Report 1

May 14, 2025

Contents

1 Research question and hypotheses

Does the brain keep track of the values associated with all possible policies, or only the currently exploited one? What drives our decision-making process as value computation becomes less tractable/more uncertain? We hypothesize that, as task complexity and uncertainty increase, the decision-making process goes from: value-comparison driven, to value-comparison-to-threshold driven, to "consequence" driven. By consequence, we mean the change in mean reward between trials ($\Delta\bar{R}$). A "consequence" driven agent seeks high reward states, but does not compute value in the traditional sense. To test the above hypotheses, we developed a novel perceptual decision-making task.

2 Task

2.1 Stimuli generation

The mean of the stimuli in Trial 1, m, has bounds:
$(|g| + \frac{\max d}{2}, 1 - |g| - \frac{\max d}{2})$
There are 10 possible values of m for each g. These values are linearly spaced between the lower and upper bounds and are sampled at random before each episode. These bounds ensure the minimum and maximum stimuli values are 0 and 1, respectively.

The stimuli heights are then computed as follows:
Trial 1 stimuli heights: $m \pm \frac{d}{2}$
Trial 2 stimuli heights: $\begin{cases} m \pm \frac{d}{2} + g \text{ if chose small} \\ m \pm \frac{d}{2} - g \text{ if chose big} \end{cases}$

Difficulty, d, determines the magnitude of the difference in stimuli heights in an episode. A single value of d $\in$ {0.05, 0.2, 0.35} is randomly chosen before each episode. d is constant within an episode.
Gain, g, determines the magnitude and sign of the consequence of the Trial 1 choice. Each block consists of 30 episodes and has a single value of g. In total there are 4 blocks, one for each value of g $\in$ {-0.3, 0, 0.1, 0.3}. The order of these blocks is random.

2.2 Determining Gain (g) and Difficulty (d)

Gain and difficulty are two of the most important task parameters. They determine the magnitude (and sign) of consequence as well as the magnitude of the difference between stimuli presented in a given trial. These parameters determine the optimal strategy. Proper selection of g and d should allow us to determine whether participants' decisions are primarily determined by 1) value comparison, 2) comparison to threshold, or 3) "consequence".

2.2.1 Original values

The previous g was 0.3 for Horizon 1. The previous d values were: [0.01, 0.05, 0.1, 0.15, 0.2].

These values of g and d meant that Small-Big was always the optimal action sequence/policy in Horizon 1. In other words, giving up 0.2 units of reward in trial 1 was always worth it since the mean reward of the stimuli would increase by 0.3.

2.2.2 Updated values

More values of g, along with appropriate values of d, are required in order to determine how humans decide in the consequential task. I propose four values of g:

Table 1: Values of gain (g) for Horizon 1 v2

Condition	G	g	π^*
A	$G < 0$	-0.3	Big-big
B	$G = 0$	0	Big-big
C	$G = c$	0.1	Small-big ~ Big-big
D	$G > c$	0.3	Small-big

I propose the following d values: [0.05, 0.2, 0.35]

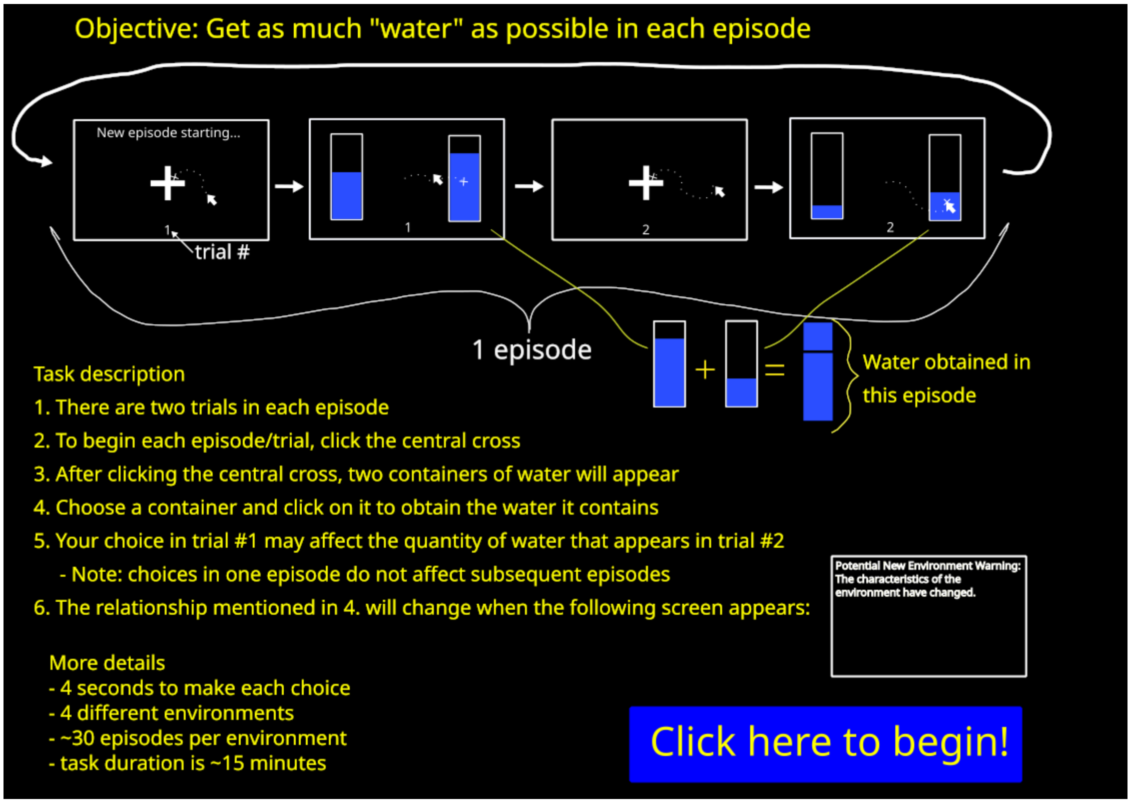
The data below is the output from a Python script in which I computed the cumulative reward for the two logically viable action sequences (small-big & big-big) and for all combinations of g and d:

```
g:  [-0.3, 0, 0.1, 0.3]
difficulty:  [0.05, 0.2, 0.35]


(cum reward small-big, cum reward big-big)
             0.05         0.2          0.35
-0.3  (0.7, 1.35)  (0.7, 1.5)  (0.7, 1.65)
0     (1.0, 1.05)  (1.0, 1.2)  (1.0, 1.35)
0.1   (1.1, 0.95)  (1.1, 1.1)  (1.1, 1.25)
0.3   (1.3, 0.75)  (1.3, 0.9)  (1.3, 1.05)

(cum reward small-big) - (sum reward big-big)
      0.05  0.2  0.35
-0.3 -0.65 -0.8 -0.95
0    -0.05 -0.2 -0.35
0.1   0.15  0.0 -0.15
0.3   0.55  0.4  0.25
```

The second "table" above shows shows the difference in cumulative reward, for all g and d, between the Small-Big and Big-Big action sequences. These values of g and d yield the optimal strategies outlined in Table 1. For these proposed values, the optimal policy is Big-Big for g=-0.3 and g=0. For g=0.1, Small-Big and Big-big yield identical cumulative reward. The optimal strategy, in this case, is Small-Big when d is large and Big-Big when d is small. Small-Big and Big-Big yield identical cumulative reward when d is 0.2 (the intermediate value). Finally, the optimal strategy is Small-Big when g=0.3.

2.3 Instructions for participants

Participants have as long as they want to read the instructions. They must click the "Click here to begin!" button at the bottom-right of the screen to begin the experiment.



2.4 Differences between Horizon 1 v1 & Horizon 1 v2

Table 2 shows most of the important differences between version 1 and version 2 of the task. There

Table 2: Consequential task differences (v1 vs. v2)

attribute	v1	v2
g	0.3	{-0.3, 0, 0.1, 0.3}
d	{0.01, 0.05, 0.1, 0.15, 0.2}	{0.05, 0.2, 0.35}
π*	Small-Big	g & d dependent
fixation timeout	skip trial	progresses trial
stimuli selection	mouse hover	mouse click

are many minor differences I have not mentioned here. In addition, please see the new task versions (table 4) which each are characterized by different levels of uncertainty.

2.5 Uncertainty in the Consequential task

2.5.1 Sources of uncertainty

Visual discrimination/perceptual uncertainty At least two kinds of uncertainty result from visual perception in the Consequential task.

1. For the smallest d, it can be difficult to determine which stimulus is larger.

2. It can be difficult to visualize and quantify the sum of the two chosen stimuli in an episode.
   - This makes value computation more difficult, which, consequently, makes value comparison between policies more difficult.

Lack of performance feedback The lack of performance feedback means participants never know if they are employing the optimal strategy.

Lack of knowledge regarding which aspects of the stimuli are important Participants don't know if the relative height of the stimuli is the only important attribute of the stimuli. Participants may expore other stimuli attributes such as position on screen (i.e., left/right) or order of presentation on the screen (i.e. first/second). It is also conceivable that participants may check whether g is a function of m, d, or reaction time.

2.5.2 Modulating uncertainty

Modulating uncertainty is important since one of our primary hypotheses is that the decision process shifts away from value-comparison as uncertainty increases. Below I propose two changes to the current Horizon 1 v2 task which would yield four versions of the task.

Table 3: 4 potential versions of the task with varying uncertainty

task version	uncertainty	g	value feedback
A	Low	constant	yes
B	Medium	stochastic	yes
C	Medium	constant	no
D	High	stochastic	no

Task version C corresponds to the current version. For the sake of feasability, I propose to run only versions A, C, and D of the task.

Stochastic g (increases uncertainty) Rather than g being held constant in each block, g could be sampled before each episode from a distribution with mean g. This would make value computation more difficult but, crucially, would not affect the optimal policy.

Value feedback (decreases uncertainty) In the present version of the task, the participants must visualize and quantify the sum of the two selected stimuli in their minds. They must also remember this value to then compare it with the approximated values of other policies. I propose making a version of the task with value feedback in which the sum of the selected stimuli is presented at the end of each episode along with a numerical representation of this sum. This would remove all uncertainty in value computation of the exploited policy.

How to set task parameters for each task version The different versions of the task can be run by changing two variables in the `"initialize_task_variables"` routine in the "params" code block.

Table 4: Python variables pertaining to uncertainty

task version	`stochastic_g_flag`	`value_feedback_flag`
A	False	True
B	True	True
C	False	False
D	True	False

2.6 Post-task survey

Participants will be redirected to a web page with a survey to fill out upon completing the task. The questions are as follows:

1. Gender
   - Man
   - Woman
   - Other/Prefer not to say
2. Age
   - 16-22
   - 23-27
   - 28-32
   - $>$32
3. Education (completed)
   - High School
   - Undergraduate
   - Masters
   - PhD/other advanced degrees
4. How well do you think you performed?
   - from 1 to 5
5. Did you understand the task instructions?
   - long text answer
6. Did you notice any differences between the different "environments"? If so, please describe them.
   - long text answer
7. What strategies did you try in order to maximize the water you received?
   - long text answer

8. How did you determine if a strategy was good?
    - long text answer
9. What strategy(ies), if any, did you decide were good? Did you use different strategies in the different "environments"? If you can remember, please describe which strategies you determined were good for each of the 4 environments?
10. How confident are you that you found good strategies?
    - from 1 to 5
11. If you can remember, how confident were you that you found a good strategy in each of the 4 environments? Please describe your confidence level for each environment individually.
12. (OPTIONAL) General feedback: all feedback is appreciated!
    - long text answer
13. (OPTIONAL) If you would like us to be able to share results with you, please provide your email address.
    - short text answer

2.7 Online deployment via Pavlovia

The task was made using the Builder of the PsychoPy desktop application. The task code and resources were then uploaded to a private GitLab repository and hosted online via the Pavlovia platform. The "low certainty" version of the task is available online and can be run by clicking on this link. The other versions of the task are fully implemented, I just have not finished getting them hosted on the Pavlovia platform at the time of writing.

2.8 Open questions regarding the task

1. Since we are primarily interested in value computation in the present project, should we try to eliminate sources of uncertainty that are less related to value computation? For example, should I include "Note: the only relevant attribute of the containers is the amount of water they contain. Other aspects of the containers (e.g., whether the container is on the left or right side of the screen) are irrelevant." in the instructions?
2. Should there be a monetary performance bonus for participants?
    - This would increase motivation. I could, at least, provide some performance-related feedback at the end of the experiment. I could also state in the instructions that they will receive a "score" at the end of the experiment. This may be a way of increasing motivation in lieu of a monetary bonus.
    - The nature of this bonus/feedback is important since participants will likely be more explorative if they only care about finding the optimal policy. If, however, participants know there is a monetary bonus or score proportional to the total amount of reward/"water" acquired in the experiment, then they may be more likely to continue exploiting suboptimal strategies if their associated values are above a certain threshold (i.e., foraging).

3. The current version of the task (task version C) takes roughly 15 minutes to complete. Should I employ a repeated measures design (i.e. every participant performs all versions of the task), or should I use an independent measures experimental design?
   - I'm leaning towards independent measures. I think it's more likely that participants will pay full attention and perform if the experiment only last 15 minutes. We can add a repeated measures group later if desired.
   - One argument in favor of a repeated measures design is seeing how exposure to a a low uncertainty version of the task (e.g. with value feedback) may affect performance in subsequent versions of the task (e.g. without value feedback).
4. In the "low uncertainty" version of the task, should I display the stimuli reward values prior to selection, or should I only show the reward values and sum of the selected stimuli post-choice? If I show the reward stimuli values pre-choice, this would eliminate all perceptual uncertainty.
5. Should I make different instructions for each version of the task?

3 Cognitive models

I propose to investigate two types of cognitive models (i.e., agents): value-comparison driven & consequence driven. I propose to implement to versions of each type of model: option comparison & comparison to threshold.

3.1 Value comparison

For the value-comparison case, I propose to use a classical model-free Q-learning algorithm with the following q-table: The update rule would be:

Table 5: q-table for value comparison agents

state	action: small	action: big
trial 1, small d	q1	q2
trial 1, large d	q3	q4
trial 2, large $\bar{R}$	q5	q6
trial 2, small $\bar{R}$	q7	q8

$$Q(S, A) \leftarrow Q(S, A) + \alpha(R + \gamma \max_a Q(S', A') - Q(S, A)) \tag{1}$$

The input to the choice rule for a given state will be the q values associated with that state (e.g., decisions in the "trial 1, small d" state will be determined by q1 and q2).

3.2 Value comparison to threshold

The value comparison to threshold agent's actions are driven by the comparison of the state-action values of the currently exploited strategy to a threshold, sometimes referred to as a "satisfaction" threshold. This decision-making strategy is compelling since we often settle for satisfactory action sequences rather than searching for truly optimal ones. The q-table for this agent is the same as in the value comparison case. The update rule is also identical. We will choice rule similar to the value comparison agent, however, the input to the choice rule function will be the q-value associated with the currently expoited strategy, and a "satistfaction" threshold, ρ. Decisions in the "trial 1, small d" state will be determined by the q-value corresponding to the currently exploited strategy (i.e., the q-value corresponding to the action taken in the previous episode) and ρ.

3.3 Consequence comparison

The consequence-driven agent does not compute value. Instead, this agent seeks high-reward ($\Delta\bar{R}$) states. The states and actions are the same as the previous cases, the "q-table", however is updated differently. Since q is typically used to refer to value, I will call the updated values consequence values, or c-values. C-values are updated based on the reward obtained in the current state as well as the change in mean reward between the current and subsequent states. The update rule is the following:

$$C(S,A) \leftarrow C(S,A) + \alpha(R + \gamma(\bar{R}' - \bar{R}) - C(S,A)) \tag{2}$$

Is is also conceivable that paricipants focus entirely on $\Delta\bar{R}$, and ignore reward acquired in the current state. For this reason, I propose to introduce one addition tunable parameter:

$$C(S,A) \leftarrow C(S,A) + \alpha(\beta_1 R + \beta_2(\bar{R}' - \bar{R}) - C(S,A)) \tag{3}$$

3.4 Consequence comparison to threshold

The consequence analog of value comparison to threshold.

3.5 Drift diffusion model choice rule

We will use a drift diffusion model as the choice rule for all agents. This will enable us to fit participants' reaction times (RT). The input to the DDM is different for each model, but the number of tunable parameters is the same, thus the addition of the DDM does not affect the complexity of the models relative to one another. To begin, I propose to use a constant scaler for the drift rate, v, constant diffusion boundaries, and a non-biased starting point, z. This reduces the complexity of the model. We may choose to fit these parameters if the simple version of the model results in a poor fit.

3.6 Model fitting & cross validation

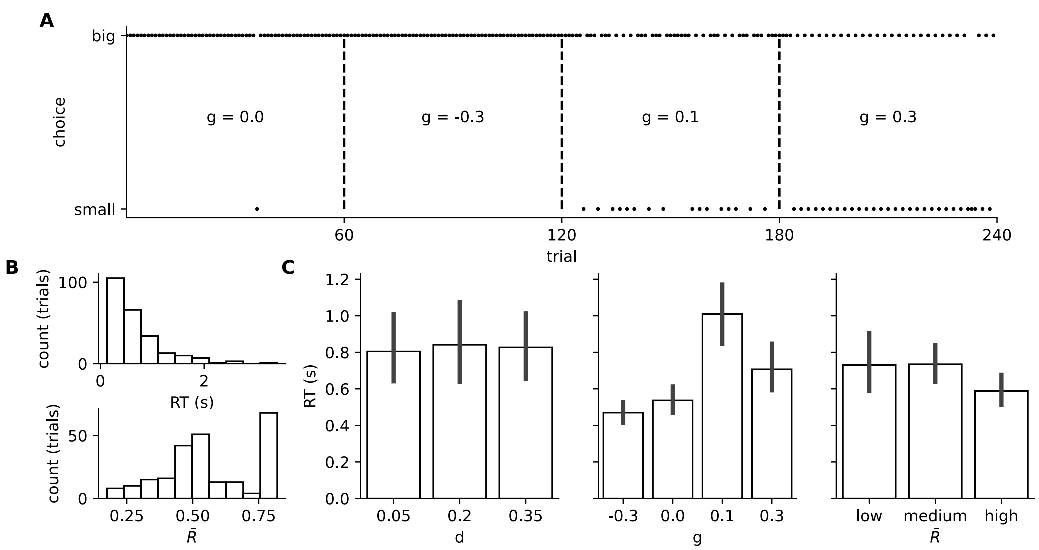
I propose to fit the reinforcement learning parameters and the DDM parameters simultaneously via hierarchical Bayesian parameter estimation. Hierarchical Bayesian parameter estimation is convenient in that it provides both individual and group-level parameters. After model fitting, I propose to use leave-one-out cross validation to determine the goodness-of-fit of each cognitive model.

4 Brief background

Many algorithms have been shown to reliably reproduce human decisions in particular contexts. The question of how to determine which of these algorithms best represent the true underlying cognitive function of the brain remains open. One common approach is to fit the tunable parameters of reinforcement learning models to maximize the likelihood of producing the observed decision data. Value-comparison based RL models have been shown to reproduce human behavior in various contexts. These models have also been combined with drift diffusion models to fit not only participants' actions, but reaction times as well[2]. Though value-comparison based models have traditionally been favored in most contexts, recent findings have shown comparison-to-threshold based models exhibit greater goodness-of-fit in classical reinforcement learning tasks irrespective of uncertainty level[3]. This finding, though fascinating, does not paint a complete picture of decision making under uncertainty. I believe humans rely on value-comparison in situations where value-computation is cheap, and then resort to comparison-to-threshold once the cost of value-computation reaches a certain level. Moreover, it seems logical that as this cost rises we should increasingly rely on heuristics since value-computation is intractable. One such heuristic is "consequence". Based on our previous research, I suspect participants may, in fact, act as $\bar{R}$ (i.e., mean reward of a particular state) maximizing agents when task uncertainty reaches a certain threshold[1]. Two key differences between the Dynamic Consequential Task and the Restless Bandit Task are: the presence of perceptual uncertainty in our task, and the fact that agent actions in our task have a predictable effect on the environment.

5 Preliminary results

I performed the task myself (version C). I confirmed the Psychopy output contained all the necessary data to perform the desired analyses.



6 Citations

References

[1] Gloria Cecchini et al. *Consequence Assessment and Behavioral Patterns of Inhibition in Decision-Making: Modelling Its Underlying Mechanisms*. Preprint. In Review, Aug. 2023. DOI: `10.21203/rs.3.rs-3241114/v1`. (Visited on 09/04/2023).

[2] Laura Fontanesi et al. "A Reinforcement Learning Diffusion Decision Model for Value-Based Decisions". In: *Psychonomic Bulletin & Review* 26.4 (Aug. 2019), pp. 1099–1121. ISSN: 1069-9384, 1531-5320. DOI: 10.3758/s13423-018-1554-2. (Visited on 02/17/2025).

[3] Meriam Zid et al. *Humans Forage for Reward in Reinforcement Learning Tasks*. July 2024. DOI: 10.1101/2024.07.08.602539. (Visited on 01/17/2025).