Getting Started (Massive Memory Group)]

Guidelines:

- **BEFORE DOING ANY CODE**, write down here with time and date (ex. 19:45 10/31)
- If you are actively working on part of the code write it down, so we don't have multiple people working on a singular task
-
- Experiment Checklist For Massive Memory

Methods Section: (500 words)
Your group's Methods/ Preregistration document, including any improvements you made or updates to the final experiment before you began data collection.

Experiment:

- Using JsPsych and GitHub.io
- Instructions Task
- 170 Image Study Block (150 images + 20 N-Back Task)
    - 4-Back Task where participants were tasked to press spacebar if an image shown earlier is shown again (shown again after 4 images)
- 50 image test block + Confidence test
    - Participants were shown two images that have changed either the exemplar (same object but changed color, changed type, or different shape) or state (object changes position or something is moved within object)
        - Provide examples
    - Participants were tasked to choose which out of the two images have they seen in the study block
- Confidence Test
    - Given a scale of 1 to 5, after each trial (two images shown), participants were asked how confident they were in their choice

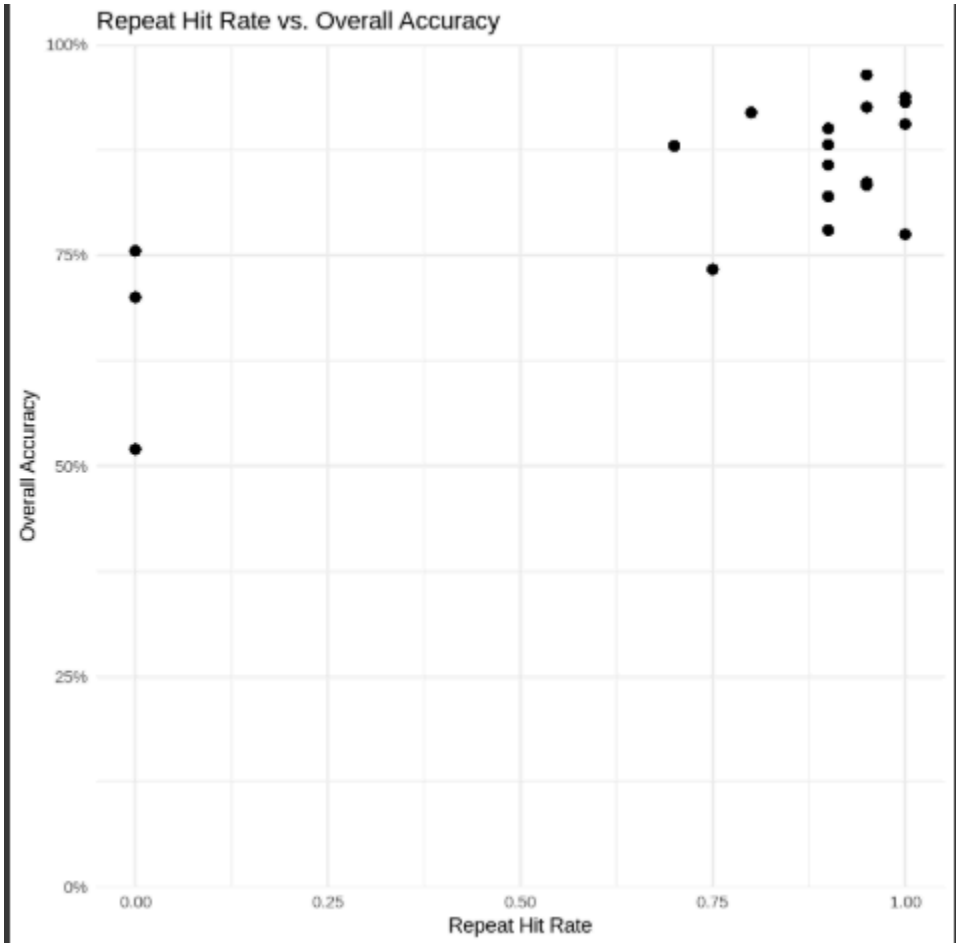
Improvements / Updates:

- Shortening experiment to 10 minutes (no novel trials, only exemplar and state, less trials for study and test period, no blocks for breaks, and fixed n-back test)
- Adding Confidence Rating for Test Trials
- Asking for what the experiment is about, technical details

Results: (500 words)
The Results section should briefly describe the main outcome from the data collected for your experiment, including the main descriptive statistics (e.g., the average and confidence intervals for the main dependent measure, for each condition; inferential

statistics are optional, but welcome!), and include one figure that communicates the main outcome. This section may be identical with the rest of your group.

Significance



Lecture 11/26: Analysis with our Data from Massive Memory Experiment

METHOD
- Descriptive statistics from Project [dependent measure & condition; provide numbers for generation -> data wrangling] -> massive memory (reaction time for n-back + accuracy for test_trials] (std, mean, range, median, standard error)
- Plot:
> general effect of what we are trying to ask in the experiment
> check the paper to view a plot comparison
DISCUSSION
- summary of what was found, what it means in introduction (implications), limitations from current experiment, future direction

- reflection on what is learned through the process of implementing an experiment

ANALYSIS ON R

- raw_data at github (each individual file)
- processed_data (all aprticipants turned to csv files)
- analysis folder shows the code on how to process the raw data into csv (if interested)

Analysis based on Original paper

LOOK AT /data/processed_data/massive_memory-processed-data.csv

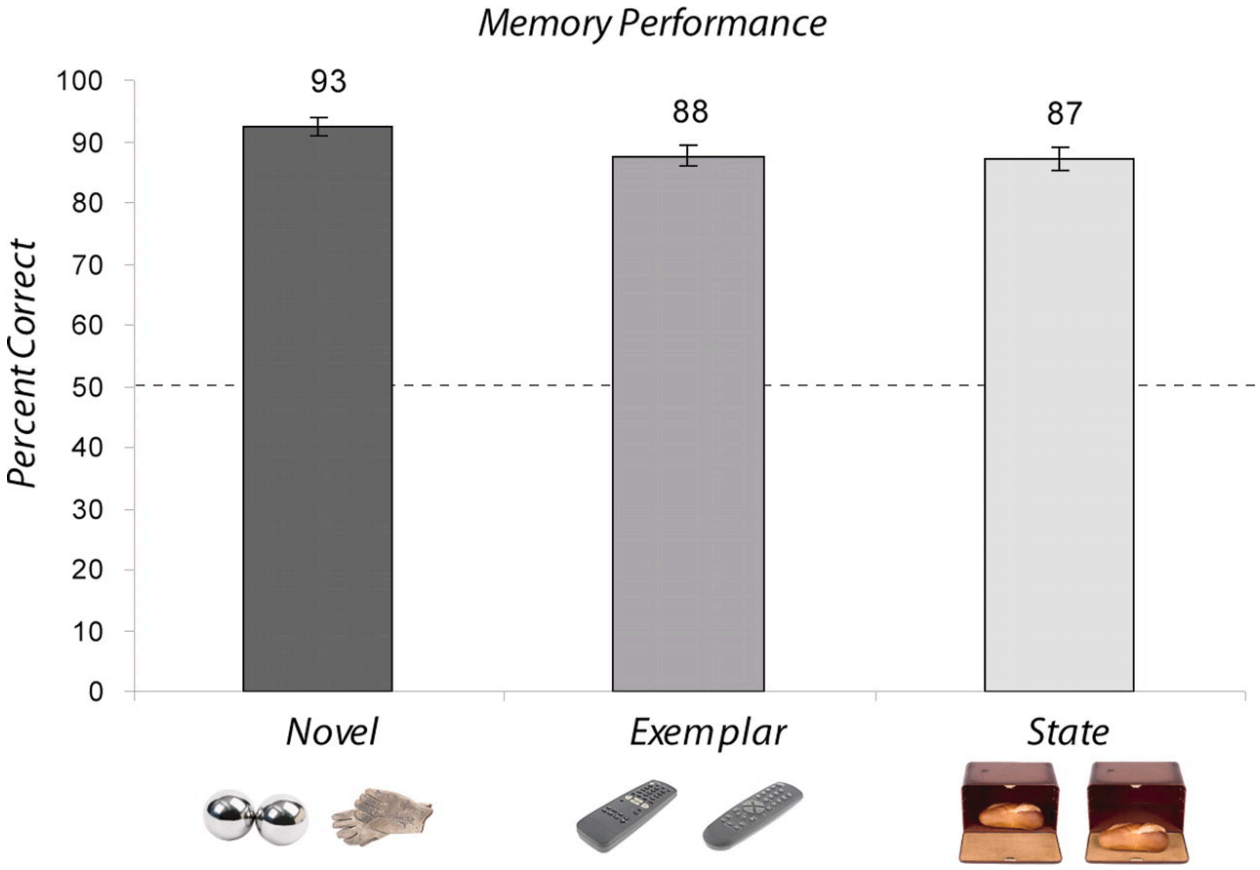
Notes:

- Data is provided:
    - Reaction Time
    - N-Back performance for each participant (mainly accuracy with false hit rates and accurate hit rates)
    - Confidence response based on image
    - Correction on each image (correct + is_right variable)

Martin suggested using the memory performance / percent correct visualization

- Posisbly add confidence scales

Example visualizations:



M

---------- **Latest Updates** ----------

Edwin - November 19 report:

- added Preload Images,
- removed feedback during test
- removed repeated incorrect trials
- completed memory trial list and test trial list to 50 images per condition (NOVEL, EXEMPLAR, STATE) 150 images total,
- added counterbalance logic to code and json file (https://gist.github.com/NeuraByte-UCSD-ITS/6a5ebaee57d49d0235ba938b518d7242 (https://gist.github.com/NeuraByte-UCSD-ITS/6a5ebaee57d49d0235ba938b518d7242)),
- added store trial_phase information for test trials

- added confidence rating per image selection
- added final debrief message,
- added Participant survey feedback,
- removed final data display ("jsPsych.data.displayData()"),
- included instructors code that adds participant ID entry field and fixation duration to 800ms

(Notes) Missing & recommendations:

1. implementing N-back
2. verifying counterbalance & trial_phase property
3. improve experiment messages/instructions/ratings/feedback descriptions (if necessary)

GitHub Repo: https://github.com/COGS119/project_materials/tree/main/massive_memory
GitHub Acc Here: SyeinW, darwinzyyu, JvalaDeus

- Edwin GitHub: NeuraByte-UCSD-ITS

Getting Started (Massive Memory Group)]

Guidelines:

- **BEFORE DOING ANY CODE**, write down here with time and date (ex. 19:45 10/31)
- If you are actively working on part of the code write it down, so we don't have multiple people working on a singular task
-
- Experiment Checklist For Massive Memory
- Copy of Group Project: Preregistration/ Methods Template ← fill out thursday

11/19

- **Meeting at Class & Online (8:30PM)**
- **Tasks:**
  - Tanishq: Preloading images and full memory trial and testing trial list

Preload 150 images -> testing 30 trials

- 30 of the 150 must match an image of the 150 for Study Image

Repeat 20 images of the 150 -> total 170; sample the positions at random, have a fixed list of possible positions

Function select different positions

Here is the participant IDs (0-249), randomly select 20, 2-back (sub section 2, 7, 23)

[list] ->

Sample without replacement

Add 2 to 5, 4 to 5, 8 to 5, final check-in for final

- Shane: Counterbalancing both conditions (old vs. new) for testing

Randomizing order of exemplar and state condition trials using multiple stimulus sets (randomly assigning participants to different sets)

Exemplar (Study & Foil)	Pair Physically Similar New Object in the same category
State (Study & Foil)	Same Studied object in different pose/state

[Left] & [Right] for testing period -> Assign each participant one of these sets and ensure it is presented an equal number of times each item appears on left / right

Around 36 items tested so:

ES & EF (18 times)	SS & SF (18 times)
EF & ES (18 times)	SF & SS (18 times)

- Edwin:

Storing trial_phase information (if the training block they receive is exemplar or state)
Break period for confidence in testing period

- Akhil:

Participant survey feedback (dynamic text) post-experiment
Debrief period

- Darwin

N-back

Meeting with Martin:

- Counterbalancing -> Randomizing each position for all categories using json
- N-back -> Look at function; is this the best way to shuffle through the list and should we tweak the percentage the 4-back is implemented
- How would we implement the OSF data collection using the participant ID

- Ask what type of post experiment questions are necessary for this experiment; what questions should be asked and how do you debrief?
-

Note for Method:

- Deviations from original paper & Reasonings
  - 3000 ms to 2000 ms duration
  - Change to n-back task
  - Experiment time

Changes to be made:

- Replace “are you still there?” with confident/not confident
- Re Add randomization to edwin’s branch
- Remove novel pairs(Change json file to 75/75 instead of 50/50/50)
- Make sure to only test images that have been shown
- Change survey questions, “what do you think this experiment was about?” and “What details did you use to distinguish between the items?”, “technical difficulties”
- Change image pushed to the timeline to implement counterbalancing, if we have time we can try to figure out true counterbalancing but

```
    timeline.push({
      type: jsPsychImageKeyboardResponse,
      stimulus: item.image1, // Displaying memory image
      choices: "NO_KEYS",
      trial_duration: 3000,
    });
  });
```

  Replacing the stimulus here some sort of random(image1, image2) would work as a pseudo counterbalance
- Add osf for data collection

```
// Filter image pairs for test phase based on current condition
var test_pairs = data.filter(item => item.condition.toLowerCase() === condition && item.role === "test");

// Add test trials (participants choose between two images & rate confidence)
test_pairs.forEach(pair => {
  var shuffledImages = jsPsych.randomization.shuffle([pair.image1, pair.image2]); // Randomize image positions
```

-
- ^ idk why this is the code, we are testing for things they have seen, this will just take a random selection out of the data, it doesn’t take a selection out of what is already seen. Solution would be to change ‘data’ to some new list containing all previously seen images in that study set. This code should work if we just change that.
- Also to add above, not sure why we are focusing on a single condition per study/testing phase(though i could have missed something you guys talked about since i was not present at the previous meeting)

-

Meeting Notes with Martin:

- Tweak n-back so its 20 repeated images
- Tweek timing from 3000 to 2000
- Test using jspsych local save
- Maybe remove confident/not confident on the same page, but rather on the next page. Also change from scale 1-5 rather than confident/not confident.(image slider response)

11/20 Meeting Questions(Shane)

- It's still 50/50 on state/exemplar, not 75/75 and we need 150 images, so we will have to fix that(fixed it) idk how to upload it
- Potentially switch to 72/72 since we doing 4 sessions of studying+testing 36 is a nice even number
- Can the testing phase choose any picture? Or just pictures from the study phase just before

Repeat-detection task

Instruction for test trial

https://github.com/SyeinW/cogs119_testrepo/tree/patch-1

- Changed fixation cross timing(800 → 500)
- Changed Image timing(3000 → 2000)
- Fixed json file so that it doesn't have anything about roles, and contains a set of 75 state and 75 exemplar images
- Randomized image order
- Counterbalancing
- Memory Phase
- Test Phase

Pseudocode:

- Memory Phase:

        - Should be in sets of 36
        - Can be any image set from the list, however once an image is shown, it is then removed from the list
        - The image set should be randomized, so you randomly get either image 1 or image 2
        - Fixation cross
        - Show image
- Test Phase:
    - Should be in sets of 10?(idk how many we decided on)
    - Can be any image shown from the list
    - Add Image slider response in between each image
    - End of the testing phase, participants type “yes” in response to “are you still there?”
- Above should repeat 4x
- Feedback Survey
    - 3 Questions
- Debrief
    - What was the experiment
- End

Changes to be made before final draft submission

```
// Randomizing image order
var shuffled_images = jsPsych.randomization.shuffle(data);
// stores images used in current round of studying
// If we are separating by each session move this into for loop
// iterate through each picture and add them to the study session and testing session
for (var i = 0; i < 4; i++) {
  // Take 1/4th of the images and puts them into the current study period
  var memory_images = jsPsych.randomization.sampleWithoutReplacement(shuffled_images, 12);
```

Change to foreach loop since we are only doing 1 study session

10/31

- **N-back 4**
  - **Sample Experiment: https://github.com/jspsych/tutorials/tree/master/n-back**
- **Focus on state and exemplar(add novel if there is time)**
- **Additional Experiment Tasks**
  - Fullscreen Mode (start open, finish exit)
  - Participant ID
    - Assign participant randomly / Remove confound in time taking experiment
    - Use for randomization for participant groups (state, exemplar, novel pairs)
    - Dictionary (key to value)
  - Test Task (trial)
  -

28 - how can we expand this list to encompass all items?

48 - how can we expand this list to encompass all items?

109 - add repeated trials

135 - how can we randomize image positions? How can we ensure that the old/new items are correctly marked?

1. Read your group's paper *carefully*.

Make sure you are clear on the basics:What is the big-picture question?

- What is the specific question in the experiment you will aim to replicate?
- What are the key conditions/ manipulations/ independent variables?
- What are the key dependent measures? How are participants asked to respond?
- What are the kinds of stimuli that are used?

2. Create a shared meeting notes document and assign a notetaker for each group meeting.

Meeting Notes (Massive Memory)

It's a good idea to keep track of your ideas/ discussions/ plans in a running document for your group. **Please create a notes document for your group using google docs (name it according to your project name, e.g. "False Recall Notes")**. Take notes as you answer some of the issues below and document decision points and questions (designing an experiment, even if it is replicating an existing experiment, will require a lot of decisions and generate a lot of questions!). Everyone should contribute to documenting your project's notes, but it usually makes sense to assign one person to the role of "note-taker" for each meeting. Make sure to rotate the note-taker for each meeting so you share responsibility for your group's work. **Once created, please share this document with both Aiden (aiveris@ucsd.edu) and me (mzettersten@ucsd.edu).** This notes document will be very useful to ***keep up-to-date for future meetings*** so we can more easily discuss issues/ questions/ goals.

3. Figure out how an individual core trial(s) of your experiment works.

Dive back into your article and read the Methods section carefully to figure out *exactly* how one single trial of your experiment works. I often like to create a small diagram showing the order of events so that I can visualize it more easily (often your paper may already include something like this! It's a good idea to screenshot or add something like this to your notes for easier reference).

- What is the sequence of events? (first, an image is shown; then,...)
- How long is the duration of each event? (including gaps between events)
- What responses are collected?

4. Try to implement a single trial in JsPsych.

Take a look at the plugins and try to figure out how you could construct the sequence of events using JsPsych plugins (remember that you can combine plugins to make more complex trials/ sequences of events!).

Then, create a fresh JsPsych script and try to implement a single trial (or key sequence of events in your experiment). You can reference the JsPsych tutorials we have used so far and the general JsPsych documentation.

Each experiment folder also has stimuli associated with it that you can use to implement your trial (don't worry too much about specifics of stimuli at this point though - just see if you can program the basic mechanics of your main trial(s)).

5. Next, focus on the broader conditions and the overarching trial structure of your experiment.

- What are different conditions that you need to assign participants into?
- How do trials vary depending on these conditions?
- How do stimuli vary across conditions?
- How many trials does each participant see? How many trials per condition? How many different kinds of trials are there? (training trials, test trials, etc.)
- What needs to be randomized? Do you want to counterbalance any of these elements? (counterbalancing means ensuring that each of a fixed set of options, e.g. a specific order of conditions, is seen approximately equally either within or across participants)
    - Order of trials?
    - Positions of stimuli on each trial?
    - Which response keys/ buttons/ options participants use?
- What could the trial list for *one* participant look like?

6. Ensure you have a way to store all of the data you care about.
Make sure you will be storing all of the information you need, including:

- DVs (reaction time, response choices, accuracy, ….)
- Condition information: between-participant condition, trial type, anything else you assigned participants too
- Information about each specific trial: the stimulus, position information, ….
- Trial numbers/ order
- Information about the participant: any information about the participant that you would like to store

---

Dictionary: input number and see if it connects to the value (key / value) and would count accuracy

Participant IDs: randomizatio