Technical Report:

Formal Modeling of the Effects of Training Variability on Classification Learning and Generalization

Mingjia Hu, Robert M. Nosofsky

The main purpose of this experiment is to study the effect of training-pattern variability on classification learning and generalization in the dot-pattern paradigm. In our experiments, subjects were first trained to classify a set of training patterns into three categories, and then tested on selected novel patterns as well as a subset of training patterns. The same set of training patterns were repeated across the 10 training blocks with the order of presentation randomized within each block.

There are four between-subject conditions that differ in terms of the variability of the training patterns. In each condition, 9 unique training patterns were generated around each of the three category prototypes that are pre-defined and shared across all subjects (27 patterns in total). The category prototypes were distorted by various levels using the Posner-Keele (1968) statistical-distortion algorithm to generate the training patterns for the four training conditions: all low-distortions, all medium-distortions, all high-distortions, and mixture (equal number) of the three distortion levels, respectively. The test patterns consisted of 27 old distortions that were presented in the training phase (9 per category), 3 prototypes (1 per category), 9 new low-level distortions (3 per category), 18 new medium-level distortions (6 per category), 27 new high-level distortions (9 per category) and 3 additional high-level distortions (1 per category) that are hand-curated to be difficult to classify. Each pattern was presented once in a random order for each subject for a total of 87 trials. The same prototypes and the hard-to-classify distortions were presented across the four conditions, but different sets of old distortions were sampled from the training patterns in the respective conditions. Due to a coding error, the novel distortions were not held fixed across conditions, but the same set of novel patterns were presented for all subjects in the same condition.

In both the learning and transfer phases, each pattern was presented at the center of the computer screen and remained visible until a subject responded with a key press. In the learning phase, the corrective feedback on each trial appeared for 2s below the presented pattern.

Figures 1 and 2 summarize the results from the experiment. Figure 1 shows the average proportion of correct classification responses over the training blocks for each of the four training conditions. Across the training conditions, the classification accuracy gradually improves over the course of training. Particularly, the low-distortion training condition exhibits the fastest rate of improvement, with accuracy reaching 90% at the end of training. The medium- and mixed-distortion conditions showed moderate rate of learning and intermediate rate of accuracy at the end of training, while the high distortion condition has the lowest terminal accuracy around 50%.

To confirm these observations, we conducted a 2x10 mixed-model ANOVA using training conditions (low, med, high, mixed) and blocks as factors. The analysis revealed a significant main effect of blocks, $F(6.35, 1905.07) = 84.44$, $p < .001$, $\eta^2 = .220$. The main effect of training conditions was also significant, $F(3,300) = 82.85$, $p < .001$, $\eta^2 = .453$, as was the interaction effect between learning condition and blocks, $F(19.05, 1905.07) = 2.865$, $p < .001$, $\eta^2 = .028$. The mean classification accuracy for the final training blocks is higher in the low condition than in the medium condition, $t(132.3) = 8.05$, $p < .001$, and higher in the medium condition than in the high condition, $t(151.0) = 6.33$, $p < .001$.

Figure 2 shows the average proportion of correct responses for various types of test patterns. The general trend is that the classification accuracy is the highest for the prototypes, and gradually decreases in the order of low-, medium- and high-level distortion test patterns. In particular, the trend of accuracy drop is markedly more pronounced and drastic in the low-

distortion training conditions than in the other conditions. Moreover, the average classification accuracy for the novel new distortions is notably lower in the high-distortion training condition than in the other conditions.

To confirm these observations, we conducted a 2x5 mixed-model ANOVA, using condition (low, medium, high, mixed) and item type (old, prototype, new-low, new-medium, new-high) as factors. The analysis revealed a significant main effect of item type, $F(3.16, 947.07) = 111.09$, $p < .001$, $\eta^2 = .270$; a significant main effect of learning condition, $F(3,300) = 19.95$, $p < .001$, $\eta^2 = .166$; and a significant interaction between the two factors, $F(9.47, 947.07) = 6.25$, $p < .001$, $\eta^2 = .059$. To assess the degree to which classification accuracy decreases with the distortion level of novel test patterns, we ran pearson correlation tests between the distortion level (proto, low, med, high coded as 1, 2, 3, 4) and the classification accuracy for each training condition. As observed in fig. 2, for all four condition, there was a significant negative correlation between the distortion level and the classification accuracy[1] ($r_{low} = -.53$, $r_{med} = -.25$, $r_{high} = -.19$, $r_{mix} = -.32$, all $p < .001$). We further computed the regression (β) coefficients for individual subjects measuring the gradients at which individual classification accuracy decreases with the distortion level. By comparing the subject-level gradients between training conditions, we found that the mean gradient in the low condition is significantly higher than that in the medium condition, $t(143.2) = 4.305$, $p < .001$. Lastly, the mean classification accuracy in the high condition is significantly lower than that in the medium condition, $t(150.7) = 4.024$, $p < .001$.

[1] The correlation analysis is based on the data pooled across all subjects in each training condition

Model-based accounts of the classification data

There has been a huge and influential previous literature that has used the dot-pattern prototype-distortion paradigm for investigating the nature of human category learning. However, rigorous formal modeling of classification responses based on the dot-pattern distortion paradigm suffers from a limitation that the psychological dimensions that compose the patterns are unknown. To compare the effectiveness of different methods of feature-space representation, we applied two traditional methods of characterizing between-pattern similarities to our modeling and devised an innovative deep-learning-based method in an attempt to address the limitations of the traditional methods.

In the first approach, we use the x-y coordinates of the 9 dots in each pattern as the assumed dimensions, and compute similarities between patterns based on distances between corresponding dots in each of the patterns. Since all three category prototypes are generated by placing the 9 dots in random positions, we need to specify the correspondences between the dots of patterns across categories. To do this, we computed the sum-squared distances between individual dots in two of the category prototypes under every possible way of alignment, and determined the alignment that yields the shortest distance to be the final solution for the subsequent distance computation. Likewise, we also defined the correspondences between the dots in the third category prototype and the ones in the first two based on the sum of the distances between the dots in the third prototype and those in the first two prototypes. Notably, such modeling approach has a major shortcoming in characterizing the psychological representations of the dot patterns: the configurations of dots in the patterns give rise to salient

emergent dimensions (e.g. spatial extent of the patterns, clustering of the dots, symmetry) that are not captured by the physical dot locations themselves.

In the second approach, we simply define free parameters representing the average similarity among various types of patterns, and substitute these parameter estimates into formal categorization models for predicting classification. Although the approach is a reasonable one, it too has a major limitation. Specifically, it fails to capture the variability across different tokens of the same types of patterns. For example, due to the random nature of the statistical-distortion algorithm used to generate dot patterns, some medium-level distortions may be extremely easy to classify, whereas others may be very difficult. Estimating an "average similarity" parameter across all the medium-level distortions fails to capture this form of individual-pattern variability.

Neural Network Training Procedure

As an innovative approach, we also used deep-learning techniques to extract feature representations for the dot patterns and used these feature representations as candidate inputs for competing cognitive models of classification. The goal of our deep learning procedure was to train convolutional neural network (CNN) that takes images of dot patterns as input and predict their category membership as output. To simulate human learning in the training condition with only low-distortion patterns, we generate as the input to the neural network 200 low-distortion patterns from each category prototype using the same dot-distortion scheme as used to generate the training patterns in the experiment. However, 600 dot patterns (200 per category) is quite small for a deep learning training set. By contrast, image-classification networks are often trained on millions of images spanning thousands of categories, and the large size of the training set is crucial for the networks to learn more robust and complex features than those trained on smaller image set. Therefore, we used a pre-trained implementation of ResNet18 as a starting point. To adapt the

network to our specific task, we replaced its output layer with a penultimate layer to encode the feature representations of the dot patterns and a classification layer that outputs the category predictions. The penultimate layer consists of 18 nodes[2] representing the latent features of dot patterns, which are fully connected to a 3-node layer with linear units. The linear activation values are then passed to a softmax function, which computes the probabilities by which the input pattern is predicted to be a member of each of the three categories.

The network was trained to minimize the discrepancy between the network's predicted category probabilities and the true category label. The discrepancy was measured by the cross-entropy loss, formulated as

$$Loss = -\sum_{k=1}^{3} t_k \ln p_k \quad (1)$$

where t_k is an indicator variable encoding the true category label that returns 1 if category k is the true category and 0 otherwise. p_k denotes the predicted probability of classifying the input pattern into category k.

Due to the huge number of trainable parameters in a neural network, it is prone to overfitting to noise and failing to generalize to new data. Therefore, we need to train the network by means of cross-validation to assess its generalization performance. To this end, we split the dot pattern images into two separate sets: training set and validation set. The network was first trained to

[2] We set the number of nodes of the penultimate layer as 18 so that the modeling results based on the activation vectors can be directly compared with those based on the dot coordinates.

minimize the error on the training set, and then the error on the validation set was computed to evaluate the generalization performance. Specifically, the training set was selected by randomly sampling 133 of the 200 dot patterns in each category, and the remaining 67 patterns were held out as the validation set. In total, there were 399 images in the training set and 201 images in the validation set. It turned out that the network trained to classify training set generalize well to the validation set, with validation accuracy over 98%.

The optimization algorithm used is adaptive moment estimation optimizer (ADAM), with default parameters except for the learning rate. Stochastic gradient descent was set with a low learning rate of 0.0001 and high momentum of 0.9. A batch size of 75 patterns was chosen for each training iteration, with the order of training patterns shuffled for each epoch of training.The training was stopped when the validation accuracy stopped increasing for at least 15 epochs, or after a maximum of 100 epochs.

Once the network parameters were fitted to classify the randomly generated low-distortion patterns, the full set of test patterns used in the experiment (including the training patterns and the prototypes) were fed into the network and the 18-node activation vector on the penultimate layer of the network can be extracted as the feature-space representations for each test pattern.

Formal Models of Categorization

Three versions of exemplar and prototype models each are fitted to the classification response data in the test phase using the three different methods of feature-space representation.

According to the exemplar model (i.e. GCM), the probability that a pattern *i* is classified into category A is found by summing its similarity to all the training examples *a* that belong to

category A, and dividing by the summed similarity of *i* to all the training examples of all categories:

$$\Pr(A|i) = \frac{(\sum_{a\in A} S_{ia})^{\gamma}}{(\sum_{a\in A} S_{ia})^{\gamma} + (\sum_{b\in B} S_{ib})^{\gamma} + (\sum_{c\in C} S_{ic})^{\gamma}} \quad (2)$$

Where the parameter γ is a response-scaling parameter. When γ grows larger in magnitude, the observer responds more deterministically with the category that yields the largest summed similarity.

For the average-similarity-based version, five free parameters were set to represent the average similarities between low-distortion training patterns to the prototype (S_p), low (S_l), medium (S_m) and high-level distortions (S_h) in the same category as well as to all the patterns in the contrasting categories (S_c). For the dot-coordinate-based and CNN-activation-based versions, the similarity between test pattern *i* and training example *j* (s_{ij}) was defined as an exponential-decay function of the psychological distance between the two patterns in the feature space:

$$s_{ij} = e^{-cd_{ij}} \quad (3)$$

where *c* is a sensitivity parameter that describes the rate at which similarity declines with distance. The sensitivity parameter provides a measure of overall discriminability among patterns in the feature space.

The distances between individual patterns for the two versions of exemplar models are derived from the x-y dot coordinates and the CNN activation vectors, respectively. In particular,

the standard Euclidean distance formula is used to compute the distance between test pattern *i* and training example *j*,

$$d_{ij} = \left[\sum_m w_m (x_{im} - x_{jm})^2\right]^{1/2} \quad (4)$$

where x_{im} and x_{jm} denotes the coordinate value on dimension *m* of the patterns *i* and *j* respectively. w_m represents the attention weight given to dimension m. To constrain the number of freely estimated parameters, the weights for all 18 dimensions (either of the dot-coordinate-based or the CNN-activation-based space) are all set to be equal to 1. In sum, there are two free parameters in the dot-coordinate and CNN-activation exemplar models: the sensitivity parameter c and the response-scaling parameter γ.

According to the prototype model, the probability that pattern i is classified into category A is given by

$$\Pr(A|i) = \frac{(S_{i,A})^{\gamma}}{(S_{i,A})^{\gamma} + (S_{i,B})^{\gamma} + (S_{i,C})^{\gamma}} \quad (5)$$

where $S_{i,A}$ is the similarity between the test pattern i to the prototype of category A. The response-scaling parameter γ serves analogous function as in eq. 2.

The average-similarity-based version includes four free parameter representing the average similarities between the category prototype to the low (S_l), medium (S_m) and high-level distortions (S_h) in the same category as well as to all the patterns in the contrasting categories (S_c), with the self-similarity between the prototype to itself set as 1. For the dot-coordinate-based and CNN-activation-based versions, the similarities between the test pattern and the category prototypes are derived from the corresponding dot coordinates and CNN activation vectors,

according to the formulae specified in eq. 3 & 4[3]. Note that the response-scaling parameter γ (as defined in eq. 5) cannot be estimated separately from the sensitivity parameter c (as defined in eq. 3) in the prototype models, so γ was fixed at 1 for all the prototype models. As such, there is only one free parameters in the dot-coordinate and CNN-activation prototype models: the sensitivity parameter c.

For each of the six models explained above, the individual model fits are evaluated by the negative log maximum likelihood (-lnL) of observing the classification responses of all test patterns given the model:

$$-lnL = -\sum_{i \in test} \sum_{k \in K} \mathrm{F}(k|i) * \ln[\Pr(k|i)] \quad (6)$$

where Pr(k|i) denotes the probability of classifying a test pattern i into category k as predicted by the model, with K = 3 representing the three category responses and i taken from the full set of test patterns in the experiment. F(k|i) denotes the absolute frequency with which item i is classified into category k. The particle swarm optimization algorithm coded in MATLAB is used to find the best-fitting parameters that minimize the –lnL for each model.

Because the alternative models have differing numbers of free parameters, new statistical metrics are needed to compare the individual model fits that correct for the number of free parameters. Therefore, we compared the model fits using AIC[4] and BIC[5] statistics in addition to

[3] In the distance and similarity formulae for prototype models, the training exemplar j in equations 3 and 4 are substituted with the category prototype K.

[4] AIC = -2ln(L) + 2P, where ln(L) is the maximum likelihood of the data, and P is the number of free parameters in the model.

[5] BIC = -2ln(L) + Pln(N), where ln(L) is the maximum likelihood of the data, P is the number of free parameters in the model, and N is the total number of observations in the data set.

the –lnL statistic. All the summary-fit statistics are the badness-of-fit measures, meaning that higher values indicate worse fit of the models. Notably, the AIC tends to favor relatively complex models with more free parameters, whereas the BIC tends to favor simpler models with less free parameters.

Model-fitting Results

Figure 3 shows the maximum-likelihood fits from the three versions of exemplar models to the classification performance in the test phase. In the test phase, observers classified 87 dot patterns into three different categories. Each point in each scatterplot indicates the probability that a particular item was classified into a particular category; thus, there are 261 points in each plot. For each item, the probability of correct classification is indicated with solid dots (color coded to represent the pattern types). In addition, for each item, the probabilities of the remaining two incorrect classifications are indicated with open dots. The y-axis shows the observed probabilities, whereas the x-axis shows the predicted ones. Likewise, figure 4 illustrates the model fits of the prototype models to the classification test performance. The summary-fit statistics from the three versions of exemplar models (upper panel) and prototype models (lower panel) are reported in the top rows of table 1. In addition, the best-fitting parameters for each of the six models are reported in table 2.

The summary-fit statistics reported in table 1 suggest that the feature-space representations based on the dot coordinates and CNN activation cannot provide a better account of the classification data than the traditional average-similarity representation. For both exemplar

and prototype models, the AIC and BIC statistics favor the average-similarity-based representation compared to the representations based on the dot coordinates and CNN activation, so both methods fail to adequately account for the classification performance at the level individual patterns. There is also no substantial advantage of models based on the CNN activation relative to those based on the dot coordinates. For the exemplar models, the CNN activation version yields only slightly better summary fits to the data than does the dot coordinate version. For the prototype model, however, the dot coordinate version yields slightly better fit than does the CNN activation version.

Multidimensional Scaling

Although the node activations extracted from neural network may encode the psychological representations of the dot patterns to some extent, the representations in a neural network are distributed across many nodes. Therefore, the activation values of individual nodes per se contain little information about the underlying psychological dimensions, and further analysis is needed to reduce the activation values to psychological feature vectors with lower dimensionality.

For this purpose, we applied a metric-scaling model to the between-pattern distances derived from the deep-learning activation values. First, we computed the standard Euclidean distances between the vectors of activation values associated with every pair of test pattern (as in Eq. 1). Second, we used mdscale function from MATLAB to conduct the metric scaling analysis[6]. The MDS program searches for the locations of the points each representing a test

[6] Metric scaling is used because we believe the Euclidean distances between deep-learning activation vectors are interval-scaled. In other words, we intend for the MDS solution to capture the numerical differences between the dissimilarities of dot patterns in addition to the mere ordering, which could be achieved by non-metric scaling. Nevertheless, the resulting MDS solution turned out to be virtually the same for both scaling options.

pattern in a multidimensional space so as to approximate a linear relation between the inter-point distance in the MDS space and the pairwise distances computed from the activation values. Thus, patterns that seem more similar tend to be located closer together in the space. The departure from a perfect linear relation is known as stress (Kruskal & Wish, 1978). As one increases the number of dimensions, one can reduce the stress, but at the expense of requiring a greater number of free coordinate parameters to achieve this fit.

The number of dimensions was varied from 1 through 10. Figure 5 shows a plot of stress against the number of dimensions assumed in the analysis. As can be seen, there is a drastic decrease in stress with increases in dimensionality from 1 to 2, and very little to no decreases in stress thereafter. Based on the value of stress alone, the minimum number of dimensions need to provide a good fit is 2. However, we decided to choose the three-dimensional MDS solution in order to derive more interpretable dimensions, which will be explained shortly.

After obtaining the MDS solution, the next step is to test if the dimensions of the MDS configuration have natural interpretations that correspond to important characteristics of the dot patterns. It is important to note that the MDS modeling analyses, with the inter-point distance conforming to a Euclidean metric, has the property of rotation-invariance, namely, any rigid rotation of the scaling solution will yield the same inter-point distances in the space. Therefore, the orientation of the MDS solution is arbitrary, so additional analyses are needed to address the issue of the interpretability of the derived dimensions.

To address the interpretability problem, we first need to establish a number of quantitative measures characterizing some of the most salient emergent properties of the dot patterns that subjects can rely on for classification decisions. Based on preliminary inspections of two-dimensional projections on the three-dimensional MDS solution, we hypothesized three

objective dimensions of dot pattern features whose values can be computed from the x-y coordinates of the composing dots. The first dimension is the extent to which the overall size of the pattern is wider than it is taller (fig. 6, left panel, x axis). Intuitively, it can be formally measured as the width-to-height ratio:

$$m1 = [\max(x_i) - \min(x_i)]/[\max(y_i) - \min(y_i)] \quad (7)$$

where x_i and y_i denote the x and y coordinate of the i^{th} out of the 9 dots, respectively

The second dimension represents the extent to which the pattern seems split in half horizontally (fig. 6, left panel, y axis). To measure it objectively, we first divided the pattern into two clusters by bisecting the range of the x-coordinates of all 9 dots and grouping the subset of dots in the same section into one cluster. Then, the degree of splitness is measured as the ratio of the average inter-point distance for dots within the same clusters and that for any pair of dots between the clusters. Formally,

$$dist\_between = \frac{1}{pair(i.j)} \sum_i \sum_{j:j>i} |xl_i - xr_j| \quad (8a)$$

$$dist\_within = \frac{1}{pair(i_1,i_2)+pair(j_1,j_2)} \left[\sum_{i_1} \sum_{i_2:i_2>i_1} |xl_{i_1} - xl_{i_2}| + \sum_{j_1} \sum_{j_2:j_2>j_1} |xr_{j_1} - xr_{j_2}| \right] \quad (8b)$$

$$m2 = dist\_between/dist\_within \quad (8c)$$

where xl_i denotes the i^{th} dot in the left cluster and xr_j denotes the j^{th} dot in the right cluster. pair(i,j), pair(i_1,i_2), pair(j_1,j_2) represents the number of all possible pairs of dots between the two clusters, and within the left and right clusters, respectively.

The third dimension measures the extent to which there are two distinctive dots lying in the center of a pattern (fig. 6, right panel, y axis). The centrality of individual dots are measured as the dot distances to the centroid, the ideal central location found by averaging over the coordinates of all nine dots. With the notion of centrality quantified, the dimension value is then defined as the ratio of the average distance of all nine dots to the centroid and that of the closest two dots to the centroid. Formally,

$$m3 = \frac{\frac{1}{9}\sum_{i=1}^{9}\sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2}}{\underset{i}{\operatorname{argmin}}\frac{1}{2}\sum_{i=1}^{2}\sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2}} \quad (9)$$

where $\bar{x}$ denotes the mean of the x coordinates of all 9 dots, and $\bar{y}$ denotes the mean of the y coordinates of all 9 dots.

Having defined the physical dimensions, we then ran a FORTRAN program which rotates, translates, and scales the derived MDS solution in an attempt to bring it into correspondence with the normative solutions consisting of individual-pattern measurements along the three explicitly defined feature dimensions. The "target" MDS solution (produced by rotation, translation and scaling) was defined to be the one that minimized the sum of squared deviations (SSD) between the x im values and the corresponding z im values across all test patterns and the three hypothesized dimensions, formalized as

$$SSE = \sum_i \sum_m (x_{im} - z_{im})^2 \quad (10)$$

Where xim and zim denote the coordinate value of the pattern i on dimension m, in the MDS solution and the normative solution respectively.

The correlations between the three dimensions of the rotated MDS solution and those of the normative solution were shown in Table 3. As can be seen, the second dimension of the MDS solution highly correlates with the horizontal splitness measures, but the first and third dimensions are less strongly correlated with the measures of width-to-height and the dual centrality. Notably, these correlations were computed with respect to a single rigid rotation of all three axes of the original MDS solution. However, even though rotating the solution separately for individual dimensions slightly increases the correlations, the general trend still stays the same for individual-dimension fits.

Exemplar model fitting

Using the rotated MDS solution as the feature representations, we first fitted two versions of the GCM to the response data in the test phase. As can be seen from the rotated MDS solution (fig. 7), some dimensions are more diagnostic of the category membership of the test patterns than others. Therefore, it stands to reason that assigning freely-estimated attention weights to different dimensions in the MDS-derived feature representations could largely improve the model fit to the observed responses. As such, we fitted a baseline version of GCM assuming equal attention weights to the three dimensions and another version with freely estimated dimensional weights. The other aspects of GCM models were formulated in the same way as for the GCM model with raw activation representations (as specified in equations 2-4). The scatter plots in figure 8 compare the observed category response probabilities of individual test patterns with the corresponding predictions yielded by the two GCM models. Again, the solid dots indicate correct classification responses.

Comparison of the summary fits (table 1, exemplar models) reveals that applying MDS techniques and dimension weighting to CNN activation patterns substantially improves the fit to the classification data. Although the unweighted MDS exemplar model performs slightly more poorly than does the average-similarity exemplar model, the weighted version yield noticeably better summary fits to the data than does the average-similarity model.

Prototype model fitting

We also fitted four versions of the prototype models using the same MDS-derived feature representations. As can be seen from the rotated MDS solution (fig. 7, left panel), the category prototypes tend to lie at extreme, peripheral regions of the category distributions, which is in contrast to the general intuition that the category prototypes should be centrally located among the training exemplars of the same category. In other words, the prototypes as represented by the MDS coordinates are more like exaggerated, ideal characterization of the training exemplars in the same category than an average representation thereof. To explore the effects of the alternative notions of prototypes on the model fitting performance, we defined the feature-space representation of the prototypes in two methods to capture both notions. We called the first method ideal-point version where the prototype representations are defined to be their extreme, ideal-point MDS coordinates, and the second were labeled central-tendency version where the prototype representations are computed by averaging across the MDS coordinates for each of the training exemplars of the corresponding category. For each of the two methods of the feature-space representations, we fitted two versions of the prototype models: one with equal dimension weights, and another with freely estimated dimension weights. The scatter plots in figure 9 show the observed and predicted probabilities of category responses for individual test patterns. In

addition, the summary-fit statistics from the four prototype models are reported in the bottom rows of table 1, and the best-fitting parameters for the two models are reported in table 4.

The summary fits of prototype models reported in table 1 show similar patterns as the ones of exemplar model. First, both central-tendency prototype models are chosen over the respective ideal-point prototype models using the AIC and BIC statistics, indicating that the central tendency is the more valid prototype representation in terms of prototype model fits. As with the exemplar model fits, the weighted MDS central-prototype model outperforms the unweighted counterpart in fitting the data. Moreover, the weighted MDS prototype model even yields better summary fits than do both the average-similarity prototype and exemplar models. It is also noteworthy that there is little difference in the summary fits of the weighted MDS exemplar model and the weighted MDS central-prototype model. The reason may be that in the low-distortion training condition, the similarity of a test pattern to the prototype of each category is roughly the same as the similarity of a test pattern to the low-distortion training examples.

Table 1. Summary Fits of Models to the Classification Test Data

Exemplar Models

Model	-lnL	AIC	BIC	P
Average Similarity	1159.7	2331.4	2370.8	6
Dot Coordinates	1227.5	2459.0	2471.1	2
CNN Activation	1221.7	2447.4	2460.5	2
CNN_MDS_unweighted	1178.8	2361.6	2374.7	2
CNN_MDS_weighted	1119.3	2246.6	2272.8	4

Prototype Models

Model	-lnL	AIC	BIC	P
Average Similarity	1159.9	2329.8	2354.0	4
Dot Coordinates	1195.3	2392.6	2399.2	1
CNN Activation	1272.0	2546.0	2552.6	1
CNN_MDS_unweighted_extreme	1272.5	2547.0	2553.6	1
CNN_MDS_weighted_extreme	1186.4	2378.8	2398.5	3
CNN_MDS_unweighted_central	1179.3	2360.6	2367.2	1
CNN_MDS_weighted_central	1123.5	2253.0	2272.7	3

Table 2. Best-Fitting Free Parameters of Models with Alternative Methods of Stimulus Representation.

Exemplar models

Parameter	Average Similarity	Dot Coordinates	CNN Activation
S_p	1.000	--	--
S_1	0.523	--	--
S_m	0.370	--	--
S_h	0.277	--	--
S_c	0.204	--	--
c	--	0.170	0.212
γ	2.906	1.000	1.000

Prototype models

Parameter	Average Similarity	Dot Coordinates	CNN Activation
S_p	1.000*	--	--
S_1	0.208	--	--
S_m	0.062	--	--
S_h	0.026	--	--
S_c	0.011	--	--
c	--	0.166	0.172
γ	1.000	1.000**	1.000**

* S_p is set to 1 for self-match similarity
** γ is fixed at 1 as it cannot be estimated separately from c

Table 3. correlation coefficients between the corresponding dimensions of the rotated MDS solution and the normative solution.

Dimension	Correlation
1.height-to-width ratio	0.373
2.horizontal splitness	0.802
3.dual centrality	0.600

Table 4. Best-Fitting Free Parameters of Models based on the Rotated CNN-MDS Solution

Exemplar Models

Parameter	GCM_unweighted	GCM_weighted
w_1	--	0.084
w_2	--	0.669
$*w_3$	--	0.247
c	0.221	0.421
γ	1.000	1.000

Prototype Models

Parameter	PM central unweighted	PM extreme unweighted	PM central weighted	PM extreme weighted
w_1	--	--	0.091	0.000

w_2	--	--	0.652	0.636
$*w_3$	--	--	0.257	0.364
c	0.213	0.173	0.402	0.341
γ	1.000	1.000	1.000	1.000

$*w_3 = 1 - w_1 - w_2$

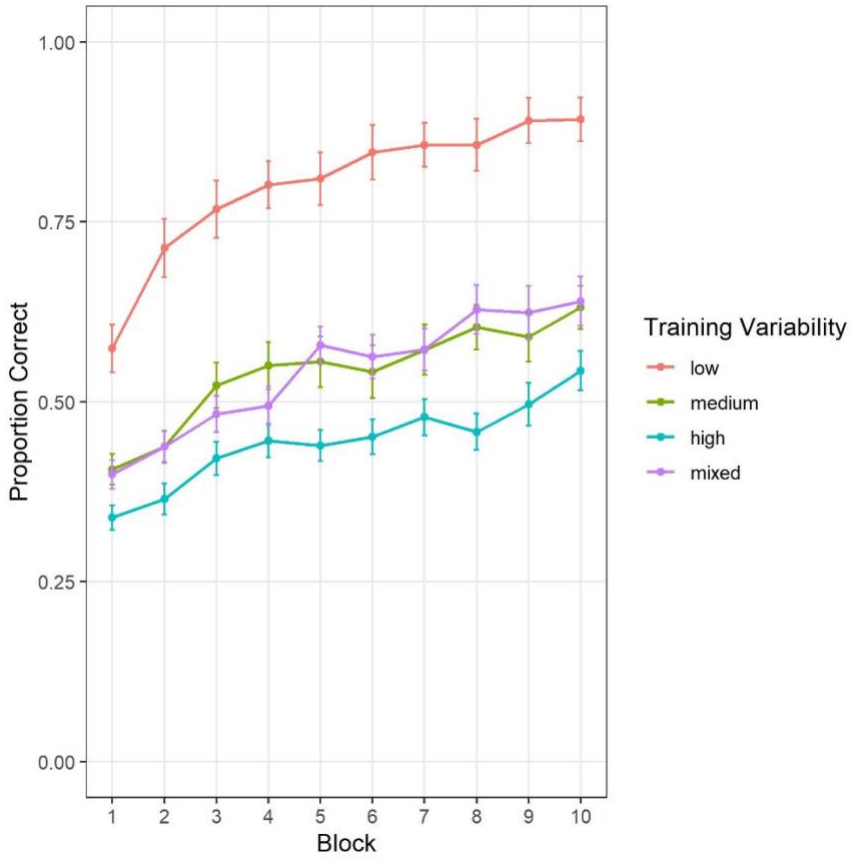

Figure 1

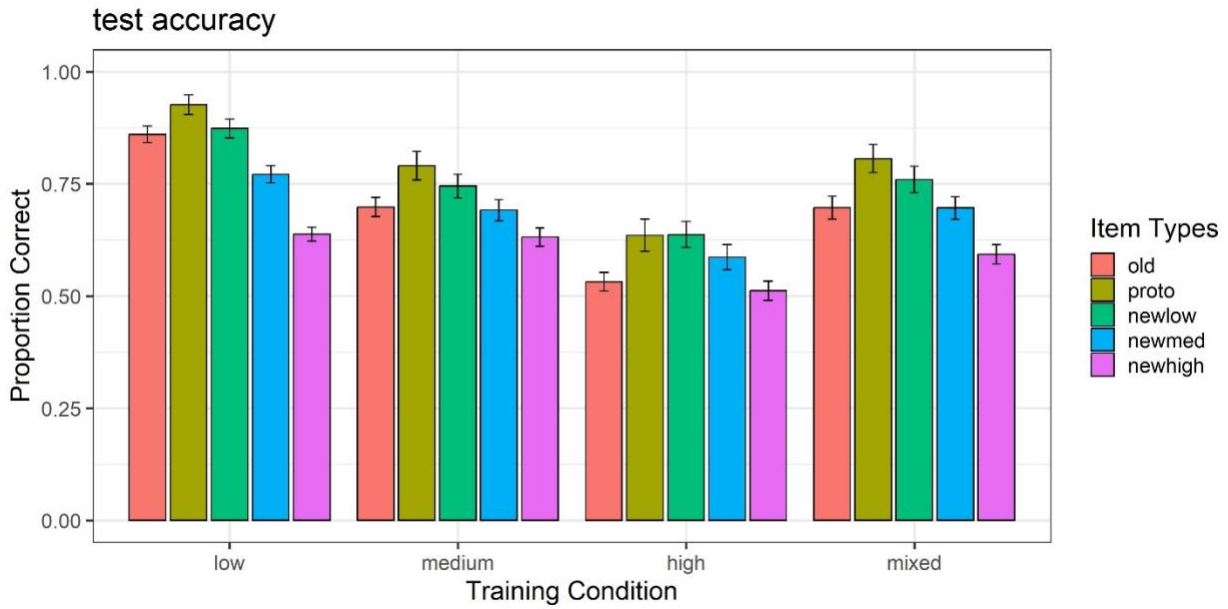

Figure 2

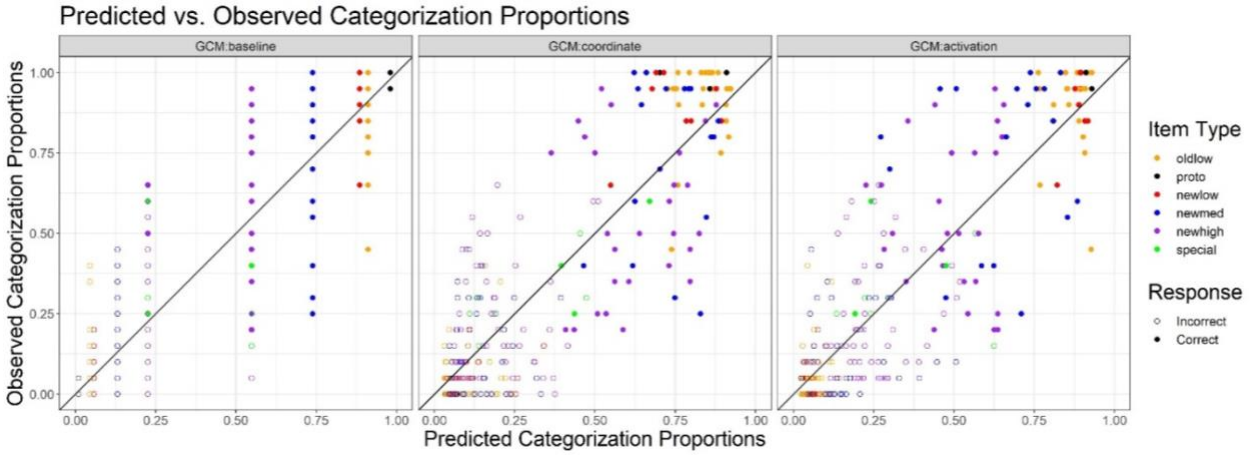

Figure 3

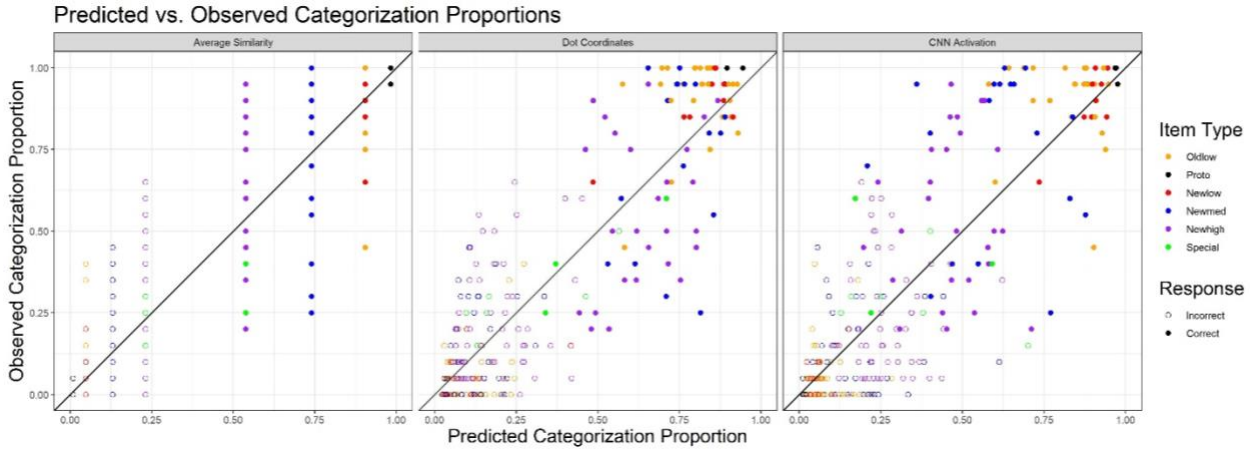

Figure 4

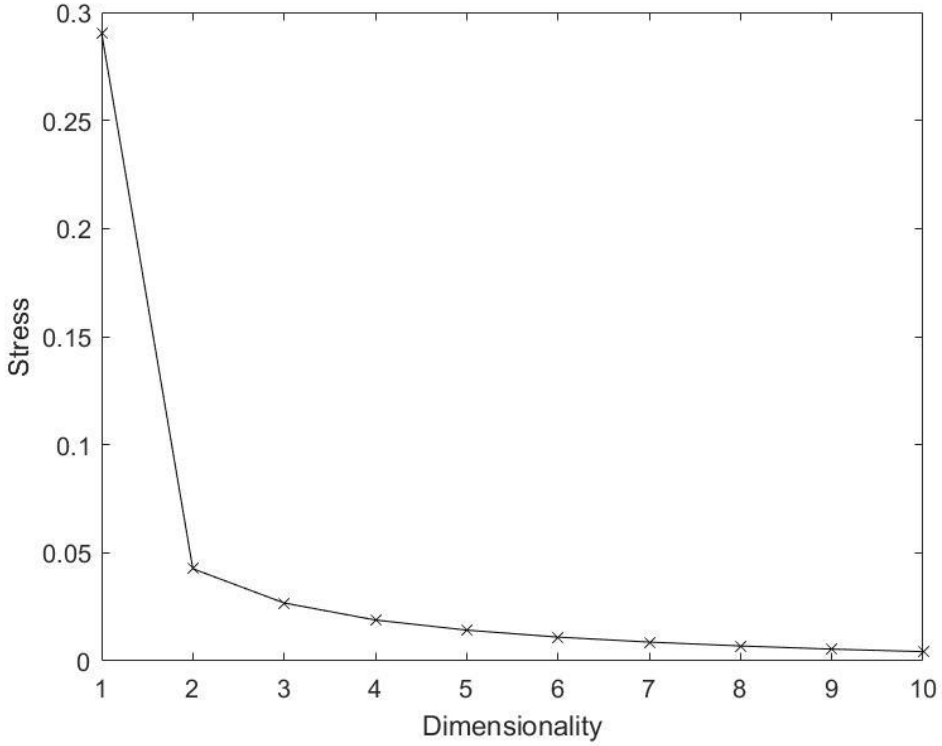

Figure 5

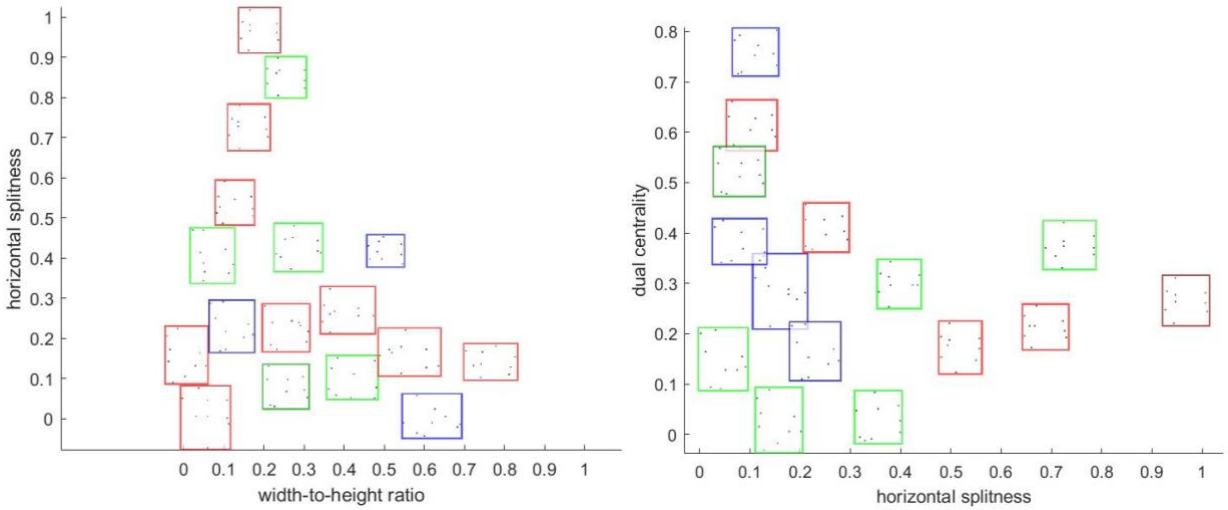

Figure 6

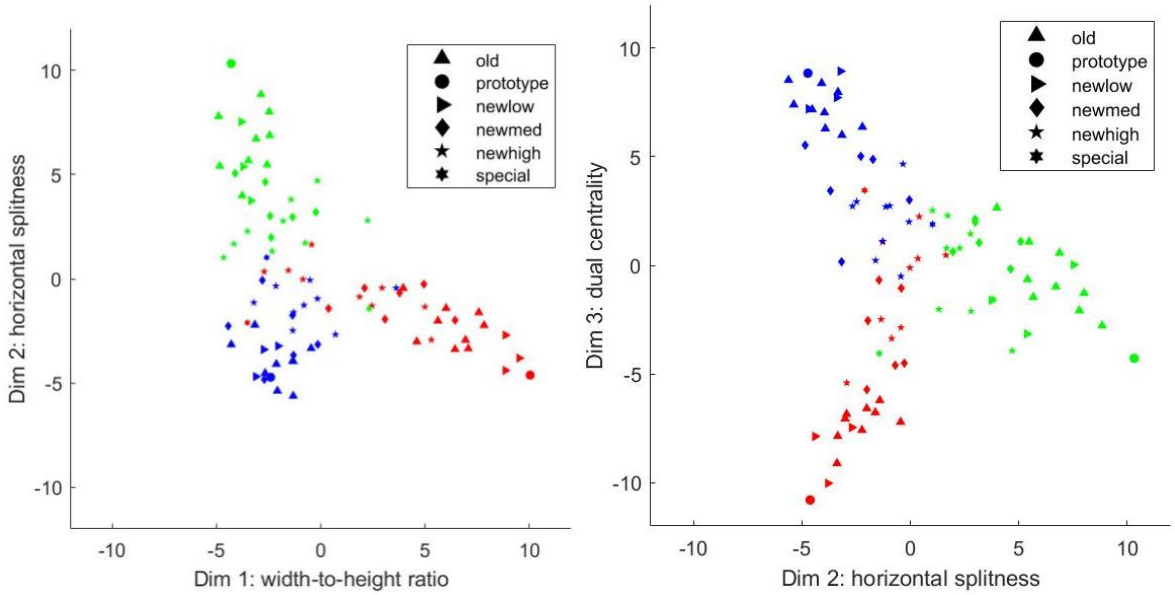

Figure 7. MDS configurations for the test patterns. The different colors indicate the category membership of individual test patterns. The different shapes represent different item types of the patterns. The left panel shows the coordinate values of test patterns on dimensions 1 and 2, and the right panel shows the values on dimensions 2 and 3.

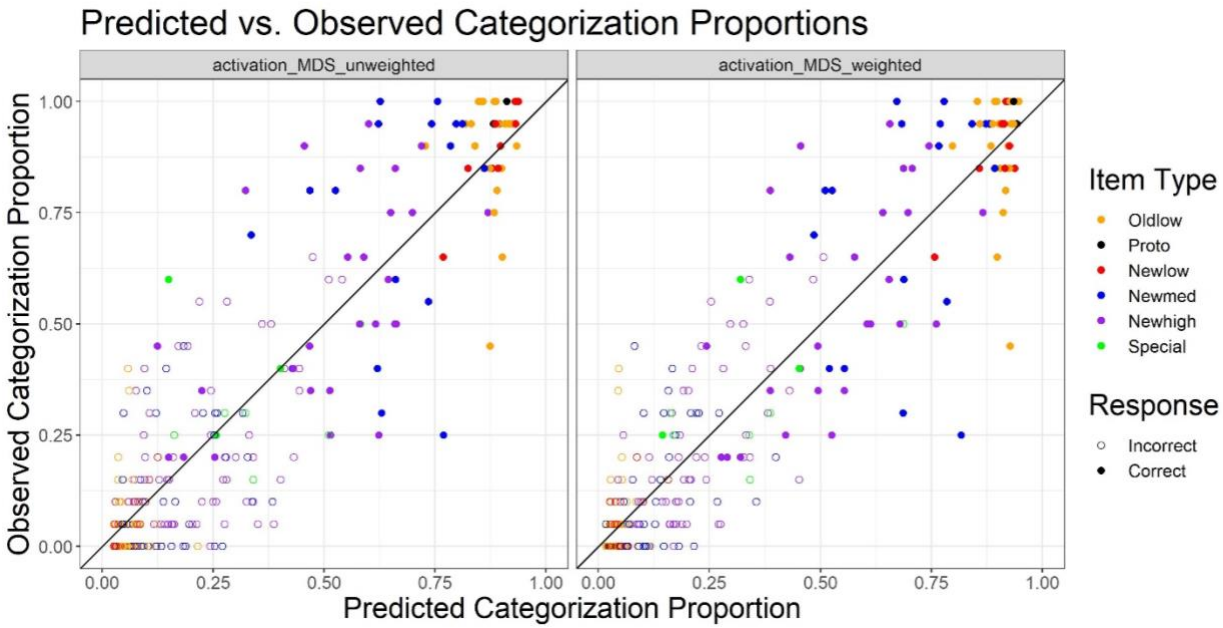

Figure 8

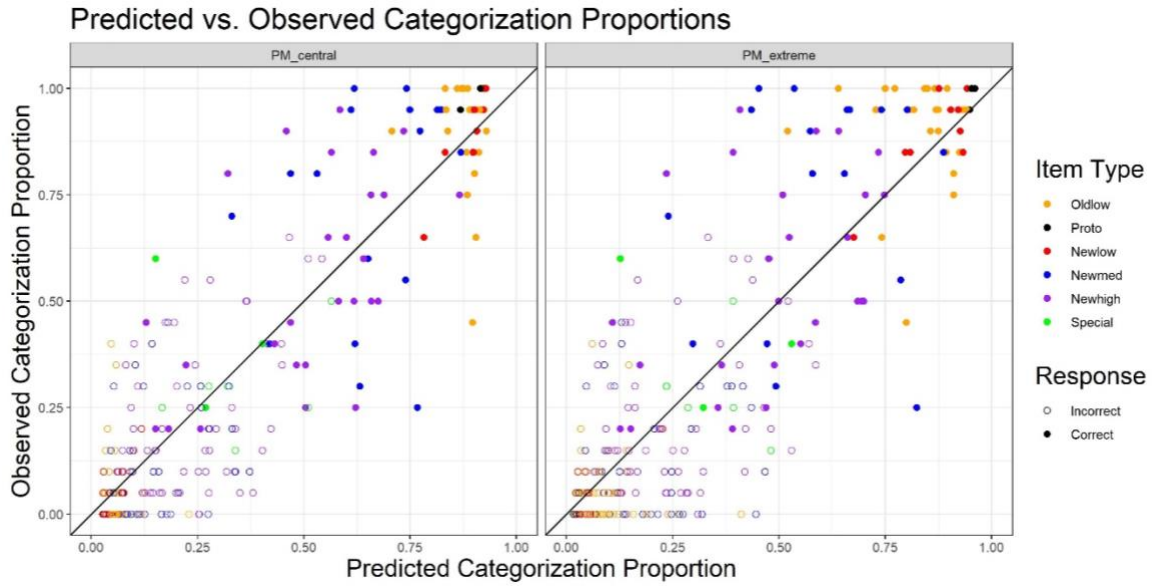

Figure 9 (unweighted)

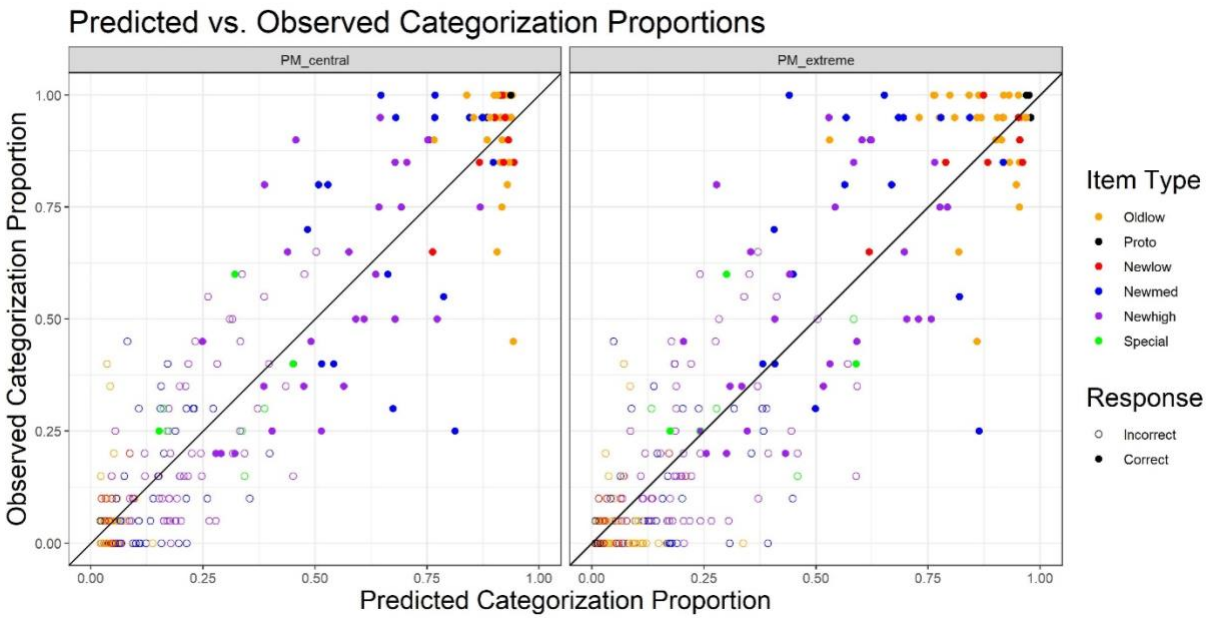

Figure 10 (weighted)