**A. How to use this folder**

Files/folders you needn’t interact with

1. This *valence\_README*.
2. *input\_data* folder, which contains files that are used by the psychopy files.
3. A *results* folder, which will be created when you run your first task. **Please actively back up the results folder. Files will not be overwritten, but it is safer to have a copy.**

Files you will interact with

1. *expt\_day\_checklist* which you should follow on the day of the experiment.
2. *AC\_slides* and *BD\_slides*, which, should be shown to the patient depending on their session type: (A or C) or (B or D). More below.
3. *AC\_script* and *BC\_script* which should be read to the patient as they refer to the slides, depending on their session type. The framing of this script is highly important.
4. *session\_type\_checklist*, whichtracks which session type was used for which subject on which date. I have filled out the session type to use for the first 20 patients; do continue the cycle when needed. You will be prompted to enter the session type (A, B, C or D) when running the practice & experiment files. This variable ensures counterbalancing of block order and shape-valence association across subjects. Read more in Section E below.
5. *valence\_practice.psyexp* practice file, which you should run after instructions.
6. *valence\_final.psyexp* actual experiment file.

**B. Description of task, valences, and outcomes**

The overall task is to determine if a shape stimulus falls on the left or right side of a continuous spectrum that varies along a particular dimension of shape-ness (curviness to flatness). While ostensibly the task is binary (left vs right), subjects are given an end-of-block bonus if their answers have been close to the correct exact position of the shape (and not just on the correct side). Subjects can use the L or R keys to move their slider to the desired location on the spectrum, and then hit the ‘space’ key to submit.

Each side of the spectrum is associated with a net-penalizing or net-rewarding contingency. For example, let’s say the *curvier* shapes are **net-penalizing** and​​*flatter* shapes are **net-rewarding** for session x.

* This means that if a shape stimulus is *curvier* (**net-penalizing**),
  + placing it **incorrectly** on the *flatter* side of the spectrum offers **high penalty** **(-2 coins)**, while
  + placing it correctly on the *curvier* side of the spectrum offers low reward (+1 coin).
* Conversely, if a shape stimulus is *flatter* (**net-rewarding**),
  + placing it **correctly** on the *flatter* side of the spectrum offers **high reward** **(+2 coins)**, while
  + placing it incorrectly on the *curvier* side of the spectrum offers low penalty (-1 coin).

Table 1: summarizing valence & outcome possibilities

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Curvy stimulus (net-penalizing)** | **Flat stimulus (net-rewarding)** |  |
| Placed on curvy side | 1 | -1 | Expected Value of choice = 0 |
| Placed on flat side | -3 | 3 | Expected Value of choice = 0 |
|  | Expected Value of stimulus = **-1** | Expected Value of stimulus = **1** |  |

Notes:

* Either placement choice has equal expected value (=0), precluding choice bias.
* Stimulus-valence association (i.e., whether curviness indicates penalty or reward context) will be counterbalanced across sessions.

Table 2: refactoring through the lens of classical Valence-Arousal emotion space

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Net-negative Valence** | **Net-positive Valence** |  |
| Safe choice  (Low Arousal) | 1 | -1 | Expected std. deviation = 1 |
| Risky choice  (High Arousal) | -3 | 3 | Expected std. deviation = 3 |
|  | Expected Value of stimulus = **-1** | Expected Value of stimulus = **1** |  |

**C. Trial schematicA diagram of a diagram of a task

AI-generated content may be incorrect.**

**Notes**

* We counterbalance whether the penalizing (e.g., curvy) shape is on the left or right of the screen, to preclude motor preparation. Read more in section E.
* Subjects use L & R keys to place their answer on the spectrum, and then hit the up arrow (U) key to submit, *using the same hand*.
* If subjects don’t hit U within 3s, or if they don’t hit the L or R key at all, they are shown the message ‘Respond faster!’, or ‘You must move the slider!’, respectively.

**D. Valence space modulation across blocks**

One of our key manipulations is increasing or decreasing the space of rewarding and penalizing stimuli. We achieve this by changing what we mean by the ‘left’ or ‘right’ side of the spectrum, thus changing which trials are rewarded or penalized.

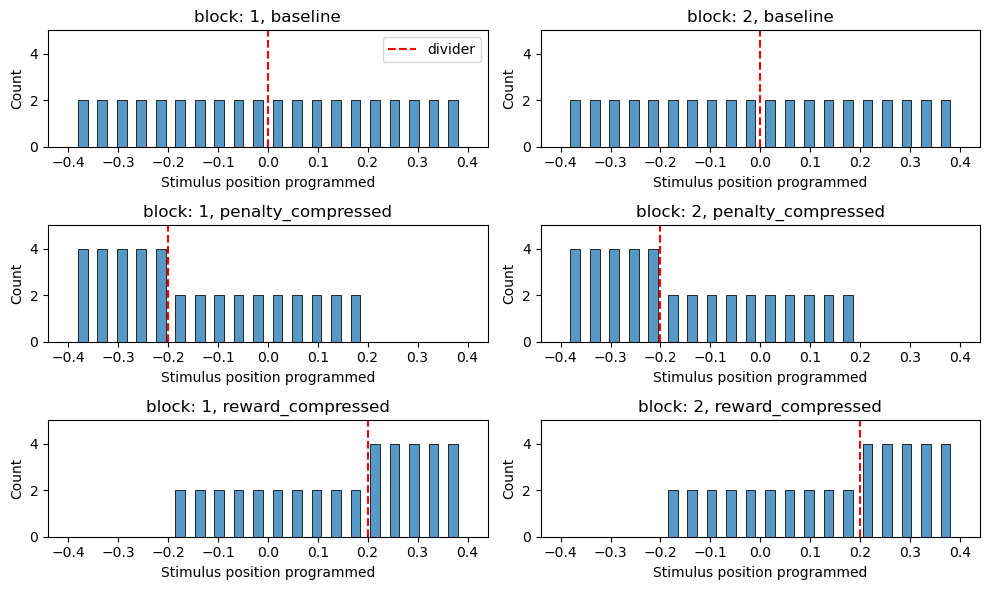
Specifically, in the baseline (‘No Compression’) condition block described above, the continuous spectrum is divided into 2 equal halves by a vertical line. Conversely, in the ‘Penalty Compressed’ block, the continuous spectrum is divided such that the vertical divider appears closer to the penalizing side, thus compressing the space of stimuli that are net-penalizing. Importantly, stimuli are still sampled equally from either side of the divider, precluding choice bias based on stimuli frequencies. Additionally, there is a ‘Reward Compressed’ block, which compresses the space of stimuli that are net-rewarding. See schematics in section F for reference.

**E. Stimulus counterbalancing**

* Each compression type (baseline, penalty, reward) appears in 2 blocks.
* Each of the 6 blocks has 40 trials (with 2 *repetitions* of 20 valid\* unique stimuli) and lasts <5 mins.
  + ‘valid’ = some stimuli are under & over sampled during the penalty & compressed blocks, as seen in the figure below. For ease of explanation, we consider here the baseline block, but the logic persists for the other blocks.
* For a valid, unique stimulus,
  + *Repetition* 1
    - Penalty shapes on the left, reward on the right
      * 50% of trials: starting point of the purple response line is on the *same* side of the divider as the correct answer
      * Other 50%: starting point of the purple response line is on the *other* side of the divider
  + *Repetition* 2
    - Reward shapes on the left, penalty on the right
      * 50% of trials: starting point of the purple response line is on the *same* side of the divider as the correct answer
      * Other 50%: starting point of the purple response line is on the *other* side of the divider
  + Note that repetitions are randomized (not ordered as 1 & 2)
* We also jitter noisiness of the stimulus, to increase the difficulty of the task. Specifically, each stimulus’ noise value can be +-1 (never 0). Note that +- do not indicate high vs low noise, but just the axis of noise; absolute value of noise is always 1.
  + For **block 1**, half (H1) of the trials have noise = 1, and the other half (H2) have noise = *-1*.
  + For **block 2**, half (H1) of the trials have noise = *-1*, and the other half (H2) have noise = 1.
  + There is also a harder version of the task with noise = +-2 which should not be used on the patients, but can be used on non-patient subjects.
* Subject/Session type: across sessions, we want to counterbalance the 1) block order and 2) which type of shape (say, curvier or flatter) is penalizing or rewarding:
  + Block order 1: BRP-BPR (B = baseline, P = penalty compressed, R = reward compressed)
  + Block order 2: BPR-BRP
  + Valence order 1: Curve=Penalty & Flat=Reward
  + Valence order 2: Curve=Reward & Flat=Penalty
  + Thus, there are 4 possible session types that can be run. **As noted in section A of this document, when 1) reading instructions to the patient, and 2) running the experiment, you will be prompted to type in the session type as A, B, C, or D. Please cycle through these session types to ensure across-subject counterbalancing, and update the *session\_type\_checklist*.**

**F. Block figures**

1. Simplified schematic below, assuming penalizing stimuli on left, and rewarding on right. This is the true distribution of the stimuli.

****

1. Actual schematic below, where stimulus space gets equally sampled due to counterbalancing penalty & reward appearing on each side. This is how the stimuli appear distributed to the subject. 2 dividers appear for each compressed block because the dividers also get flipped during counterbalancing.

