A PSYCHOPHYSICS PARADIGM FOR THE COLLECTION AND ANALYSIS OF SIMILARITY JUDGMENTS

A PREPRINT

Suniyya A. Waraich

Weill Cornell Graduate School of Medical Sciences
New York, NY 10065
saw4003@med.cornell.edu

Jonathan D. Victor

Feil Family Brain and Mind Research Institute
Weill Cornell Medical College
New York, NY 10065
jdvicto@med.cornell.edu

January 14, 2022

Abstract

Similarity judgments are commonly used to study mental representations and their neural correlates. This approach has been used to characterize perceptual spaces in many domains: colors, objects, images, words, and sounds. Ideally, one might want to compare estimates of perceived similarity between all pairs of stimuli, but this is often impractical. For example, if one asks a subject to compare the similarity of two items with the similarity of two other items, the number of comparisons grows with the fourth power of the stimulus set size. An alternative strategy is to ask a subject to rate similarities of isolated pairs, e.g., on a Likert scale. This is much more efficient (the number of ratings grows quadratically with set size rather than quartically), but these ratings tend to be unstable and have limited resolution, and the approach also assumes that there are no context effects. Here, a novel ranking paradigm for efficient collection of similarity judgments is presented, along with an analysis pipeline (software provided) that tests whether Euclidean distance models account for the data. Typical trials consist of 8 stimuli around a central reference stimulus: the subject ranks stimuli in order of their similarity to the reference. By judicious selection of combinations of stimuli used in each trial, the approach has internal controls for consistency and context effects. The approach was validated for stimuli drawn from Euclidean spaces of up to 5 dimensions. The approach is illustrated with an experiment measuring similarities among 37 words. Each trial yields the results of 28 pairwise comparisons of the form, “Was A more similar to the reference than B was to the reference?” While directly comparing all pairs of pairs of stimuli would have required 221445 trials, this design enables reconstruction of the perceptual space from 5994 such comparisons obtained from 222 trials.

Introduction

Humans mentally process and represent incoming sensory information in order to perform a wide range of tasks, such as object recognition, navigation, making inferences about the environment, and many others. Similarity judgments are commonly used to probe these mental representations (Edelman, 1998). Understanding the structure of mental representations can provide insight into the organization of conceptual knowledge (Hahn and Chater, 1997). It is also possible to gain insight into neural computations, by relating similarity judgments to brain activation patterns (Kriegeskorte and Kievit, 2013). Additionally, similarity judgments reveal what features are salient in perception (Hebart et al., 2020). Studying how mental representations change during development can shed light on how they are learned (Deng and Sloutsky, 2015). Thus, similarity judgments provide valuable insight into information-processing in the brain.

An oft-used model of mental representations using similarities is a geometric space model (Shepard, 1958; Coombs, 1954; Gärdenfors, 2000). Applied to sensory domains, this kind of model is often referred

to as a ‘perceptual space’ (Zaidi et al., 2013). Points in the space represent stimuli and distances between points correspond to the perceived dissimilarity between them. From similarity judgments, one can obtain quantitative estimates of dissimilarities; these pairwise dissimilarities (or perceptual ‘distances’) can then be used to model the perceptual space via multidimensional scaling (de Leeuw and Heiser, 1982).

There are multiple methods for collecting similarity judgments, each with its advantages and disadvantages. The most straightforward way of obtaining quantitative measures of dissimilarity is to ask subjects to rate on a scale the degree of dissimilarity between each pair of stimuli. While this is relatively quick, estimates tend to be unstable across long sessions as subjects cannot go back to previously made judgments, and context effects, if present, cannot be detected. (Here, a context effect is defined as a change in the judged similarity between two stimuli, based on the presence of other stimuli that are not being compared.) Alternatively, subjects can be asked to compare all pairs of stimuli to all other pairs of stimuli. While this would yield a more reliable rank ordering of dissimilarities, the number of comparisons required scales with the fourth power of the number of stimuli, making it feasible for only small stimulus sets. Quicker alternatives, like sorting into a predefined number of clusters (Tsogo et al., 2000) or free sorting have their own limitations. Free sorting (into any number of piles) is intuitive, but it forces the subject to categorize the stimuli, even if the stimuli do not easily lend themselves to categorization. The more recent multi-arrangement method, ‘inverse MDS,’ circumvents many of these limitations and is very efficient (Kriegeskorte and Mur, 2012). However, this method requires subjects to project their mental representations onto a 2D Euclidean plane and to consider similarities in a specific geometric manner, making the assumption that similarity structure can be recovered from Euclidean distances on a plane. Thus, there remains a need for an efficient method to collect large amounts of similarity judgments, without making assumptions about the geometry underlying the judgments.

Described here is a method that is both reasonably efficient and also avoids the above potential pitfalls. By asking subjects to rank stimuli in order of similarity to a central reference in each trial (Rao and Katz, 1971), relative similarity can be probed directly, without assuming anything about the geometric structure of the subjects’ responses. The paradigm repeats a subset of comparisons with both identical and different contexts, allowing for direct assessment of context effects as well as the acquisition of graded responses in terms of choice probabilities. The analysis procedure decomposes these rank judgments into multiple pairwise comparisons and uses them to build and search for Euclidean models of perceptual spaces that explain the judgments. The method is suitable for describing in detail the representation of stimulus sets of moderate sizes (e.g., 19 to 49).

To demonstrate the feasibility of the approach, an experiment was conducted, using a set of 37 animals as stimuli. Data was collected over the course of 10 hour-long sessions and then analyzed separately for each subject. Analysis revealed consistency across subjects and negligible context effects. It also assessed consistency of perceived dissimilarities between stimuli with Euclidean models of their perceptual spaces. The paradigm and analysis procedures outlined in this paper are flexible and are expected to be of use to researchers interested in characterizing the geometric properties of a range of perceptual spaces.

Protocol

Prior to beginning the experiments, all subjects provide informed consent in accordance with institutional guidelines and the Declaration of Helsinki. In the case of this study, the protocol was approved by the institutional review board of Weill Cornell Medical College.

1. Installation and Set-up

- 1.1 Download the code from the GitHub repository similarities (<https://github.com/jvlab/similarities>), by running:
`git clone https://github.com/jvlab/similarities.git`

If git is not installed, download the code as a zipped folder from the repository.

NOTE: In the repository are two subdirectories: “experiments”, which contains two sample experiments, and “analysis”, which contains a set of python scripts to analyze collected similarity data. In the experiments directory one (word_exp) makes use of word stimuli and the other (image_exp) displays image stimuli. Some familiarity with Python will be helpful, but not necessary. Familiarity with the command line is assumed: multiple steps require running scripts from the command line.

1.2 Install the following tools and set up a virtual environment:

1.2.1 **python3**: See the link for instructions: <https://realpython.com/installing-python/>. This project requires Python version 3.8.

1.2.2 **PsychoPy**: From the link (<https://www.psychopy.org/download.html>), download the latest standalone version of PsychoPy for the relevant operating system, using the blue button, under Installation. This project uses PsychoPy version 2021.2; the provided sample experiments must be run with the correct version of PsychoPy as specified below.

1.2.3 **conda**: From the link (<https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html#regular-installation>), download conda, through Miniconda or Anaconda, for the relevant operating system.

1.2.4 In the command line, run the following to create a virtual environment with the required python packages:

```
cd ~/similarities
conda env create -f environment.yaml
```

1.2.5 Check to see if the virtual environment has been created and activate it as follows:

```
conda env list # venv_sim_3.8 should be listed
conda activate venv_sim_3.8 # to enter the virtual environment
conda deactivate # to exit the virtual environment after running scripts
```

NOTE: Running scripts in an environment can sometimes be slow. Please allow up to a minute to see any printed output in the command line when you run a script.

1.3 To ensure that downloaded code works as expected, run the provided sample experiments using the steps below:

NOTE: The experiments directory (similarities/experiments) contains sample experiments (word_exp and image_exp), making use of two kinds of stimuli: words and images.

1.3.1 Open PsychoPy. Go to **View**, then click **Coder**, because PsychoPy’s default “Builder” cannot open .py files. Go to **File**, then click **Open**, and open **word_exp.py** (similarities/experiments/word_exp/word_exp.py).

1.3.2 To load the experiment, click the green **Run Experiment** button. Enter initials or name and session number and click **OK**.

1.3.3 Follow the instructions and run through a few trials to check that stimuli gray out when clicked. Press **Escape** when ready to exit.

NOTE: PsychoPy will open in fullscreen, first displaying instructions, and then a few trials, with placeholder text instead of stimulus words. When clicked, words gray out. When all words have been clicked, the next trial begins. At any time, PsychoPy can be terminated by pressing the Escape key.

NOTE: If the program terminates here, it is possible that the user’s operating system requires access to the keyboard and mouse. If so, a descriptive error message will be printed in the PsychoPy Runner window, which will guide the user.

1.3.4 Next, check that the image experiment runs with placeholder images. Open PsychoPy. Go to **File**. Click **Open** and choose **image_exp.pyexp** (similarities/experiments/image_exp/image_exp.pyexp).

1.3.5 To ensure the correct version is used, click the **Gear** icon. For the option Use PsychoPy version select **2021.2** from the dropdown menu.

1.3.6 As before, click the green **Run Experiment** button. Enter initials or name and session number and click **OK**.

NOTE: As in Step 1.3.2, PsychoPy will first display instructions and then render trials after images have been loaded. Each trial will contain 8 placeholder images surrounding a central image. Clicking on an image will gray it out. The program can be quit by pressing Escape.

1.3.7 Navigate to the data directory in each of the experiment directories to see output:
 similarities/experiments/image_exp/data
 similarities/experiments/word_exp/data

NOTE: Experimental data are written to the data directory. The responses.csv file contains trial-by-trial click responses. The log file contains all key presses and mouse clicks. It is useful for troubleshooting, if PsychoPy quits unexpectedly.

1.4 Optionally, to verify that the analysis scripts work as expected, reproduce some of the figures in the Results section as follows:

1.4.1 Make a directory for preprocessed data:

```
cd ~/similarities
mkdir sample-materials/subject-data/preprocessed
```

1.4.2 Combine the raw data from all the responses.csv files to one json file. In the command line, run the following:

```
cd ~/similarities
conda activate venv_sim_3.8
python -m analysis.preprocess.py
```

1.4.3 When prompted, enter the following values for the input parameters: 1) path to subject-data: “./sample-materials/subject-data”, 2), name of experiment: sample_word, and 3) subject ID: S7. The json file will be in similarities/sample-materials/subject-data/preprocessed.

1.4.4 Once data is preprocessed, follow the steps in the project README under Reproducing Figures. These analysis scripts will be run later to analyze data collected from the user’s own experiment.

2. Data Collection: Setting up a custom experiment

NOTE: Procedures are outlined for both the image and word experiments up to Step 3.2. Following this step, the process is the same for both experiments so the image experiment is not explicitly mentioned.

2.1 Select an experiment to run. Navigate to the word experiment or the image experiment:
 similarities/experiments/word_exp or similarities/experiments/image_exp.

2.2 Decide on the number of stimuli.

NOTE: The default size of the stimulus set is 37, but it can be varied, by changing the `num_stimuli` parameter in the analysis configuration file. This configuration file (similarities/analysis/config.yaml) can be opened in a source code editor. The experimental design requires the stimulus set size to equal $mk + 1$, for integers k and m . In the standard design, $k \geq 3$ and $m = 6$. Therefore, valid values for `num_stimuli` include 19, 25, 31, 37, 43, and 49 (see Table 1 for possible extensions of the design).

- 2.3 Finalize the experimental stimuli. If the word experiment is being run, prepare a list of words. For the image experiment, make a new directory and place all the stimulus images in it.

NOTE: Supported image types are png and jpeg. Do not use periods as separators in filenames: e.g., ‘image.1.png’ is invalid, but ‘image1.png’ or ‘image_1.png’ are valid.

- 2.4 If running the word experiment, prepare the stimuli as follows:

2.4.1 Create a new file in `experiments/word_exp` called `stimuli.txt`. This file will be read in Step 3.2.

2.4.2 In the file, write the words in the stimulus set as they are meant to appear in the display, with each word in a separate line. Avoid extra empty lines or extra spaces next to the words. See sample materials for reference (`similarities/sample-materials/word-exp-materials/sample_word_stimuli.txt`).

- 2.5 If the image experiment is being run, set the path to the stimulus set as follows.

2.5.1 In the `experiments` directory, find the configuration file called `config.yaml` (`similarities/experiments/config.yaml`).

2.5.2 Open the file in a source code editor and update the value of the `files` variable to the path to directory containing the stimulus set (Step 2.3). This is where PsychoPy will look for the image stimuli.

3. Next, use the stimuli to create ranking trials, as follows.

- 3.1 The user will need a `stimuli.txt` file. If the word experiment is being run, the file created in Step 2.4 can be used. Otherwise, it should be the list of filenames (for reference, see `similarities/sample-materials/image-exp-materials/sample_image_stimuli.txt`). Place this file in the appropriate experiment directory (`word_exp` or `image_exp`).

NOTE: Avoid extra empty lines, as well as any spaces in the names; use camelCase or snake_case for stimulus names.

- 3.1.1 Next, we create trial configurations. Open the `config.yaml` file in the analysis directory and set the value of the `path_to_stimulus_list` parameter to the path to `stimuli.txt` (created in Step 3.1). From the `similarities` directory, run the script by executing the following commands one after the other:

```
cd ~/similarities
conda activate venv_sim_3.8
python -m analysis.trial_configuration
conda deactivate # exit the virtual environment
```

- 3.1.2 This creates a file called `trial_conditions.csv` in `similarities` in which each row contains the names of the stimuli appearing in a trial, along with their positions in the display. A sample `trial_conditions.csv` file is provided (`similarities/sample-materials`).

NOTE At this point, a full set of 222 trials for one complete experimental run, i.e., for one full data set, has been generated. Figure 1A shows part of a conditions file generated by the above script, for the word experiment (see Results). Each row corresponds to a trial. The stimulus in the `ref` column appears in the center of the display. The column names `stim1` to `stim8` correspond to 8 positions along a circle, running counterclockwise, starting from the position to the right of the central reference. A sample trial from the word experiment is shown in Figure 1B.

- 3.2 Next, break these 222 trials into sessions and randomize the trial order. In the typical design, sessions comprise of 111 trials, each of which requires approximately an hour to run.

A

1	ref	stim1	stim2	stim3	stim4	stim5	stim6	stim7	stim8
2	monkey	ladybug	sparrow	hog	bear	snail	whale	lizard	goat
3	spider	sheep	pigeon	goldfish	horse	owl	snake	dolphin	turkey
4	turkey	cat	sheep	mouse	snail	snake	owl	giraffe	butterfly
5	cow	shark	sheep	whale	spider	goat	snail	duck	crocodile
6	butterfly	crocodile	sheep	tiger	snail	turkey	ant	owl	bear
7	mouse	rat	goldfish	giraffe	snake	whale	dolphin	cow	frog
8	fox	crocodile	turkey	shark	spider	butterfly	owl	snail	duck
9	butterfly	dog	spider	mouse	bear	goldfish	ant	pigeon	rat
10	cow	whale	crocodile	owl	mouse	turtle	bluebird	elephant	giraffe
11	fox	elephant	mouse	snail	ant	spider	hog	sheep	cat

B

Figure 1: **Representative examples of trials.** (A) Each row contains the details of a single trial. Headers indicate the position of the stimulus around the circle. The stimulus under ‘ref’ appears in the center and ‘stim1’ to ‘stim8’ appear around the reference. (B) The first trial (row) from Fig. A is rendered by PsychoPy to display the 8 stimuli around the reference stimulus, ‘monkey’.

3.2.1 To do this, in the command line run the following:

```
conda activate venv_sim_3.8
cd ~/similarities
python -m analysis.randomize_session_trials
```

3.2.2 When prompted, enter the following input parameters: (1) path to trial_conditions.csv created in Step 3.1.2, (2) output directory, (3) number of trials per session: 111 and (4) number of repeats: 5. When prompted, enter the following input parameters: (1) path to trial_conditions.csv created in Step 3.1.2, (2) output directory, (3) number of trials per session: 111 and (4) number of repeats: 5. (Step-by-step instructions for doing the above manually can be found in the README under Create Trials.)

3.3 Rename and save each of the generated files as “conditions.csv”, in its own directory. See the recommended directory structure (similarities/sample-materials/subject-data) and in the project README.

NOTE: Each experiment is repeated 5 times, over 10 hour-long sessions, each on a separate day. Subjects are asked to come for only one session per day to avoid fatigue. See Table 1 for the number of trials and sessions needed for stimulus sets of different sizes.

4. Running the experiment and collecting similarity data

4.1 Explain the task to the subjects and give them instructions. In each trial, subjects will view a central reference stimulus surrounded by 8 stimuli and be asked to click the stimuli in the surround, in order of similarity to the central reference, i.e., they should click the most similar

first and least similar last.

- 4.2 Ask them to try to use a consistent strategy. Tell them that they will be shown the same configuration of stimuli multiple times over the course of the 10 sessions.
- 4.3 If the study probes representation of semantic information, ensure that subjects are familiar with the stimuli before starting.
- 4.4 Navigate to the experiment directory (Step 2.1).
- 4.5 If this is the first time running the experiment, create a directory called `subject-data` to store subject responses. Create two subdirectories in it: `raw` and `preprocessed`. For each subject, create a subdirectory within `subject-data/raw`.
- 4.6 Copy the `conditions.csv` file (prepared in Step 3) for the specific session and paste it into the current directory, i.e., the directory containing the `psyexp` file. If there is already a file there, named `conditions.csv`, make sure to replace it with the one for the current session.
- 4.7 Open PsychoPy and then open the `psyexp` or `py` file in the relevant experiment’s directory.
- 4.8 In PsychoPy, click on the green play button to run the experiment.
- 4.9 In the modal pop-up, enter the subject name or ID and session number. Click **OK** to start. Instructions will be displayed at the start of each session.
- 4.10 Allow the subject about an hour to complete the task. The task is self-paced, and subjects should be encouraged to take breaks if needed.
- 4.11 When the subject finishes the session, PsychoPy will automatically terminate, and files will be generated in the `similarities/experiments/<image or word>_exp/data` directory. Transfer these into the `subject-data/raw/<subjectID>` directory (created in Step 4.5). See README for the recommended directory structure.

NOTE: As mentioned, the log file is for troubleshooting. The most common cause for PsychoPy to close unexpectedly is that a subject accidentally presses “Escape” during a session. If this happens, responses for trials up until the last completed trial will still be written to the `responses.csv` file. Create a new `conditions.csv` file, with only the trials that had not been attempted, replace the existing session’s `conditions` file with this one, and rerun the experiment. Be sure to save the generated files in the appropriate place. At the end of the session, the two `responses` files can be manually combined into one, though this is not necessary.

- 4.12 For each of the remaining sessions, repeat Steps 4.6 to 4.11.
5. After all sessions are completed, combine the raw data files and reformat them into a single json file for further processing. To do this, run `preprocess.py` in the terminal (`similarities/analysis/preprocess.py`).
 - 5.1

```
cd ~/similarities
conda activate venv_sim_3.8
python -m analysis.preprocess
```
 - 5.2 When prompted, enter the requested input parameters: the path to the `subject-data` directory, subject IDs for which to preprocess the data, and the experiment name (used to name the output file). Press **Enter**.

5.3 Exit the virtual environment: `conda deactivate`

NOTE: This will create a json file in the output directory that combines responses across repeats for each trial. Similarity data is read in from subject-data/raw and written to subject-data/preprocessed.

6. Analyzing similarity judgments

NOTE: Subjects click stimuli in order of similarity to the reference, thus providing a ranking in each trial. As each trial is repeated 5 times, 5 rank orderings of the same 8 stimuli are obtained (see Figure 2B). These rank judgments are interpreted as a series of comparisons in which a subject compares pairs of perceptual distances. It is assumed the subject is asking the following question before each click: "Is the (perceptual) distance between the reference and stimulus A smaller than the distance between the reference and stimulus B?" As shown in Figure 2C, this yields choice probabilities for multiple pairwise similarity comparisons for each trial. The analysis below uses these choice probabilities.

Figure 2: **Obtaining choice probabilities from ranking judgments.** (A) An illustration of a trial from the word experiment we conducted. (B) 5 rank orderings obtained for the same trial, over the course of multiple sessions. (C) Choice probabilities for the pairwise dissimilarity comparisons that the ranking judgments represent.

6.1 Determine pairwise choice probabilities from rank order judgments.

6.1.1 In similarities/analysis run `describe_data.py` in the command line. When prompted, enter the path to subject-data/preprocessed and next, the list of subjects for which to run the analysis.

NOTE: This will create three kinds of plots: i) the distribution of choice probabilities for a given subject's complete data set, ii) heatmaps to assess consistency across choice probabilities for pairs of subjects, and iii) a heatmap of choice probabilities for all comparisons that occur in two contexts to assess context effects. Operationally, this means comparing choice probabilities in pairs of trials that contain the same reference and a common pair of

stimuli in the ring but differ in all other stimuli in the ring: the heatmap shows how the choice probability depends on this context. For details on input parameters for the analysis scripts, refer to the project README (under Usage).

- 6.2 Generate low-dimensional Euclidean models of the perceptual spaces, using the choice probabilities. Run `model_fitting.py` in the command line as follows:

```
6.2.1 cd ~/similarities
      conda activate venv_sim_3.8
      python -m analysis.model_fitting
```

- 6.2.2 Provide the following input parameters when prompted: directory to the subject-data/preprocessed, the number of stimuli (37 by default), the number of iterations (the number of times the modeling analysis should be run), the output directory, and the amount of Gaussian noise (0.18 by default).

NOTE: This script takes a few hours to run. When finished, npy files containing the best-fit coordinates for 1D, 2D, 3D, 4D and 5D models describing the similarity data will be written to the output directory. A csv file containing log-likelihood values of the different models will be generated.

- 6.3 Visualize the log-likelihood of the obtained models and assess their fit. To do so, run `model_fitting_figure.py` in the command line:

```
6.3.1 cd ~/similarities
      python -m analysis.model_fitting_figure
```

- 6.3.2 When prompted, input the needed parameter: the path to the csv files containing log-likelihoods (from Step 6.2).

NOTE: A figure will be generated, showing log-likelihoods on the y-axis and model dimensions on the x-axis. As a sanity check, two models in addition to the Euclidean models are included: a random choice model and a "best" possible model. The random choice model assumes subjects click randomly. Thus, it provides an absolute lower bound on the log-likelihood for any model that is better than random. Similarly, as an upper bound for the log-likelihood (labeled "best"), there is the log-likelihood of a model that uses the empirical choice probabilities as its model probabilities. No Euclidean model should outperform the "best" model, as the "best" model is, by design, overfit and unconstrained by geometrical considerations. The likelihoods plotted are relative to the "best" log-likelihood.

- 6.4 Visualize the perceptual spaces for each subject: generate scatterplots showing the points from the 5D model projected onto the first two principal components. To do so, run `perceptual_space_visualizations.py` in the command line:

```
6.4.1 cd ~/similarities
      python -m analysis.perceptual_space_visualizations
```

- 6.4.2 When prompted, input the parameters: the subject IDs (separated by spaces) and the path to the npy file containing the 5D points obtained from Step 6.2.

- 6.4.3 After the script has finished executing, exit the virtual environment:
`conda deactivate`

NOTE: This script is for visualization of the similarity judgments. It will create a 2D scatter plot, by projecting the 5D points on to the first two principal components, normalized to have equal variance. Two points will be farther apart if the subject considered them less similar and vice versa.

Representative Results

To demonstrate feasibility and reproducibility, an experiment was conducted in which the stimulus set comprised the names of 37 animals. Complete datasets were collected from eight healthy subjects with normal vision as part of a study. To demonstrate the method, data from 3 of those subjects are shown here, two of whom were naïve to the purpose of the study. Informed consent was obtained in accordance with the Declaration of Helsinki and the institutional guidelines of Weill Cornell Medical College.

After data collection, the initial processing outlined above (Protocol Section 5) was performed. Subjects' responses in each trial were interpreted as a set of independent, binary choices of the form "Is the distance between the reference and s_1 less than that between the reference and s_2 ?" for all pairs of stimuli in the surrounding ring. Rank judgments were decomposed into such pairwise choices, as shown in Fig 2C.

Figure 3: **Consistency across subjects.** (A) Distribution of choice probabilities across three subjects for all pairwise comparisons. (B) Choice probabilities for the same pairwise comparisons across pairs of subjects. The color bar shows the ratio of the observed joint probability to the independent joint probability. High values along the main diagonal indicate consistency across subjects.

Fig. 3A shows the distribution of these choice probabilities, which was highly consistent across subjects (Protocol Section 6.1). Since each trial was repeated 5 times, choice probabilities took the following values: 0, 0.2, 0.4, 0.6, 0.8 and 1. The most frequent choice probabilities are 0 and 1, amounting to 50-70% of all decisions in each of the subjects; these are the judgments for which one option is chosen each time. For example, judging the distance between s_1 and s_2 as less than that between s_1 and s_3 0 out of 5 times would correspond to a choice probability of zero; making this judgment 5 out of 5 times would correspond to a choice probability of 1. Notably, there is a great deal of consistency in choice probabilities between the subjects, even for the judgments that are not at the extremes, as seen by the clustering of the data near the diagonal in each of the panels in Figure 3B.

Next, context effects were assessed. This was possible because of an important feature of the experimental design: many triplets of a reference stimulus and two comparison stimuli s_1 and s_2 are repeated in two contexts (i.e., with distinct sets of 6 other stimuli to complete the stimulus array). Then, the choice probability for each pairwise comparison was tabulated in each context separately. The dominant diagonal in Figure 4 indicates that for each subject the choice probabilities in the two contexts – including the choice probabilities that are intermediate between 0 and 1 – are close to identical. If choice probabilities were heavily dependent on context, they would not be strongly correlated, and this diagonal would not be prominent.

Context effects were also assessed by a statistical measure. The measure of context effect is constructed as follows. The first step is to compute an “imbalance” statistic for the observed dataset (detailed below), which quantifies the extent to which the observed judgments appear to depend on context. We then construct 10000 simulated datasets with the same trial configurations, trial counts, and overall choice probabilities

Figure 4: **Context effects.** Choice probabilities for all pairwise comparisons that were made in two contexts, for each of three subjects. “A” refers, arbitrarily, to one context in which a triad was presented, and “B” refers to the other context. The color bar shows the ratio of the observed joint probability to the independent joint probability. High values along the main diagonal indicate a lack of context effects.

as the actual data, but generated in a way that contains no context effects – by randomly assigning the observed judgments to the two contexts. We next compute the imbalance statistic for these simulated datasets just as was done for the observed responses. Finally, we compare the imbalance statistic for the observed responses with the imbalance statistic for the simulated datasets, to determine the probability that the observed imbalance could have been obtained from a dataset with no context effect. An empirical p-value of < 0.05 suggests that a context effect is present. For the data in Figure 4, p-values were 0.976, 0.299 and 0.329, for S4, S7 and S9 respectively, i.e., all > 0.05 .

The imbalance statistic for a dataset is computed by a sum of contributions over all triads that occur in two contexts. The contribution for each triad (comparing, say, $d(ref, s_1)$ with $d(ref, s_2)$) is determined as follows. First, the judgments for this triad are tallied into a 2x2 table. The columns correspond to the two contexts, so the column sums are constrained by the total number of presentations in that context. The rows correspond to counts of the alternative judgments, $d(ref, s_1) < d(ref, s_2)$ or $d(ref, s_1) > d(ref, s_2)$, so the row sums are constrained by the observed choices, summed across contexts. Since the two-tailed Fisher exact test¹⁴ yields the probability that a table with the observed (or greater) interaction between rows and columns (judgments and contexts) would be seen if no interaction is actually present, we use the negative logarithm of this probability as the contribution of this triad to the overall imbalance statistic. Summing the negative logarithms to create an overall imbalance statistic thus captures the joint probability of the observed imbalance across triads, under the null hypothesis.

To model the mental representation of the animal names, Euclidean models of perceptual spaces of 1, 2, 3, 4 and 5 dimensions were derived using a maximum likelihood approach. Subjects’ responses were modeled as decisions reflecting the comparison of two distances with additive Gaussian noise representing errors in estimation, i.e., noise at the decision stage. Figure 5 shows log-likelihoods (per decision) of 5 Euclidean models. The log-likelihoods are shown relative to the log-likelihood of the “best” model, i.e., a model that assigns the observed choice probability to each comparison, without constraining these probabilities by any geometric consideration. To put these log-likelihoods into perspective, the log-likelihood of a “random-choice” model is also indicated; this serves as a lower bound for model performance. The model fit improves with each added dimension. The biggest jump is between the 1D and 2D model, indicating that a simple 1D model fails to fully explain the data. However, the plateau around dimensions 4 to 5 indicates that even the 5D model does not completely capture the distances that account for the similarity judgments. To validate the approach, the pipeline was also run on simulated data. Separate experiments were simulated to generate similarity judgments between points drawn from 1D, 2D, 3D, 4D and 5D spaces respectively. In all cases, the method correctly identified the dimensionality. Furthermore, a model with the correct dimensionality yielded a log-likelihood that agreed with the ground truth log-likelihood obtained from the model.

Figure 5: **Results of model-fitting analysis.** Relative log-likelihoods for models of different dimensions as well as for the random choice (lower bound) model, shown for three subjects. A relative log-likelihood of zero corresponds to the log-likelihood of the “best” model, in which choice probabilities match the empirical choice probabilities without consideration of geometry.

Figure 6: **The perceptual space of one subject (S7) in more detail.** The projection of 5D coordinates obtained from the modeling, projected onto the first two principal components in (A) and onto the first and third principal components in (B). Axes are scaled so that the variance along each axis is equal.

Finally, the organization of points in the perceptual space models was visualized. Figure 6 shows these data for one subject, S7. Principal component analysis (PCA) was performed on the points from the 5D model of the perceptual space. Points projected onto the first two and first and third principal components respectively are shown in Figures 6A and 6B, with axes normalized for equal variance. Distances between points aligned with the similarity judgments obtained experimentally: animals perceived as similar were denoted by points that were near each other.

Discussion

The protocol outlined here is effective for obtaining and analyzing similarity judgments for stimuli that can be presented visually. The experimental paradigm, the analysis and possible extensions are discussed first, and later the advantages and disadvantages of the method.

Experimental Paradigm

The proposed method is demonstrated using a domain of 37 animal names, and a sample dataset of perceptual judgments is provided, so that one can follow the analysis in Step 6 and reproduce parts of Figures 3-6 (Protocol Section 1.4). The experimental design groups these 37 stimuli into 222 trials – each containing a reference stimulus in the center and 8 comparison stimuli in the surrounding ring – such that several criteria hold: a) each of the 37 stimuli appears as the reference an equal number (6) of times ($222=37 \times 6$), b) over the 6 trials in which a stimulus is the reference, all of the remaining 36 stimuli are used as comparison stimuli at least once, c) 24 stimuli occur in exactly one comparison with a given reference and d) 6 pairs of stimuli appear with the reference in two separate trials. This aspect of the paradigm, that 6 pairs of comparison stimuli occur in separate contexts for each reference stimulus, allows for checking context effects in Step 6 (see Results: Figure 4). This standard design yields $6216=222 \times 28$ comparisons of the form “Is the similarity of the reference to s_1 greater or less than the similarity of the reference to s_2 .” This efficiency is possible because each of the 222 trials yields a ranking of 8 similarities, and the 8 ranked similarities generate 28 pairwise comparisons. Of these 6216 comparisons, 222 are repeated, giving us 5994 unique comparisons.

Once the stimulus domain is chosen, the next most important design decision is the number of samples. Many alternative designs are possible (Table 1), with other choices for the way that stimuli are repeated in different contexts. As mentioned in the Results (Figure 4), within each trial there is a triplet – comprising the reference and two surrounding stimuli – that appears together in one other trial. The number of the surrounding stimuli overlapping with another trial with a common reference – in this case equal to 2 – is controlled by the `overlap` parameter in the analysis configuration file. Increasing this parameter would result in more stimuli being shared between two trials, allowing for more extensive comparisons of distance ranking, e.g., “Is s_1 more similar to the reference than s_2 and is s_2 more similar than s_3 ?” across two contexts. For examples of other experimental designs possible with different values of this and other parameters, see Table 1. For details on all the parameters, what they control and where to change them, see Table 2. Notably, it is also possible to change the number of stimuli that appear around a reference in each trial by changing the `num_stimuli_per_trial` parameter. Increasing the size of the surround would increase the number of comparisons per trial and better study context effects; decreasing it would reduce task complexity. The number of comparison stimuli in a trial (N_{circle}), the number of stimuli in the experiment (N_{stim}), the number of trials (N_{trials}), the number of unique comparisons ($N_{comparisons}$) and the number of repeated comparisons ($N_{repeated}$) are inter-related and depend on the size of the previously mentioned overlap between trials ($N_{overlap}$) and the number of trials per reference stimulus (k). The stimulus set size is determined by m , which is an arbitrary integer. These relationships are listed below:

$$\begin{aligned}
 N_{stim} &> N_{circle} \\
 N_{stim} &= mk + 1, m \geq 1 \\
 N_{circle} &= \frac{N_{stim} - 1}{k} + N_{overlap} \\
 N_{trials} &= kN_{stim} \\
 N_{comparisons} &= N_{trials} \frac{N_{circle}(N_{circle} - 1)}{2} \\
 N_{repeated} &= N_{trials} \frac{N_{overlap}(N_{overlap} - 1)}{2}
 \end{aligned}$$

There are other details of the paradigm and data collection procedures that help to minimize confounds. Randomizing the placement of stimuli and trial order (Step 3.2) is important, so that even when sessions are repeated, the subject does not begin to recognize spatial or temporal patterns in the placement of stimuli. It is also important not to give subjects any direct cues about how to gauge similarity (Step 4) as this can bias the results. They should themselves decide on what ‘similarity’ means to them in the context of the specific experiment. However, it is useful to debrief the subjects after they complete the experiment, as this may help understand how findings vary across subjects. If, for some reason, a session is corrupted or aborted, then we recommend deleting the entire session, so that all trials are completed an equal number of times.

Analysis of Similarity Data

The experiment yields, for each trial, rank-orderings of similarity between the N_{circle} comparison stimuli and the reference. When decomposed into comparisons of pairs of stimuli, these trials yield choice probabilities for each of the unique comparisons. The choice probabilities are then analyzed to search for geometric models of the perceptual space (Protocol Sections 5-6). The analysis attempts to account for the choice probabilities in terms of distances between stimuli, $d(s_i, s_j)$, in a Euclidean space. That is, the goal is to assign coordinates to each stimulus, so that the choice probability for clicking s_1 before s_2 reflects the probability that the subject judged $d(ref, s_1) < d(ref, s_2)$. This fitting procedure is described here both because it has some novel elements, and to enable a user to modify it (Protocol Section 6.2).

The analysis is a kind of multidimensional scaling problem, but with some distinguishing features. First, the data provide rank-ordering of dissimilarity judgments, rather than estimates of the distances. Second, the dataset, while extensive, only contains a subset of all possible comparisons of pairwise distances. Finally, the objective is to account for the choice probabilities, not just a binary decision of which distance is larger. With these considerations in mind, the cost function is chosen such that its value is minimized when model-predicted choice probabilities are most likely to yield the experimentally observed choice probabilities. It is therefore defined as the negative log-likelihood of the observed choice probabilities under the model, normalized by the total number of pairwise comparisons, and is adapted from previous work in the Victor lab (Victor et al., 2017):

$$cost = -\frac{1}{N_0} \sum_{r,i,j} (c_{r,i,j} \log_2 p_{r,i,j} + (N_{repeats} - c_{r,i,j}) \log_2 (1 - p_{r,i,j})) \quad (1)$$

where $N_0 = N_{comparisons} \cdot N_{repeats}$, and $N_{repeats}$ is the number of repeats of the protocol, (i.e., the number of times each unique trial is repeated), and

$$\begin{aligned} p_{r,i,j} &= P(d(s_r, s_i) < d(s_r, s_j)) \\ c_{r,i,j} &= C(d(s_r, s_i) < d(s_r, s_j)) \end{aligned}$$

Here, s_r denotes the reference stimulus in a trial, and s_i and s_j the stimuli in the ring around s_r . $P(d(s_r, s_i) < d(s_r, s_j))$ represents the model probability that the distance between s_r and s_i is judged as smaller than the distance between s_r and s_j and C denotes the number of times that the subject judged $d(s_r, s_i) < d(s_r, s_j)$. The objective of the modeling analysis is to find a model, i.e., a configuration of points in a Euclidean space, that accounts for the empirical choice probabilities. Iteratively, the minimization adjusts the coordinates assigned to each stimulus, and in so doing, the model choice probabilities (P). The minimization terminates when the cost function stops decreasing below the tolerance (an adjustable parameter **tolerance** controls this) or if the maximum number of iterations are reached (controlled by parameter **max_iterations**).

To connect the stimulus coordinates with the model choice probabilities, it is assumed that a subject – when choosing between two stimuli to click in a trial – will make an internal comparison of their relative distances to the reference, namely $d(s_r, s_i)$ and $d(s_r, s_j)$. These distances are (by default) the ordinary Euclidean distances between points assigned to stimuli s_r , s_i and s_j . Further, it is supposed that this mental comparison has an internal noise, which we model as an additive Gaussian source of standard deviation σ , a model introduced for one-dimensional domains by Maloney et al. (Maloney and Yang, 2003; Knoblauch and Maloney, 2008) and used for multidimensional domains in the Victor lab (Victor et al., 2017), the model choice probabilities are related to the coordinates by:

$$P(d(s_r, s_m) < d(s_r, s_n)) = \frac{1}{2} \left(1 + \left(\frac{d(s_r, s_m) - d(s_r, s_n)}{2\sigma} \right) \right) \quad (2)$$

The internal noise, σ , can be controlled by varying sigma in the analysis config file. To initialize the algorithm with a set of stimulus coordinates, the rank-order judgments were used to obtain a set of approximate distances and then applied standard multidimensional scaling10 to these distances to obtain the initial coordinates. These approximate distances were determined by tallying the “wins” and “losses” for each pair of stimuli. That is, looking over all pairwise comparisons in the data, each time a distance, $d(s_r, s_k)$ is judged larger than another, $d(s_r, s_n)$, a win is logged for the bigger distance $d(s_r, s_k)$ and a loss is logged for $d(s_r, s_n)$. The core idea is that the greater the distance between two stimuli, the more often it would be judged as greater than another distance (in terms of ‘wins’) and vice versa. After iterating through all comparisons and tabulating the wins and losses of each pair of stimuli, distance estimates are calculated as follows:

$$d_{init}(s_i, s_j) = \frac{wins(s_i, s_j) + 1}{wins(s_i, s_j) + losses(s_i, s_j) + 2}$$

Once this is done, the initial set of coordinates is determined by applying standard metric multidimensional scaling to $d_{init}(s_i, s_j)$.

The minimization routine thus described is run independently to obtain models of 1, 2, 3, 4 and 5 dimensions. In each case, the optimal coordinates of the inferred stimulus points, as well as the value of the cost function, i.e., the negative log-likelihood of the empirical choice probabilities are returned. The log-likelihood is plotted in Figure 5 relative to the best possible log-likelihood, which is calculated similarly as in Equation 1, with

$$P(d(s_i, s_j) < d(s_i, s_k)) = \frac{1}{N_{repeats}} C(d(s_i, s_j) < d(s_i, s_k))$$

for all comparisons. As a sanity check, in Figure 5, the “random model” log-likelihood, a lower bound by which to judge the performance of the models is also plotted. When calculating the random choice log-likelihood, we set $P(d(s_i, s_j) < d(s_i, s_k)) = \frac{1}{2}$ for all comparisons.

Possible Extensions:

Firstly, as mentioned previously, the experimental paradigm may be modified to accommodate stimulus sets of different sizes, and the number of stimuli in the ring can be altered to yield different numbers of pairwise comparisons per trial (see Table 1).

Secondly, it may be useful to utilize non-Euclidean distance metrics in the analysis. For example, one study found that the city-block metric better represented a perceptual space of surface lightness and illumination (Logvinenko and Maloney, 2006). The proposed method can be generalized, so models with other distance metrics e.g., a city-block distance, a Minkowski distance, or a hyperbolic distance (Zhou et al., 2018), are fit to similarity data. To do so, one would have to modify the provided code and implement an alternative distance metric. The main change needed is in line 105 (function name: `dist_model_ll_vectorized`) in the file `similarities/analysis/pairwise_likelihood_analysis.py`.

Strengths and Limitations:

A key strength of the proposed approach is that it provides a flexible framework for designing experiments with various stimulus set sizes, various numbers of comparisons, repeats, or number of stimuli per trial as well as various overlapping set sizes to measure context effects. By changing the size of overlap between trials and the size of the surround in a trial, one can probe the role of context in similarity judgments, while obtaining a high number of pairwise similarity judgments per trial. The method addresses many limitations of previous experimental paradigms for collecting similarity data. For example, unlike the arrangement-based methods (Kriegeskorte and Mur, 2012; Goldstone, 1994), (that require stimuli to be dragged and dropped on a 2D

Euclidean plane with similar items placed together and different items placed apart) and the sorting methods, that require stimuli to be categorized into piles (Tsogo et al., 2000), the ranking method does not prompt subjects to project their internal representation onto any geometric structure. Another limitation of some past methods – e.g., confusion matrices - in which two stimuli are considered similar if they are confused with each other in rapid recognition tasks (Townsend, 1971) - is that they do not yield graded measures. This method does yield graded measures, i.e., the choice probabilities.

As emphasized above, the collection method is flexible in that it does not assume that the internal representation is a Euclidean space. Here, the analysis method tests only Euclidean models; however, it can be extended to include non-Euclidean models as well, by localized modifications in the source code. However, the modeling framework is not designed to take context effects into account. Were they significant, it would place a caveat on the conclusions that could be drawn.

The proposed method is more time-efficient than the paired comparison approach. Each trial of the paradigm takes about 30 sec (subjects perform 111 trials in an hour), yielding $111 \times 28 = 3108$ comparisons per hour. Single-comparison trials are unlikely to take less than 3 sec per trial, which would yield 1200 comparisons per hour. Additionally, there is a second level of efficiency: the present approach does not require comparisons of all pairwise distances. For the example in the manuscript, the full set of pairwise distances amounts to 221445 comparisons, but in the present approach, a sparse subset of 5994 unique comparisons, each repeated 5 or 10 times, suffice to model the similarity data. However, the method, though efficient, is still time-consuming, and it requires a significant commitment from subjects. As a result, it is not a feasible approach for a set of hundreds of stimuli, unless data is pooled across subjects. Finally, the approach is not directly applicable to nonvisual stimuli.

Tables

Overlap Between Trials (N_overlap)	# Trials Per Ref (k)	m	# Stimuli (N_stim)	Ring Size (N_circle)	# Trials (N_trials)	Total Pairwise Comparisons (N_comparisons)	Total Repeated Comparisons (N_repeated)	Total Unique Comparisons
2	3	6	19	8	57	1596	57	1539
2	4	6	25	8	100	2800	100	2700
2	5	6	31	8	155	4340	155	4185
2	6	6	37	8	222	6216	222	5994
2	7	6	43	8	301	8428	301	8127
2	8	6	49	8	392	10976	392	10584
2	5	4	21	6	105	1575	105	1470
2	9	2	19	4	171	1026	171	855
3	4	8	33	11	132	7260	396	6864
2	10	3	31	5	310	3100	310	2790

Table 1: Example parameter sets. The experimental paradigm can be varied to have fewer or more stimuli, trials and pairwise comparisons. The fourth row contains the parameters we used.

Parameter	Description	Default Value
Parameters in analysis/config.yaml		
num_stimuli	Size of stimulus set	37
overlap (N_overlap)	Size of the overlapping set of stimuli across two trials	2
num_stimuli_per_trial (N_circle)	Number of stimuli appearing around a reference in each trial	8
path_to_stimulus_list	Path to the text file containing names of all stimuli, one per line.	'./sample-materials/image-exp-materials/sample_image_stimuli.txt'
model_dimensions	The number of dimensions for which to test Euclidean models, in a list.	[1, 2, 3, 4, 5]
sigma	Standard deviation of the Gaussian error distribution	0.18
epsilon	A small number to prevent division by zero errors in the calculation of log-likelihoods. There is no need to change this.	1.00E-30
num_repeats	Number of times each trial is repeated. This parameter depends on the experimental design.	5
max_trials	Maximum number of trials to use in the modeling analysis. By default, we use all the data, but one can also run the analysis on subsets of comparisons.	6000
minimization	The type of minimization algorithm to use: gradient descent or Nelder-Mead minimization. Allowed options are 'nelder-mead' and 'gradient-descent.'	gradient-descent'
tolerance	The minimum allowed change between consecutive iterations of gradient descent. If the gradient times learning rate falls below this value, the minimization routine terminates.	1.00E-05
max_iterations	The maximum number of iterations of gradient descent.	50000
learning_rate	Learning rate or step size of gradient descent. Only applicable when minimization='gradient-descent'	0.01
fatol	The minimum allowed change in cost function between consecutive iterations of Nelder-Mead minimization. Only applicable when minimization='nelder-mead.'	1.00E-05
Parameters in experiments/config.yaml		
files	Path to the directory containing stimulus set of images for running the image experiment	'./../sample-materials/sample-images'
num_images_per_trial (N_circle)	Number of stimuli appearing around a reference in each trial of the image experiment	8
display_radius	Size of the display in the image experiment (in pixels)	250.8
image_size	Height and width of stimulus images (in pixels)	115.0
num_words_per_trial (N_circle)	Number of stimuli appearing around a reference in each trial of the word experiment	8
word_display_radius	Size of the display in word experiment (in degrees, NOT pixels)	10.0
text_stim_height	Parameter controlling the height of the text stimuli in word experiment	1.2

Table 2: Parameters in analysis/config.yaml and experiments/config.yaml

Acknowledgments

The work is supported by funding from the National Institutes of Health (NIH), grant EY07977. The authors would also like to thank Usman Ayyaz for his assistance in testing the software, and Muhammad Naeem Ayyaz for his comments on the manuscript.

References

- S. Edelman. Representation is representation of similarities. *Behav. Brain Sci.*, 21(4):449–467, Aug 1998. ISSN 0140-525X. doi:10.1017/s0140525x98001253.
- U. Hahn and N. Chater. Concepts and similarity. In *Knowledge, Concepts and Categories*, page 43–84. MIT Press, Cambridge, MA, USA, Oct 1997.
- N. Kriegeskorte and R.A. Kievit. Representational geometry: integrating cognition, computation, and the brain. *Trends in Cognitive Sciences*, 17(8):401–412, 2013. doi:10.1016/j.tics.2013.06.007.
- M.N. Hebart, C.Y. Zheng, F. Pereira, and C.I. Baker. Revealing the multidimensional mental representations of natural objects underlying human similarity judgments. *Nature Human Behaviour*, 4(11):1173–1185, 2020. doi:10.1038/S41562-020-00951-3.
- W.S. Deng and V.M. Sloutsky. The development of categorization: Effects of classification and inference training on category representation underlying human similarity judgments. *Developmental Psychology*, 51(3):392–405, 2015. doi:10.1037/A0038749.
- R.N. Shepard. Stimulus and response generalization: Tests of a model relating generalization to distance in psychological space. *Journal of Experimental Psychology*, 55(6):509–523, 1958. doi:10.1037/H0042354.
- C.H. Coombs. A method for the study of interstimulus similarity. *Psychometrika*, 19(3):183–194, 1954. doi:10.1007/BF02289183.
- Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, Mar 2000. doi:10.7551/mitpress/2076.001.0001.
- Q. Zaidi, J. Victor, J. McDermott, M. Geffen, and T.A. Bensmaia, S. and Cleland. Perceptual spaces: mathematical structures to neural mechanisms. *The Journal of neuroscience: the official journal of the Society for Neuroscience*, 33(45):17597–17602, 2013. doi:10.1523/JNEUROSCI.3343-13.2013.
- Jan de Leeuw and Willem Heiser. 13 Theory of multidimensional scaling. In *Handbook of Statistics*, volume 2, pages 285–316. Elsevier, Waltham, MA, USA, Jan 1982. doi:10.1016/S0169-7161(82)02016-1.
- L. Tsogo, M. H. Masson, and A. Bardot. Multidimensional Scaling Methods for Many-Object Sets: A Review. *Multivariate Behav. Res.*, 35(3):307–319, Jul 2000. ISSN 0027-3171. doi:10.1207/S15327906MBR3503_02.
- Nikolaus Kriegeskorte and Marieke Mur. Inverse MDS: Inferring Dissimilarity Structure from Multiple Item Arrangements. *Front. Psychol.*, 0, 2012. ISSN 1664-1078. doi:10.3389/fpsyg.2012.00245.
- Vithala R. Rao and Ralph Katz. Alternative Multidimensional Scaling Methods for Large Stimulus Sets on JSTOR. *Journal of Marketing Research*, 8(4):488–494, Nov 1971. URL <https://www.jstor.org/stable/3150241>.
- Jonathan D. Victor, Syed M. Rizvi, and Mary M. Conte. Two Representations of a High-Dimensional Perceptual Space. *Vision Res.*, 137:1, Aug 2017. doi:10.1016/j.visres.2017.05.003.
- Laurence T. Maloney and Joong Nam Yang. Maximum likelihood difference scaling. *J. Vis.*, 3(8):5, Sep 2003. ISSN 1534-7362. doi:10.1167/3.8.5.
- Kenneth Knoblauch and Laurence T. Maloney. Estimating classification images with generalized linear and additive models. *J. Vis.*, 8(16):10, Dec 2008. ISSN 1534-7362. doi:10.1167/8.16.10.
- Alexander D. Logvinenko and Laurence T. Maloney. The proximity structure of achromatic surface colors and the impossibility of asymmetric lightness matching. *Percept. Psychophys.*, 68(1):76–83, Jan 2006. ISSN 1532-5962. doi:10.3758/BF03193657.
- Yuansheng Zhou, Brian H. Smith, and Tatyana O. Sharpee. Hyperbolic geometry of the olfactory space. *Sci. Adv.*, Aug 2018. URL <https://www.science.org/doi/10.1126/sciadv.aag1458>.
- Robert Goldstone. An efficient method for obtaining similarity data. *Behavior Research Methods, Instruments, & Computers*, 26(4):381–386, Dec 1994. ISSN 1532-5970. doi:10.3758/BF03204653.
- J. T. Townsend. Theoretical analysis of an alphabetic confusion matrix. *Percept. Psychophys.*, 9(1):40–50, Jan 1971. ISSN 1532-5962. doi:10.3758/BF03213026.