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 additionalhigh-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, and different sets of novel distortions were also randomly generated for each 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 at 55%. 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. Moreover, the trend of accuracy drop is markedly more pronounced and drastic for the training conditions with low- and mixed-level distortions than the medium- and high-distortion conditions. Using formal models of categorization, we attempt to model the classification performance in the test phase at the level of individual test patterns. Since the subjects in the low-distortion condition show substantially higher classification accuracy at the end of training and for the old items in the test phase compared to other conditions, we decided to focus the modeling analysis on the low-distortion 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 to use 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[[1]](#footnote-1) 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

(1)

where tk is an indicator variable encoding the true category label that returns 1 if category k is the true category and 0 otherwise. pk 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 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 actual test patterns in the experiment 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

For the dot-coordinate-based and deep-learning-based methods, the distances between individual patterns are derived from the x-y dot coordinates and the penultimate-layer node activations, respectively. In particular, the standard Euclidean distance formula is used to compute the distance between test pattern *i* and training example *j*,

(2)

where and denotes the coordinate value on dimension *m* of the stimuli *i* and *j* respectively. represents the attention weight given to dimension m. To constrain the number of freely estimated parameters, the weights for all 18 dimensions are all set to be equal to 1.

The similarity between test pattern *i* and training example *j* (*sij*) is an exponential-decay function of this distance,

(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.

After defining the between-pattern similarities, we formalized two different mechanisms by which the similarity information can be used for making classification decisions. 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:

(4)

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.

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

(5)

where is the similarity of item i to the prototype of category A (computed using equations analogous to Eqs. 1 and 2 above). The feature representations of the prototypes will be explained in detail in the prototype model fitting section.

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. Both average-similarity-based models share five free parameters: the average similarity parameters between low-distortion training patterns to four different types of test patterns (Sl ,Sm, Sh, Sc) and the response-scaling parameter γ. An additional parameter representing the average similarity between the training patterns and the prototype (Sp) is included in the exemplar model, while the similarity of the prototype to itself is fixed to 1 for the prototype model. The coordinate-based and deep-learning-based models (both exemplar and prototype models) have only two parameters: the sensitivity parameter c and the response-scaling parameter γ. The model fits are evaluated by the negative log maximum likelihood (-lnL) of the data given the model. The model that minimizes -lnL is viewed as providing the best fit. We used the particle swarm optimization algorithm as implemented in MATLAB to locate the best-fitting 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 table 2. Because the alternative models have differing numbers of free parameters, statistical metrics are needed to evaluate the comparative model fits that correct for the number of free parameters. Therefore, we compared the model fits using AIC[[2]](#footnote-2) and BIC[[3]](#footnote-3) statistics in addition to the –lnL statistic. All the summary-fit statistics are the badness-of-fit measures, meaning that higher values indicates 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. In addition, the best-fitting parameters for each of the six models are reported in table 3.

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[[4]](#footnote-4) 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. The MDS program searches for the locations of the points each representing a test 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](https://link.springer.com/article/10.3758/s13428-017-0884-8" \l "Fig4) 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 ensure that 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:

(6)

where xi and yidenote the x and y coordinate of the ith 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,

(7a)

(7b)

(7c)

where xli denotes the ith dot in the left cluster and xrj denotes the jth dot in the right cluster. pair(i,j), pair(i1,i2), pair(j1,j2) 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,

(8)

where denotes the mean of the x coordinates of all 9 dots, and denotes the mean of the y coordinates of all 9 dots.

Having defined the physical dimensions, we ran the ROTAT program implemented in FORTRAN, 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

(9)

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 1. As can be seen, the second dimension of the MDS solution highly correlates with the horizontal splitness measures, and the first and third dimensions are not 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.

Prototype model fitting

We also fitted two 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 proposed two prototype models with the feature-space representation of the prototypes defined to capture either notions. One model is named extreme prototype model where the prototype representations are defined to be their extreme, ideal-point MDS coordinates, and the other is named central prototype model where the prototype representations are computed by averaging across the MDS coordinates for each of the training exemplars of the corresponding category. For the sake of comparing the prototype models with the best-fitting exemplar model, we set the dimension weights to be free parameters in the prototype models. 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 two prototype models are reported in table 4, and the best-fitting parameters for the two models are reported in table 5.

Table 1. correlation coefficients between the corresponding dimensions of the rotated MDS solution and the normative solution.

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

Table 2. Summary Fits of Models with Alternative Methods of Stimulus Representation to the Classification Test Data.

**Exemplar models**

Model -lnL AIC BIC P

Average Similarity 1159.7 2331.4 2352.8 6

Dot Coordinates 1227.5 2459.0 2466.1 2

CNN Activation 1221.7 2447.4 2454.5 2

**Prototype models**

Model -lnL AIC BIC P

Average Similarity 1159.8 2329.6 2347.4 5

Dot Coordinates 1195.3 2394.6 2401.7 2

CNN Activation 1272.0 2548.0 2555.1 2

Table 3. Best-Fitting Free Parameters of Models with Alternative Methods of Stimulus Representation.

**Exemplar models**

Parameter Average Similarity Dot Coordinates CNN Activation

*Sp*  1.000 -- --

*Sl* 0.523 -- --

*Sm* 0.370 -- --

*Sh* 0.277 -- --

*Sc* 0.204 -- --

*c* -- 0.170 0.212

*γ* 2.906 1.000 1.000

**Prototype models**

Parameter Average Similarity Dot Coordinates CNN Activation

*Sp*  1.000\* -- --

*Sl* 0.406 -- --

*Sm* 0.220 -- --

*Sh* 0.140 -- --

*Sc* 0.091 -- --

*c* -- 0.153 0.172

*γ* 1.967 1.088 1.001

\*Sp is set to 1 for self-match similarity

Table 4. Summary Fits of Models based on the Rotated CNN-MDS Solution to the Classification Test Data.

**Exemplar Models**

Model -lnL AIC BIC P

GCM\_unweighted 1178.8 2361.6 2368.7 2

GCM\_weighted 1119.3 2246.6 2260.9 4

**Prototype Models\***

Model -lnL AIC BIC P

PM\_central 1123.5 2255.0 2269.3 4

PM\_extreme 1186.4 2380.8 2395.1 4

\*Both prototype models have free dimension weights

Table 5. Best-Fitting Free Parameters of Models based on the Rotated CNN-MDS Solution

**Exemplar Models**

Parameter GCM\_unweighted GCM\_weighted

*w1*  -- 0.084

*w2* -- 0.669

*\*w3* -- 0.247

*c* 0.221 0.421

*γ* 1.000 1.000

**Prototype Models**

Parameter PM\_central PM\_extreme

*w1*  0.091 0.000

*w2* 0.652 0.636

*\*w3* 0.257 0.364

*c* 0.402 0.333

*γ* 1.000 1.022

*\*w3 = 1 - w1 - w2*

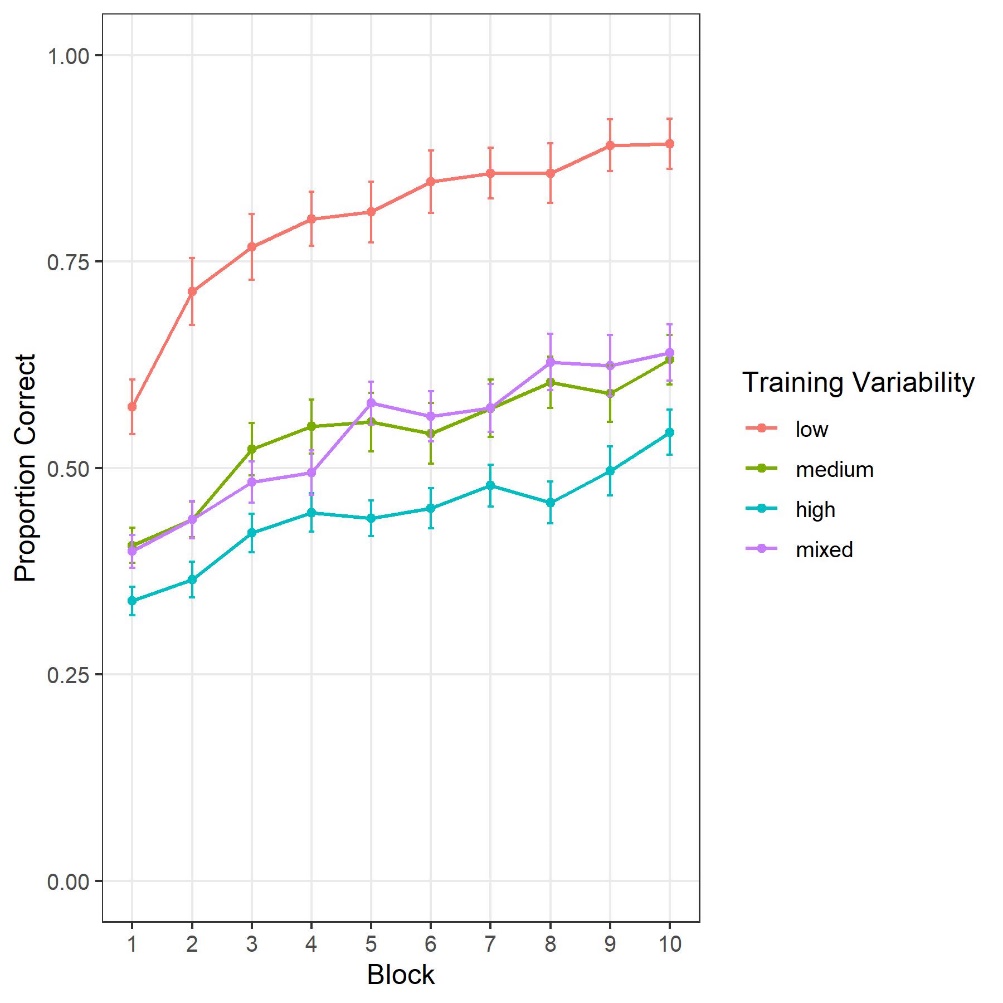


Figure 1

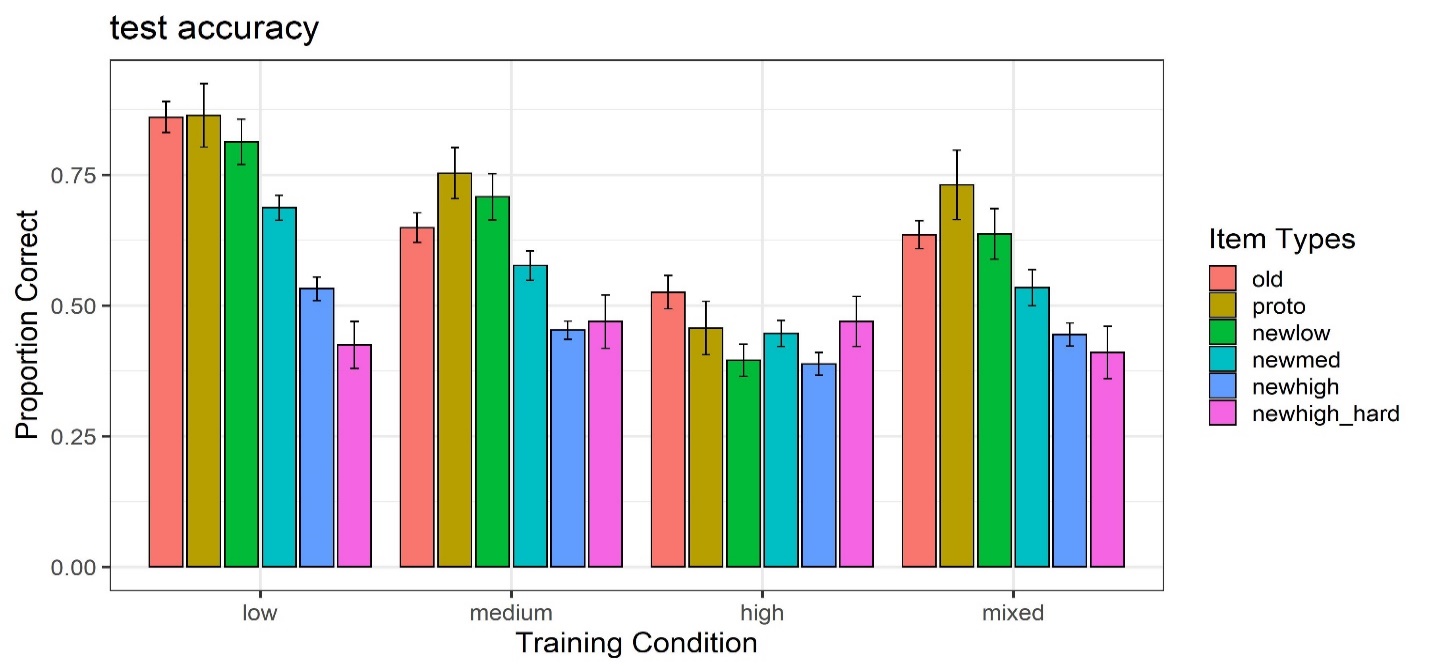
Figure 2

Figure 3

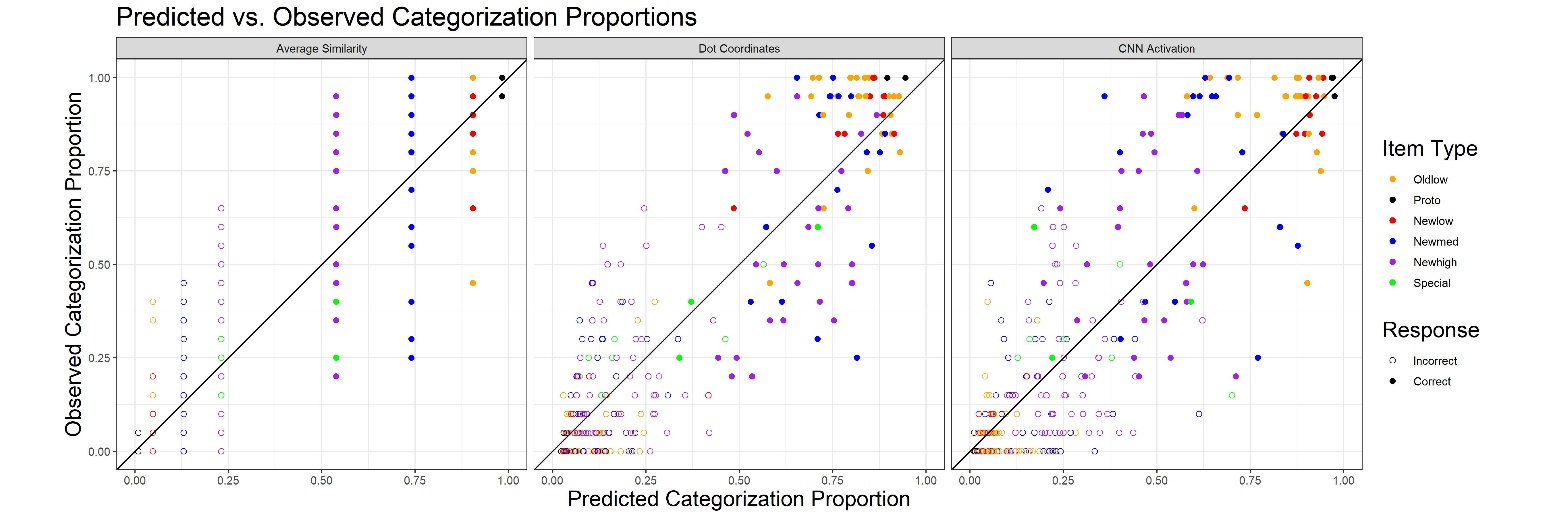


Figure 4

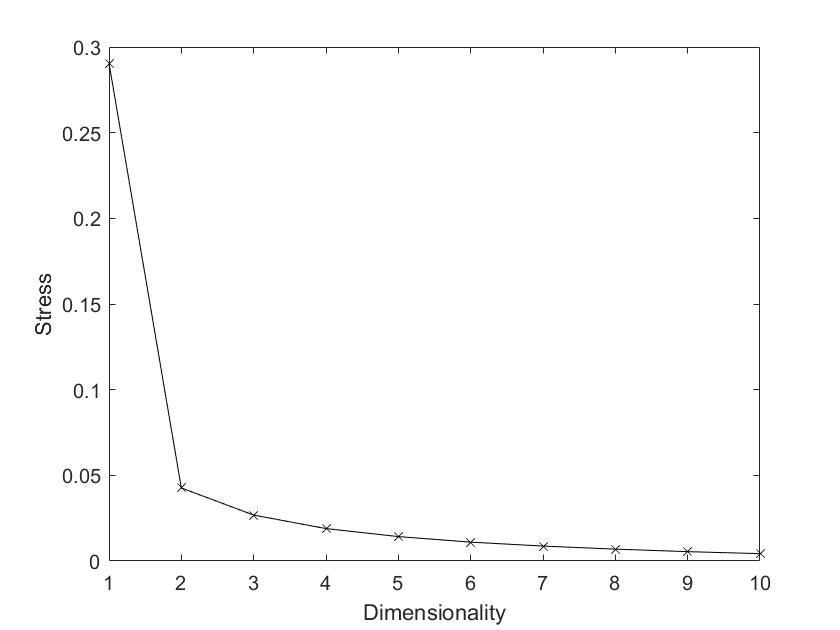


Figure 5

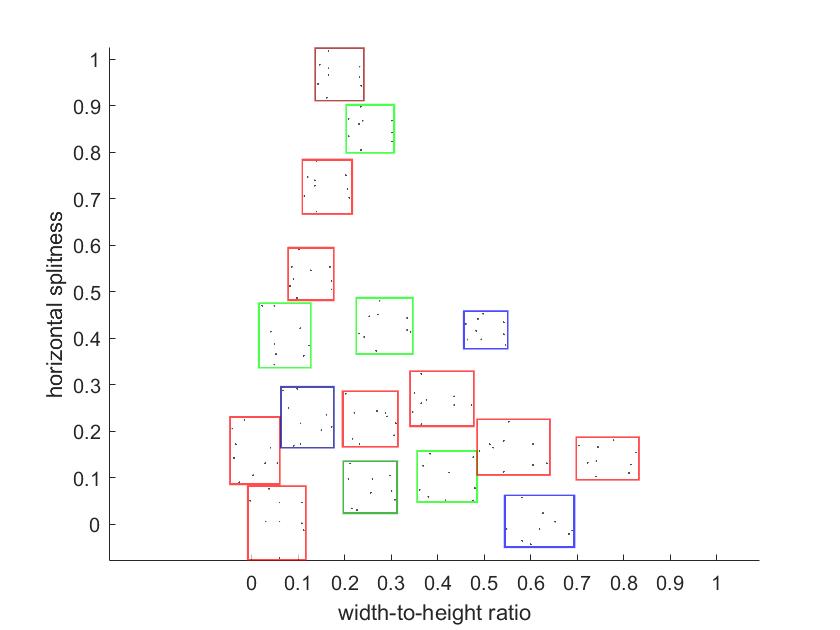


Figure 6



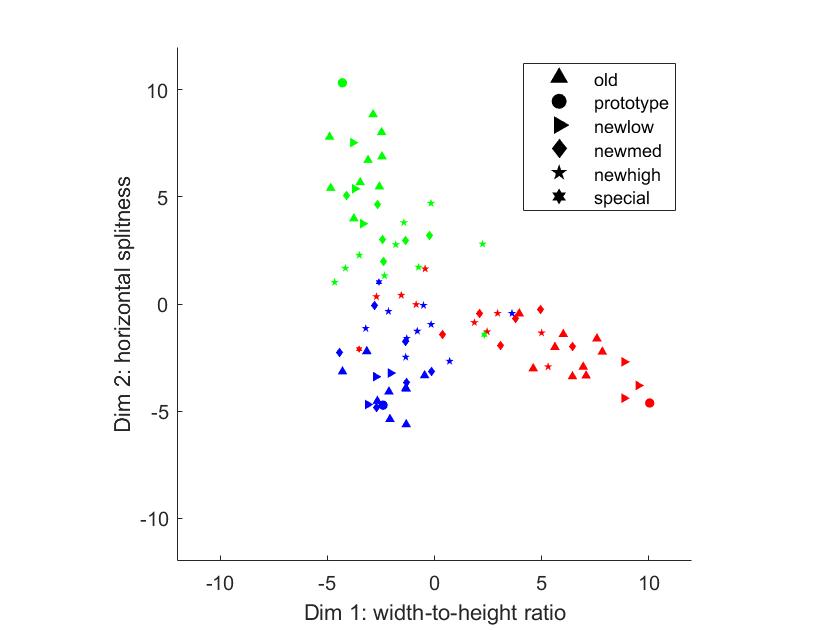


Figure 7 MDS configurations for the test patterns. The different colors indicate the category membership of individual test patterns. The different shapes represent different item types of the patterns. The left panel shows the coordinate values of test patterns on dimensions 1 and 2, and the right panel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

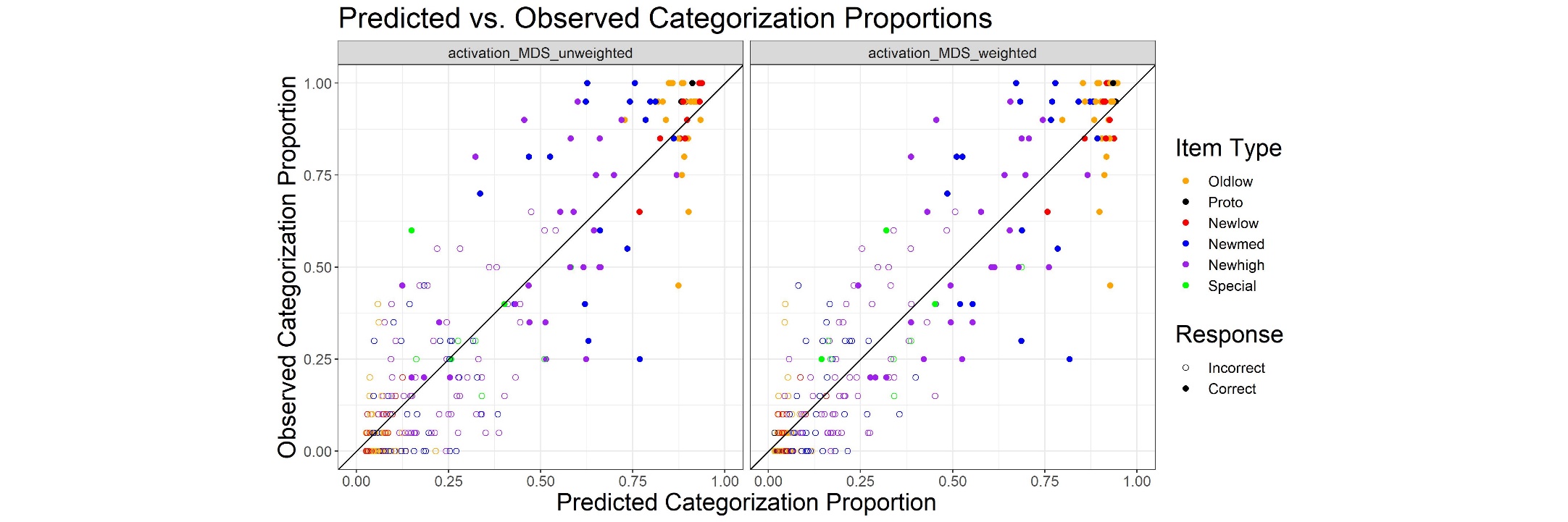


Figure 8

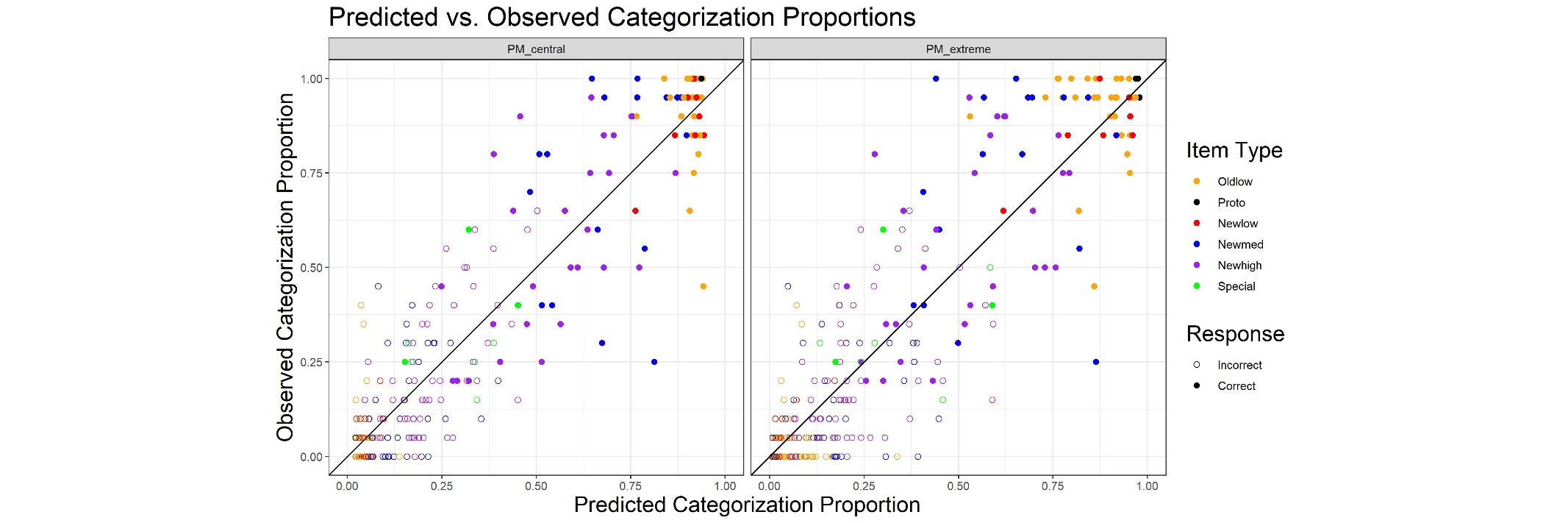


Figure 9

1. We set the number of nodes of the penultimate layer to be 18 so that the modeling results based on the activation vectors can be directly compared with those based on the dot coordinates. [↑](#footnote-ref-1)
2. AIC = -2ln(*L*) + 2*P*,where ln(L) is the maximum likelihood of the data, and P is the number of free parameters in the model. [↑](#footnote-ref-2)
3. BIC = -2ln(*L*) + *P*ln(*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. [↑](#footnote-ref-3)
4. Metric-scaling option is applied instead of non-metric scaling as the metric scaling preserves the numerical differences of the Euclidean distances between deep-learning activation vectors in the MDS solution. Regardless, the resulting MDS solution turns out to be virtually the same for both scaling options. [↑](#footnote-ref-4)