Learning what is relevant for rewards via value learning and hypothesis testing

Mingyu Song[1*], Ming Bo Cai[1], Yael Niv[1,2]
[1]Princeton Neuroscience Institute and [2]Department of Psychology, Princeton University; *mingyus@princeton.edu

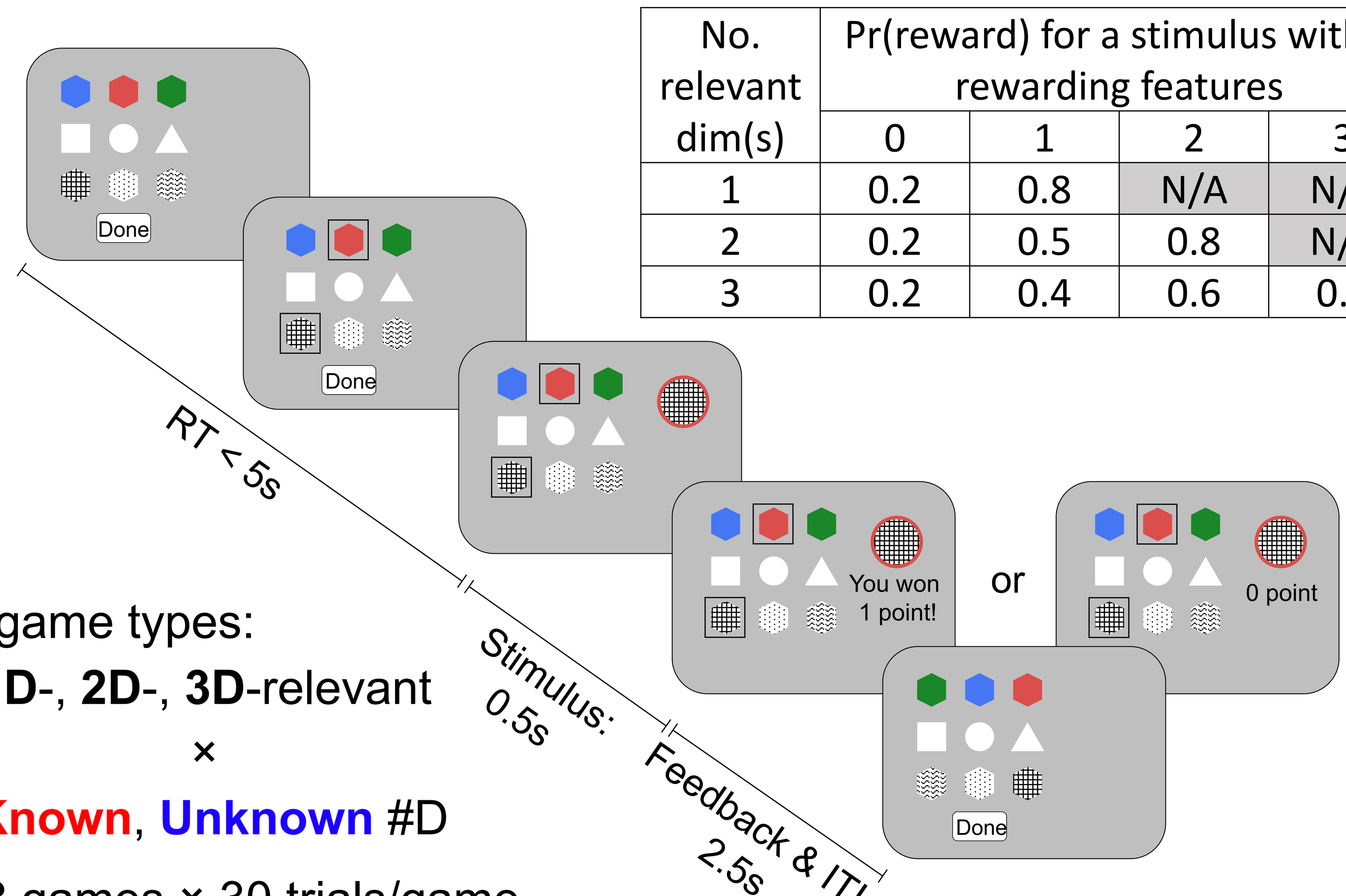
Research Question

How do people learn what is relevant for reward in a multi-dimensional environment, with probabilistic outcomes and multiple (or even unknown number of) relevant dimensions?

What makes a good coffee?
Brand? Origin? Roast level? Brewing method? …

The build-your-own-stimulus task

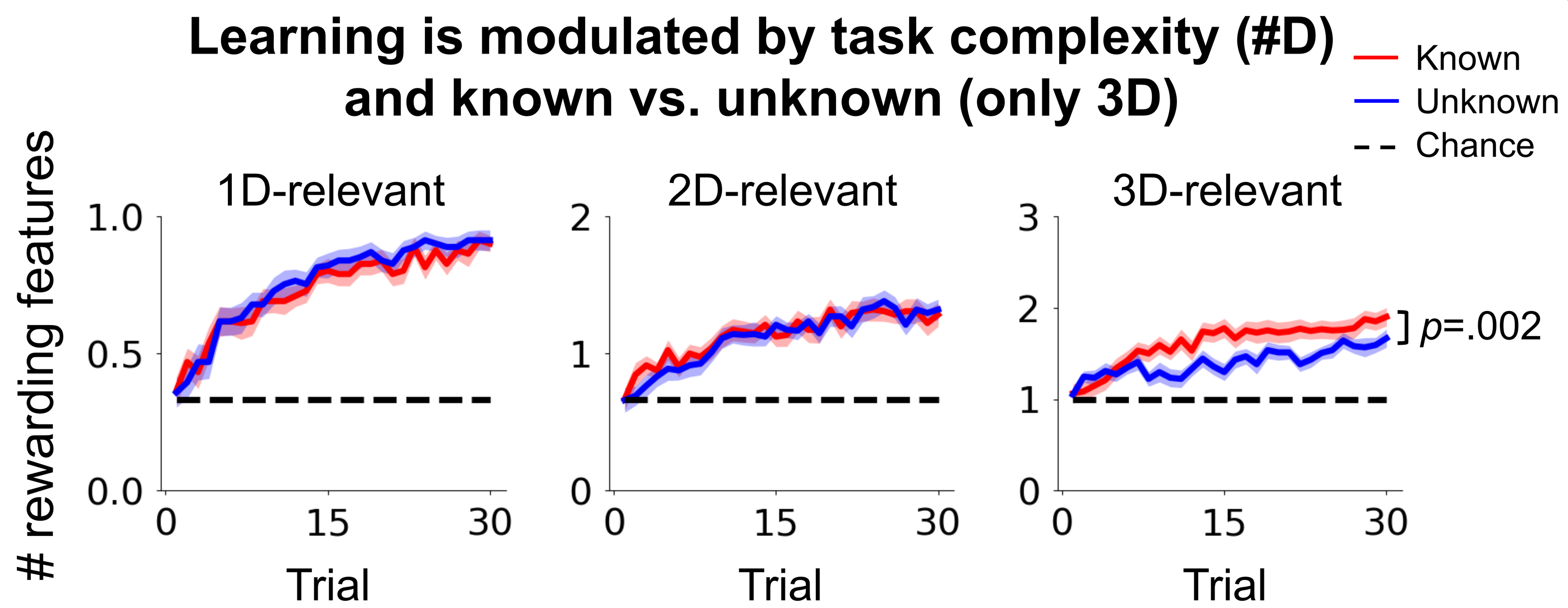
No. relevant dim(s)	Pr(reward) for a stimulus with ? rewarding features			
	0	1	2	3
1	0.2	0.8	N/A	N/A
2	0.2	0.5	0.8	N/A
3	0.2	0.4	0.6	0.8



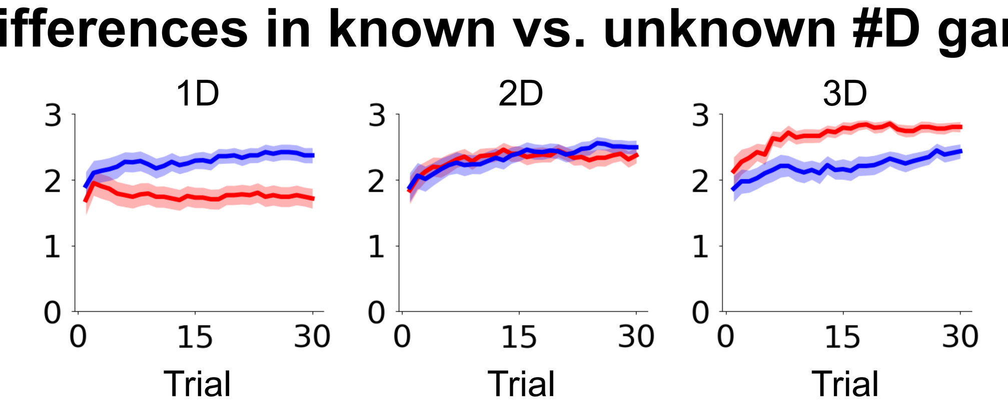
6 game types:
1D-, **2D-**, **3D-**relevant
×
Known, **Unknown** #D

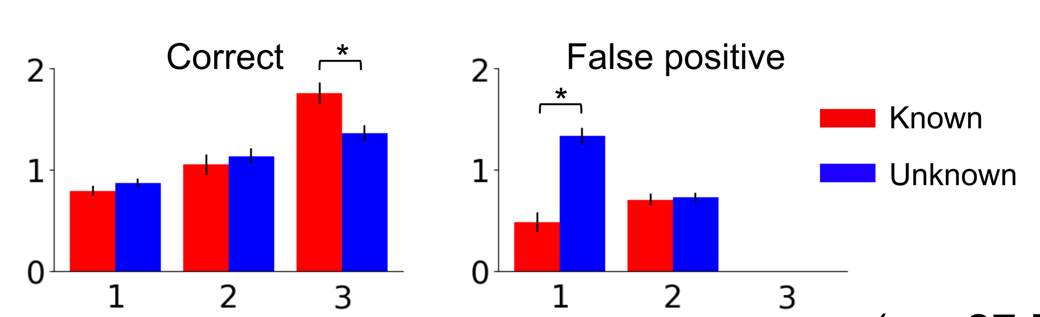
18 games × 30 trials/game

Learning is modulated by task complexity (#D) and known vs. unknown (only 3D)



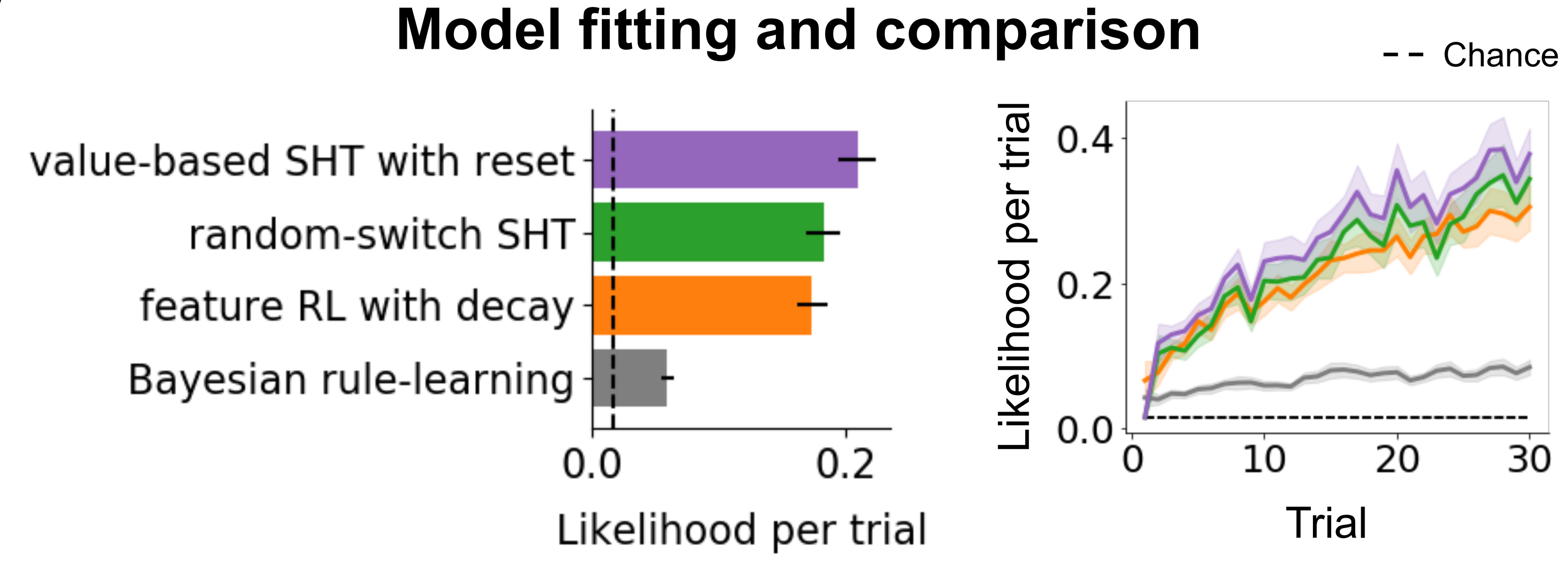
Strategy differences in known vs. unknown #D games



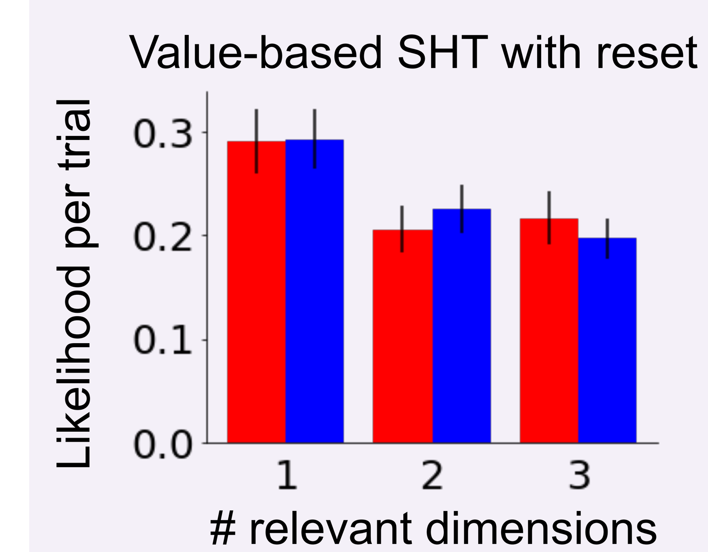


(n = 27 Mturk)

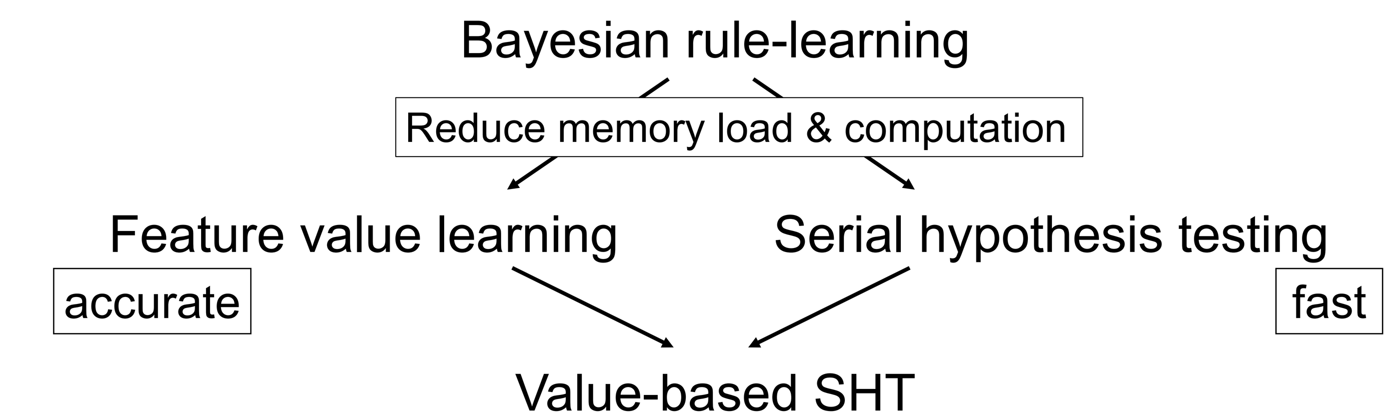
Model fitting and comparison

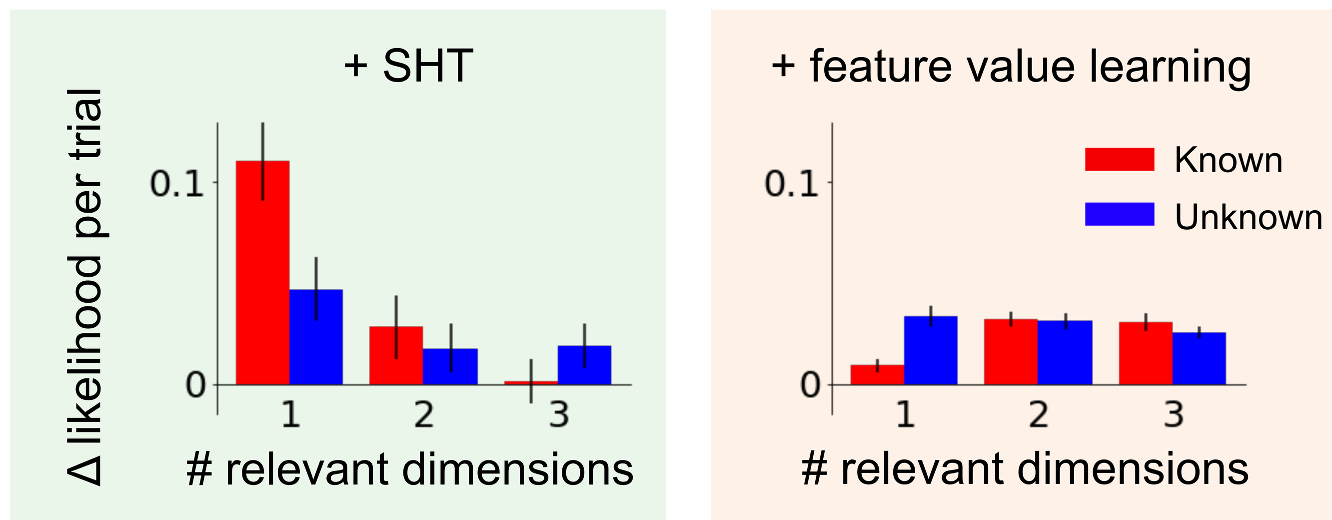


- Both feature RL with decay and serial hypothesis testing fit better than Bayesian model;
- Mixture model combining both strategies fits best;
- Easier games (1D) are fitted better than harder ones (2/3D).



Mixture of value learning and hypothesis testing; Strategy depends on task condition





Computational models

Choice policy (all models): softmax on the expected reward of choices, with additional costs associated with selecting features.

$$P(a) = \frac{e^{\beta\left(ER(a) - c\cdot\sum_i \delta_i(a)\right)}}{\sum_{a'} e^{\beta\left(ER(a') - c\cdot\sum_i \delta_i(a')\right)}}$$

(1) Bayesian rule-learning model

- Performs Bayesian inference over all possible hypotheses

$$P(h|a_{1:t}, r_{1:t}) \propto P(r_t|h, a_t)P(h|a_{1:t-1}, r_{1:t-1})$$

- Expected reward of choices: $ER(a) = \sum_h P(h)P(r|h,a)$

(2) Reinforcement learning model: feature RL with decay

- Learns 9 feature values with separate learning rates for selected features (η_s) or computer-generated (η_r)

$$V_t(f_{i,j}) = V_{t-1}(f_{i,j}) + \eta(r_t - ER(a_t))$$

- Expected reward as the sum of feature values

$$ER(a) = \sum_i V(f_{i,a^i})$$

- Values of features not in stimulus decay towards zero

$$V_t(f_{i,j}) = d \cdot V_{t-1}(f_{i,j}), \text{ if } j \neq s_t^i$$

(3) Serial hypothesis testing model: random-switch SHT

- Deciding whether to stay or switch: $Pr(\text{stay}) = \frac{1}{1+e^{-\beta_{\text{stay}}(P(r|h)-\theta)}}$
- If yes, randomly switch to another hypothesis

(4) Value-based SHT with reset

- Pr(stay) defined the same as random-switch SHT
- Also learns feature values (reset at hypothesis switch), used to determine which hypothesis to switch to.

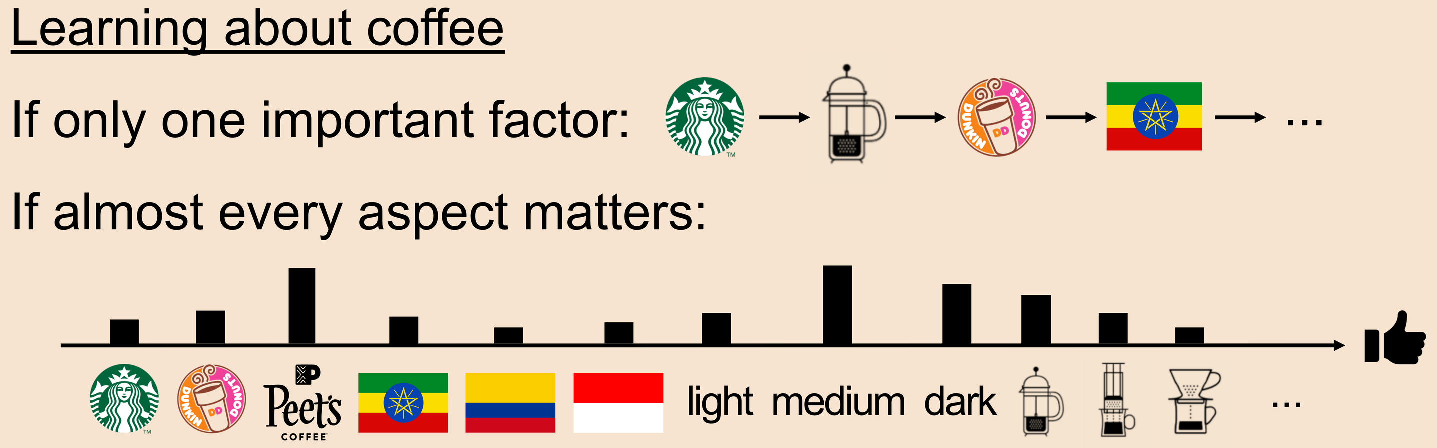
Conclusions

- Evidence for strategies involving feature-value learning and serial hypothesis testing.
- In known #D condition, people are sensitive to task complexity: serially testing hypotheses in 1D-relevant condition, and relying on feature value learning in 3D-relevant condition.
- In unknown #D condition, people use a mixed strategy.

Learning about coffee

If only one important factor:

If almost every aspect matters:



If no information is given: mixed strategy

Ongoing works

- Infer tested hypotheses (currently: choices = hypotheses)
- Test and compare different hypothesis-switch policies
  - Value-based: should feature values be reset?
  - Memory-based: cluster episodic memories?