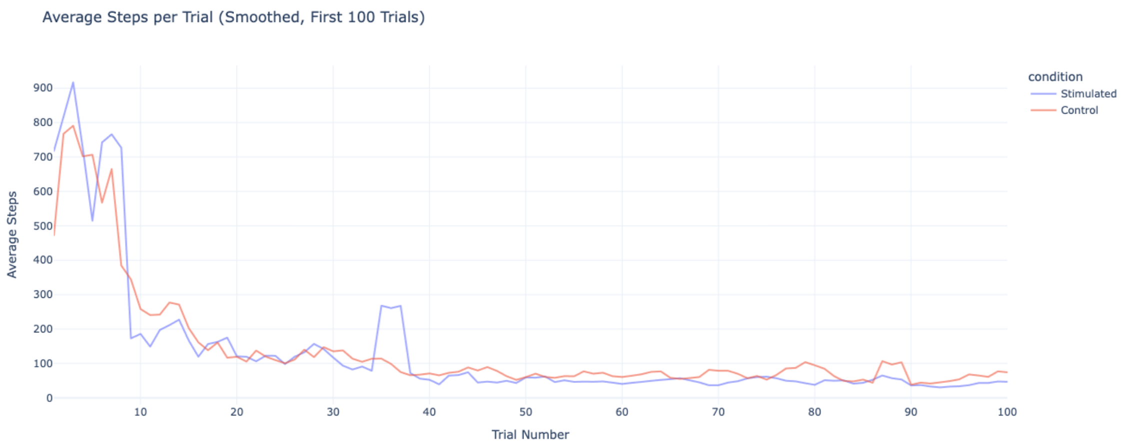
- Task structure:
  - 10 sessions (1/day)
  - 45 min each

Initial Analysis

- Number of steps to solve the maze converges quickly

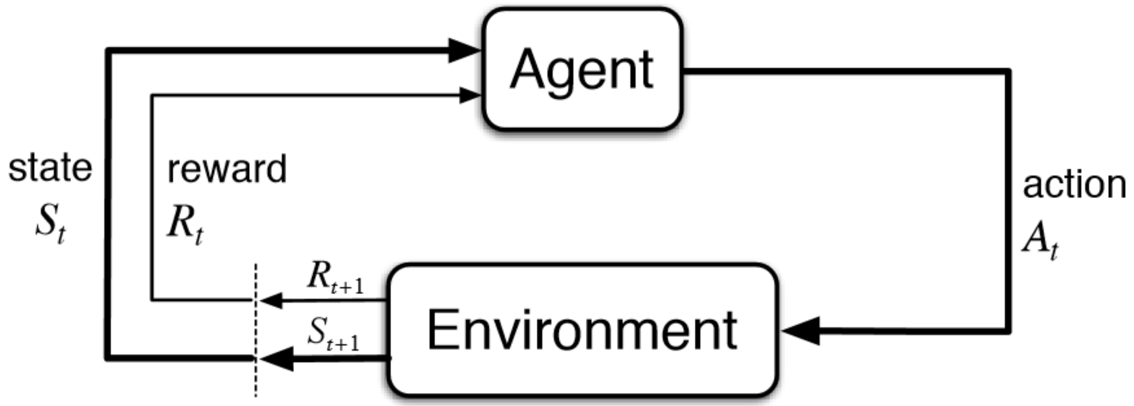
- Mouse learning largely happens within first 100 trials / 90 min





RL Basics: Markov Decision Processes

- Framework for sequential decision-making in unknown environments
- Next state is solely a function of the current state (Markov Property)
- Key components: state-action pairs, reward function, transition probabilities, discount factor



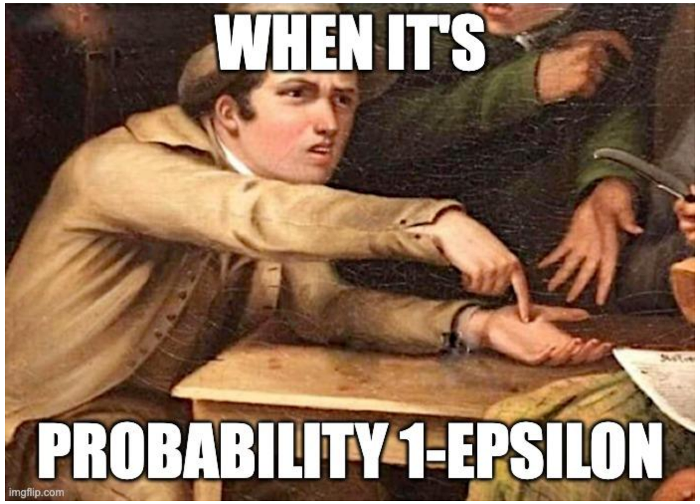
Standard algorithms

Q-learning (control):

- $Q(s,a) = Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s', \ a') - Q(s, \ a)\right)$ (for each goal)

Epsilon decay:

- Epsilon-greedy action selection
  - Explore with probability epsilon, exploit with probability 1-epsilon
- We start with a high epsilon and decay with every episode



Reward engineering

Uncertainty reward:

- Bayesian dynamics as world model
- Prior: $P(s'|s,a) \sim Dir(\alpha_1^{s,a}, \alpha_2^{s,a}, \dots, \alpha_{|S|}^{s,a})$
- $r_U^{t,k}(s,a,s') = \eta_U \cdot KL(P_{t,k}(s'|s,a) \parallel P_{t-1,k}(s'|s,a))$

Novelty reward:

- $r_N^{t,k}(s,a,s') = \eta_N \cdot \frac{1}{\sqrt{N(s')}}$

Combined:

- Total reward = uncertainty + novelty + extrinsic
- Epsilon decay

Details

- **Dirichlet distribution**
  - “Distribution of distributions” (dice factory)

$$f(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

- **KL-divergence**
  - Measure of how different two distributions are
  - Math: expected value of excess surprisal

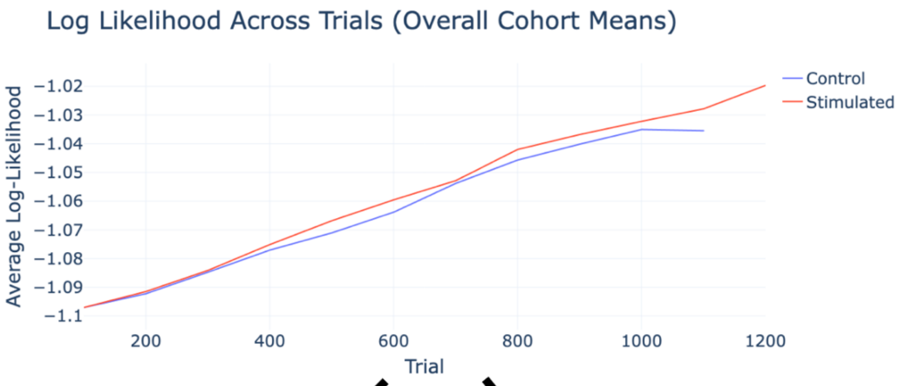
$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}.$$

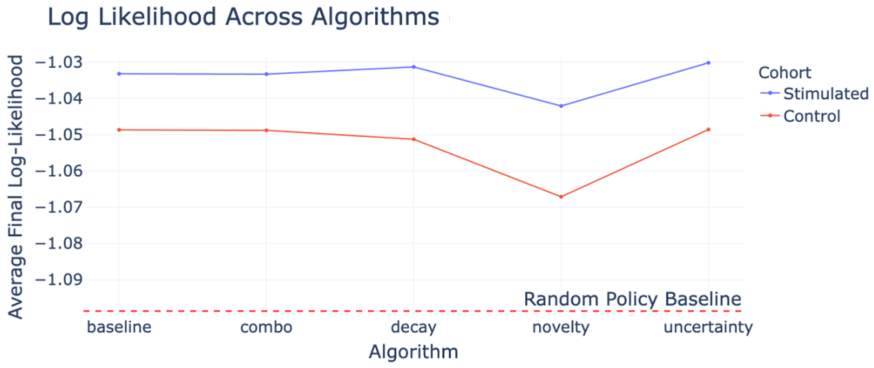
- **Switching reward nodes**
  - Q-table is num_states x num_actions x num_goals

Tuning hyperparameters via log-likelihood optimization

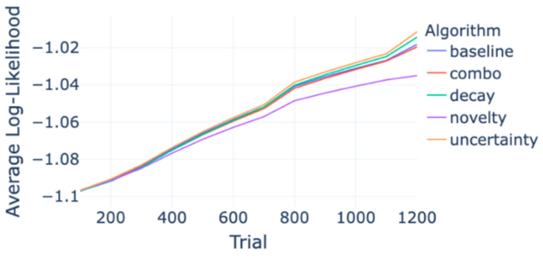
- Hyperparameters: $\eta_N, \eta_U, \gamma, \alpha, \varepsilon, \varepsilon$-decay
- Minimize: $loss = -\frac{\sum_{j=1}^{N} \sum_{i=1}^{T_j} log\pi_j(a_{ij}|s_{ij})}{\#\, total\ timesteps}$
- π_j = softmax policy for $Q\_list[j]$ frozen after trial j with $\beta = 1.0$

Uncertainty succeeds marginally

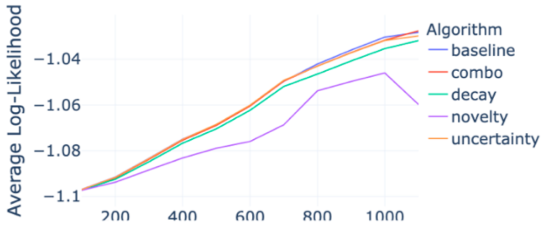




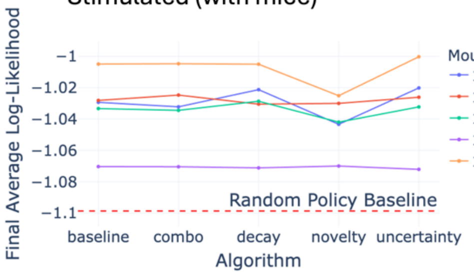
Stimulated (with algorithms)



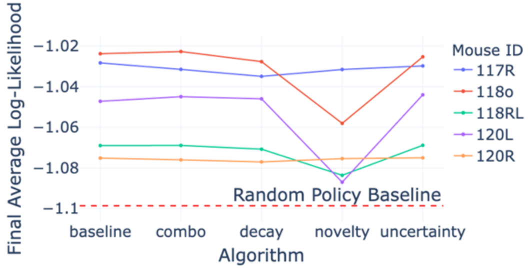
Control (with algorithms)



Stimulated (with mice)



Control (with mice)



Discussion

- Results suggest that reducing uncertainty may be a source of intrinsic reward in mice

- Generally, Q-learning algorithms more effectively predict stimulated mouse behavior

- Next step is inverse reinforcement learning → derive the reward parameterization from the ground truth data

Thank you!

(especially Aditi and the behavior modeling subgroup!)