

Evaluating ‘Graphical Perception’ with CNNs

Daniel Haehn, James Tompkin, and Hanspeter Pfister

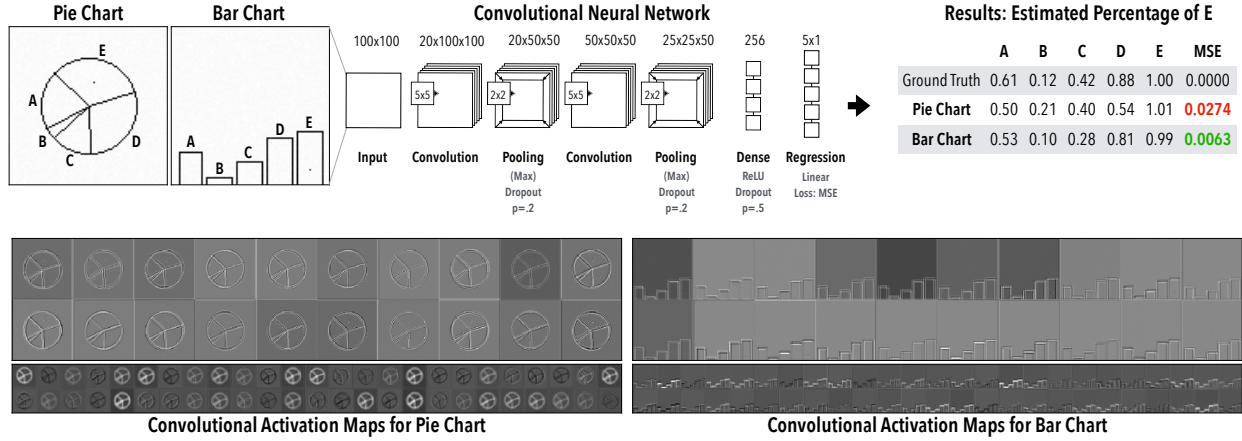


Fig. 1: **Computing Cleveland and McGill’s Position-Angle Experiment using Convolutional Neural Networks.** We replicate the original experiment by asking visual cortex inspired machine learning classifiers to assess the relationship between values encoded in pie charts and bar charts. Similar to the findings of Cleveland and McGill [7], our experiments show that CNNs read quantities more accurately from bar charts (mean squared error, MSE in green).

Abstract— Convolutional neural networks can successfully perform many computer vision tasks on images, and their learned representations are often said to mimic the early layers of the visual cortex. But can CNNs understand graphical perception for visualization? We investigate this question by reproducing Cleveland and McGill’s seminal 1984 experiments, which measured human perception efficiency of different visual encodings and defined elementary perceptual tasks for visualization. We measure the graphical perceptual capabilities of four classifiers on a) elementary perceptual tasks with increasing parametric complexity, b) the position-angle experiment that compares pie charts to bar charts, c) the position-length experiment that compares grouped and divided bar charts, and d) the bars and framed rectangles experiment where visual cues aid perception. We also study how feed-forward neural networks obey Weber’s law, which defines the proportional relation between perceivable information and distribution density. We present the results of these experiments to foster the understanding of how CNN classifiers succeed and fail when applied to data visualizations.

Index Terms—Machine Perception, Deep Learning

1 INTRODUCTION

Convolutional neural networks (CNNs) have been successfully applied to a wide range of visual tasks, most famously to natural image object recognition [31, 32], for which some claim equivalent or better than human performance. This performance comparison is often motivated by the idea that CNNs model or reproduce the early layers of the visual cortex, even though they do not incorporate many details of biological neural networks or model higher-level abstract or symbolic reasoning [13, 24, 37]. While CNN techniques were originally inspired by neuroscientific discoveries, recent advances in processing larger datasets with deeper networks have been the direct results of engineering efforts. Throughout this significant advancement, researchers have aimed to understand why and how CNNs produce such high performance [30], with recent works targeting the systematic evaluation of the visual perception limits of CNNs [17, 28].

One fundamental application of human vision is to understand data

visualizations. This is a task unlike natural image processing but which includes the abstraction of real-world objects and their effects into data, represented with visual marks. As a field, visualization catalogues and evaluates human perception of these marks, such as in the seminal *graphical perception* experiments of Cleveland and McGill [7]. This work describes nine elementary perceptual reasoning tasks, such as position relative to a scale, length, angle, area, and shading density, plus orders their reasoning difficulty. But, with increasing research interest in the machine analysis of graphs, charts, and visual encodings [10, 27], it seems pertinent to question whether CNNs are able to process these basic graphical elements and derive useful measurements from the building blocks of information visualization.

In this paper, we reproduce Cleveland and McGill’s human perceptual experiments with CNNs, and discuss to what extent they have ‘graphical perception’. To perform this evaluation, we parametrize the elementary perceptual tasks and experiments suggested by Cleveland and McGill [7], and define a set of regression tasks to estimate continuous variables. We pit four neural networks against human perception: a three-layer multilayer perceptron (MLP), the LeNet 2-layer CNN [23], the VGG 16-layer CNN [31], and the Xception 36-layer CNN [6]. As CNNs trained on natural images are said to mimic layers of the human visual cortex, we investigate whether using weights trained on natural images (via ImageNet [20]) or weights trained from scratch on elementary graphical perception tasks produces more accurate measurements and greater generalization.

We test these four networks across four scenarios presented by Cleve-

- Daniel Haehn, and Hanspeter Pfister are with Harvard University.
E-mail: {haehn,pfister}@seas.harvard.edu.
- James Tompkin is with Brown University.
E-mail: james_tompkin@brown.edu.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

land and McGill [7]: 1) Nine elementary perceptual tasks with increasing parametric complexity, e.g., length estimation with fixed x , then with varying x and y , then with varying width, including cross-network evaluations testing the generalizability of networks to unseen parameters; 2) The position-angle experiment, which compares judgments of bar charts to pie charts; 3) The position-length experiment, which compares grouped and divided bar charts; and 4) The bars and framed rectangles experiments, where visual cues aid ratio perception. We also investigate whether our CNNs can detect a proportional change in a measurement across scales in relation to Weber’s law.

With these experiments, we describe a ranking defining the ease with which our tested CNN architectures can estimate elementary perceptual tasks, equivalent to Cleveland and McGill’s ranking for human perception. Further, we discuss the implications of our results and derive recommendations for the use of CNNs in perceiving visualizations. We accompany this paper with open source code and our input and results data, both to enable reproduction studies and to spur new machine perception systems more adept at graphical perception: <http://rhoana.org/perception>

2 PREVIOUS WORK

Graphical Perception. Cleveland and McGill [7, 8] coin the phrase *graphical perception* to describe how different visual attributes and encodings are perceived by humans. They define *elementary perceptual tasks* as mental-visual stimuli to understand encodings in visualizations, and declare a ranking based on their perceptual difficulty. From these definitions, the authors propose and perform different experiments such as the *position-angle* experiment that compares bar charts and pie charts, the *position-length* experiment where users judge relations between encoded values in grouped and divided bar charts, and the *bars-and-framed-rectangles* experiment to evaluate Weber’s law [12] using the proportional relation between an initial distribution density and perceivable change.

Heer and Bostock later reproduced the Cleveland-McGill experiments via crowd-sourcing on Mechanical Turk [14], with similar results. Harrison *et al.* [11] repeated the Cleveland-McGill experiments while observing viewer emotional states, again with similar results. Our experimental setup again reproduces the Cleveland-McGill experiments, but instead of judging human perception, we judge machine perception using convolutional neural networks. While we focus on Cleveland and McGill’s work from the mid 1980s due to its repeated reproduction, many other works also investigate human perception to visual encoding [2, 5, 25, 26, 33–35].

Computational Visualization Understanding. Pineo *et al.* [27] create a computational model of human vision based on neural networks. Their simulations show that understanding visualization triggers neural activity in high-level areas of cognition, with the authors suspecting that this activity is supported by low-level neurons performing elementary perceptual tasks. Other work tries to parse infographics by finding higher-level saliency models [4], or by extracting text or key visual elements from visualizations using computer vision techniques [3, 10, 19]. However, these works do not investigate computational understanding of elementary perceptual tasks such as curvature, lengths, or position, which are the building blocks of visualization.

Visual-cortex-inspired Machine Learning. The human visual cortex allows us to recognize objects in the world seemingly without effort. This visual system is organized in layers, which inspired computational classifiers based on multilayer neural networks. Fukushima and Miyake developed the early Neocognitron quantitative model [9], which ultimately led to the work of Hinton, Bengio, and LeCun [22] and today’s GPU-powered *deep* neural networks. Such networks have been developed with many architectures. For this paper, we compare a set of networks with different architectures and depths, plus networks with weights trained on natural images and on generated stimuli of the elementary perceptual reasoning tasks.

3 EXPERIMENTAL SETUP

First, we describe the commonalities across all of our experiments. We measure how different convolutional neural networks perceive low-level visual encodings, such as positions, angles, curvatures, and lengths. We formulate these measurement tasks as logistic regression problems: given a stimuli image of an elementary visualization, the networks must estimate the single quantity present or the ratio between multiple quantities present.

For each experiment, we use a single factor between-subject design, with the factor being the network used. This lets us evaluate whether different network designs are competitive against existing human perception results. We train each network in a supervised fashion with a mean-squared error (MSE) loss between the ground-truth labels and the network’s estimate of the measurement from observing the generated stimuli images. Then, we test each network’s ability to generalize to new examples with separate test data, created using the same stimuli generator function but with unseen ground-truth measurements. This means that not only the stimuli are distinct for train and test sets but also the associated labels (Section 3.2).

3.1 Networks

Multilayer Perceptron. As a baseline, we use a multilayer perceptron (MLP), but without prior convolutional layers as is typical in network designs for solving visual tasks (Fig. 2). Our MLP contains a layer of 256 perceptrons, which are activated as rectified linear units (ReLU). We train this layer with dropout (probability = 0.5) to prevent overfitting, and then combine these ReLU units to regress our output measurement.

Convolutional Neural Networks. We train different convolutional neural networks (CNNs) such as the traditional LeNet-5 with 2 layers which was designed to recognize hand-written digits [23]. For two other networks, we use weights trained from scratch and weights that have been pre-trained on a database of natural images (1000-class ImageNet [20]): the VGG19 network with 19 layers, which won the ImageNet object recognition challenge resulting in [31]; and the Xception network with 36 layers [6], which was also designed to solve the ImageNet object recognition challenge plus the 15,000-class JFT object recognition challenge [15]. All three networks have as its last layers an MLP architecture equivalent to our baseline, and so they act as earlier image and feature processors for this final regressor. Since the networks are of different architectures, the number of trainable parameters changes, with some networks having more capacity than others (Table 1).

For VGG19 and Xception, we investigate two variants: the network trained from scratch on elementary perceptual tasks, plus the network using weights that were previously trained on the ImageNet object recognition challenge *except* for the MLP layer. This is intended to produce early-layer features that mimic human vision. We did this to compare pre-trained networks with networks trained from scratch.

Optimization. All hyperparameters, optimization methods, and stopping conditions are fixed across networks (Table 1). We train for 1000 epochs using stochastic gradient descent with Nesterov momentum, but stop early if the loss does not decrease for ten epochs.

Environment. We run all experiments on Tesla X and Tesla V100 GPUs. We use the KERAS framework with a TensorFlow backend to train the networks, and use the Python scikit-image library to generate the stimuli.

3.2 Data

Image Stimuli and Labels. We create our stimuli visualizations as 100×100 binary images, rasterized without interpolation. We develop a parameterized stimuli generator for each elementary task. The number of possible parameter values differs per experiment, and we summarize these in Table 2. Before use, we scale the generated images to the range of -0.5 to 0.5 for value balance. Then, we add random noise (uniformly

Table 1: **Network Training.** We use different feature generators as input to a multilayer perceptron, which results in different sets of trainable parameters. As a baseline, we also train the MLP directly. Optimization conditions are fixed across networks and experiments.

Network	Trainable Parameters	Optimization
MLP	2,560,513	SGD (Nesterov momentum)
<i>LeNet</i> + MLP	8,026,083	Learning rate: 0.0001
<i>VGG19</i> + MLP	21,204,545	Momentum: 0.9
<i>Xception</i> + MLP	25,580,585	Batchsize: 32
		Epochs: 1000 (Early Stopping)

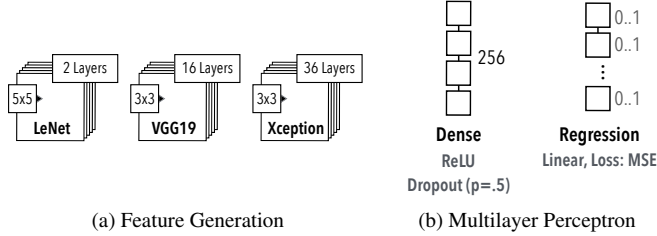


Fig. 2: **Network Architecture.** The multilayer perceptron (MLP) in our experiments has 256 neurons, activated as rectified linear units (ReLU) with dropout regularization to prevent overfitting. We perform linear regression for continuous variable output. We also learn convolutional features through LeNet (2 layers, filter size 5×5), VGG19 (16 layers, filter size 3×3), or Xception (36 layers, filter size 3×3) to test different model complexities.

distributed between -0.025 – 0.025) to each pixel to introduce variation, which prevents the networks from simply ‘remembering’ each image.

Each stimuli image also has an associated ground truth label representing the parameter set that generated the image. We scale these labels to the range of 0.0 to 1.0 and normalize to the maximum and minimum value range for each parameter.

Training/Validation/Test Splits. For each task, we use 60,000 training images, 20,000 validation images, and 20,000 test images. To create these datasets, we generate stimuli from random parameters and add them to the sets until the target number is reached, while maintaining distinct (random) parameter spaces for each set to ensure that there is no leakage between training and validation/testing.

3.3 Measures and Analysis

Cross Validation. For reproducibility, we perform repeated random sub-sampling validation, also known as Monte Carlo cross-validation, during our experiments [36]. We run every experiment separately twelve times, and randomly select (without replacement) the 60% of our data as training data, 20% as validation, and 20% as test.

Task Accuracy. In their 1984 paper, Cleveland and McGill use the midmean logistic absolute error metric (MLAE) to measure perception accuracy. To allow comparison between their human results and our machine results, we also use MLAE for presentation:

$$\text{MLAE} = \log_2(|\text{predicted percent} - \text{true percent}| + .125) \quad (1)$$

In addition to this metric, we also calculate standard error metrics such as the mean squared error (MSE) and the mean absolute error (MAE). This allows a more direct comparison of percent errors. Please note that our networks were trained using MSE loss and not directly with MLAE.

Task Confidence Intervals. We follow Cleveland and McGill and present 95% confidence intervals. We approximate the value of the 97.5 percentile point of the normal distribution for simplicity with 1.96 as suggested by the central limit theorem [1].

Confirmatory Data Analysis. To accept or reject our hypotheses, we analyze dependent variables using analysis of variance (ANOVA) followed by parametric tests.

Training Efficiency. We use the training convergence rate as a measure of how easy or hard a particular task is for the network to solve. This is defined as the MSE loss decrease per training epoch, which is an indicator of the training efficiency of the network with respect to the visual encoding. Low MSE values are better and show that the network learned the task.

Network Generalizability. With sufficient capacity of trainable parameters, it is often said that a network can ‘memorize’ the images if the data set has a low variability. Therefore it is important to consider this variability when evaluating different networks with fixed numbers of trainable parameters (Table 1). As discussed, we add noise to each stimulus image to increase variability. We also evaluate generalizability by asking a network previously trained for one task parameterization to answer questions about the same type of task stimuli but with more variability, e.g., estimating bar length without and with changes in stroke width.

Further, some experiments compare different visual encoding types, e.g., bar plot vs. stacked bar plot. We train and evaluate individual networks for each task, plus we also train and evaluate a networks on stimuli across the different visualizations. This single decision-making better mimics the judgments that a human would be able to make.

4 EXPERIMENT 1: ELEMENTARY PERCEPTUAL TASKS

Cleveland and McGill describe a set of elementary graphical perceptual tasks across ten encodings, where each encodes a quantitative variable in a graphical element or visual mark [7, 8]. These tasks are the low-level building blocks for information visualizations (Table 2): estimating position on a common scale, position on non-aligned scales, length, direction (or slope), angle, area, volume, curvature, and shading (or ink density). We leave color saturation experiments for future work.

For these tasks, we create visualizations as 100×100 raster images, and test whether each of our networks is able to regress quantities from the images. As discussed in Section 3.3, we generate multiple versions of each elementary perceptual task. This allows us to increase task complexity. For instance, for *Position Common Scale*, first we only vary the y-position of the spot to estimate against the scale, then we include translation along the x-axis, and then we vary the spot size (Table 2). Each variation increases the number of possible images for the network to ‘learn’. Since empirical evidence suggests that CNNs are able to interpolate between different training data points, we expect the networks to perform on variations of a similar perceptual task.

4.1 Hypotheses

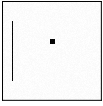
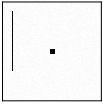
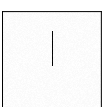
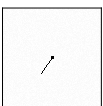
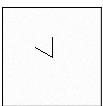
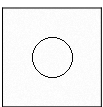
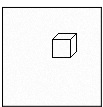
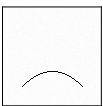
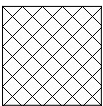
H1.1 The CNNs tested will be able to regress quantitative variables from graphical elements. We generate different visual encodings (Table 2) and test whether the CNNs can measure them.

H1.2 CNN perceptual performance will depend on network architecture. We evaluate multiple regressors with different numbers of trainable parameters. We expect a more complex network (with more trainable parameters) to perform better on elementary perceptual tasks than a network with less complexity.

H1.3 Some visual encodings will be easier to learn than others for the CNNs tested. Cleveland and McGill order the elementary perceptual tasks by accuracy. We expect this order to be relevant for computing graphical perception.

H1.4 Networks trained on perceptual tasks will generalize to more complex variations of the same task. Empirical evidence suggests that CNNs are able to generalize between different training data points. We create visual representations of the elementary perceptual tasks with different variability, and expect that networks will be able to generalize to slight task variations.

Table 2: **Elementary Perceptual Tasks.** Rasterized visualizations of the elementary perceptual tasks as defined by Cleveland and McGill [7] (color saturation excluded). We sequentially increase the number of parameters for every task (e.g., by adding translation). This introduces variability and creates increasingly more complex datasets.

Elementary Perceptual Task	Permutations
 <i>Position Common Scale</i>	
Position Y	60
+ Position X	3,600
+ Spot Size	216,000
 <i>Position Non-Aligned Scale</i>	
Position Y	600
+ Position X	36,000
+ Spot Size	216,000
 <i>Length</i>	
Length	60
+ Position Y	2,400
+ Position X	144,000
+ Width	864,000
 <i>Direction</i>	
Angle	360
+ Position Y	21,600
+ Position X	1,296,000
 <i>Angle</i>	
Angle	90
+ Position Y	5,400
+ Position X	324,000
 <i>Area</i>	
Radius	40
+ Position Y	800
+ Position X	16,000
 <i>Volume</i>	
Cube Sidelength	20
+ Position Y	400
+ Position X	8,000
 <i>Curvature</i>	
Midpoint Curvature	80
+ Position Y	1,600
+ Position X	64,000
 <i>Shading</i>	
Density	100
+ Position Y	2,000
+ Position X	40,000

4.2 Results

Overall Accuracy. The tested CNNs and MLP are able to regress the visually encoded quantities in most cases (Fig. 3), with average error across all classifiers and tasks as $MLAE=1.598$ ($SD=0.392$) and $MAE=2.903$ ($SD=0.845$). Based on these results, we **accept H1.1**.

Comparing Networks. Across network architectures and training schemes, there is considerable difference in performance. In order of decreasing error: The MLP has $MLAE=2.943$ ($SD=0.857$), for LeNet 2.125 ($SD=0.38$), Xception trained on ImageNet 1.627 ($SD=0.462$), Xception trained from scratch 1.511 ($SD=0.485$), VGG19 trained on ImageNet 0.979 ($SD=0.581$), and VGG19 trained from scratch 0.404 ($SD=0.407$). Overall, VGG19 performs best.

Across tasks, we compare the average regression performances for our networks and report the effect as statistically significant ($F_{5,48}=20.470, p<0.01$). Post hoc comparisons show that the

Table 3: **Elementary Perceptual Task Ranking.** We report midmean logistic absolute errors (MLAE) for each network averaged across multiple runs on the most complex parametrization of each task. The lower MLAE, the better (negative values are the best). For human performance, we report the ranking of Cleveland and McGill [7]. VGG19 performs best overall, while VGG19 * and Xception * networks using ImageNet yield identical rankings.

	Human (CMcG)	MLP	LeNet	VGG19 *	VGG19	Xception *	Xception
<i>Position common scale</i>							
1.		7. (3.84)	2. (1.36)	5. (1.02)	3 (-0.04)	5. (1.65)	2. (1.04)
<i>Position non-aligned scale</i>							
2.		6. (3.61)	1. (1.35)	6. (1.09)	5 (0.26)	6. (1.71)	1. (1.02)
<i>Length</i>							
3.		1. (1.99)	8. (3.19)	4. (0.87)	2 (-0.14)	4. (1.59)	3. (1.11)
<i>Direction</i>							
3.		9. (4.65)	7. (3.07)	9. (2.84)	8 (0.92)	9. (3.46)	6. (1.57)
<i>Angle</i>							
3.		8. (4.61)	9. (3.33)	8. (2.31)	9 (0.99)	8. (2.60)	7. (1.69)
<i>Area</i>							
4.		2. (2.01)	5. (2.21)	1. (0.49)	1 (-0.17)	1. (0.80)	5. (1.38)
<i>Volume</i>							
5.		4. (2.38)	4. (1.91)	7. (1.16)	7 (0.87)	7. (2.03)	9. (2.10)
<i>Curvature</i>							
5.		3. (2.34)	3. (1.81)	2. (0.71)	6 (0.28)	2. (1.17)	4. (1.13)
<i>Shading</i>							
6.		5. (3.04)	6. (2.23)	3. (0.73)	4 (0.14)	3. (1.57)	8. (1.82)

differences between LeNet and the VGG19 network, independent of the used weights, are significant ($t_{16}=4.674, p<0.01$ and $t_{16}=8.746, p<0.01$). VGG19 from scratch and Xception (both versions) perform significantly differently, with Xception from scratch ($t_{16}=4.944, p<0.01$) and Xception with ImageNet weights ($t_{16}=5.621, p<0.01$). However, differences between LeNet and both Xception networks are not significant. Taken collectively, we **partially accept H1.2**, in that higher network complexity does not automatically infer greater performance.

Ranking of Visual Encodings. Cleveland and McGill provide an ordering of elementary visual encodings based on theoretical arguments and experimental results. We compare their ranking with rankings of our networks in Table 3. Overall, there is significant variability in the rankings between architectures (Fig. 3). Area estimation is an easier task for all networks, while direction and angle estimation are more difficult. It is harder to distinguish differences between position, length, curvature, and shading tasks. Further, the volume tasks suffers high variability in performance across cross-validation splits, which suggests that the image noise affects the outcome more than for other tasks.

We note that the number of permutations across tasks does not strictly relate to network performance. While area in its most complex parameterization has 16,000 permutations, and so should be easier to learn, length has 864,000, yet VGG19 is able to achieve similar performance for both tasks. Likewise, direction has $4\times$ more permutations than angle, yet the networks achieve similar performance.

In sum, we **partially accept H1.3**. Further, the rankings between networks using ImageNet weights are identical, which suggests that the information about elementary perceptual tasks gained from natural images is similar given a sufficiently-complex network.

Cross-network Variability and Network Generalizability. We measure regression performance across networks trained with different parameterizations of the elementary perceptual tasks (Fig. 4). For our best performing network (VGG19 trained from scratch), we observe that accuracy decreases only slightly as the parameterization becomes more complex as long as training examples expressing all variability are included (diagonal entries in each matrix). However, VGG19 is unable to generalize to added translation or stroke width variations in the encodings, leading to increases in error. As such, we **reject H1.4**. These findings suggest that even slight variations in visual encodings can confuse modern CNNs by fair amounts, making it very difficult to generalize the measurement of quantities in visualizations.

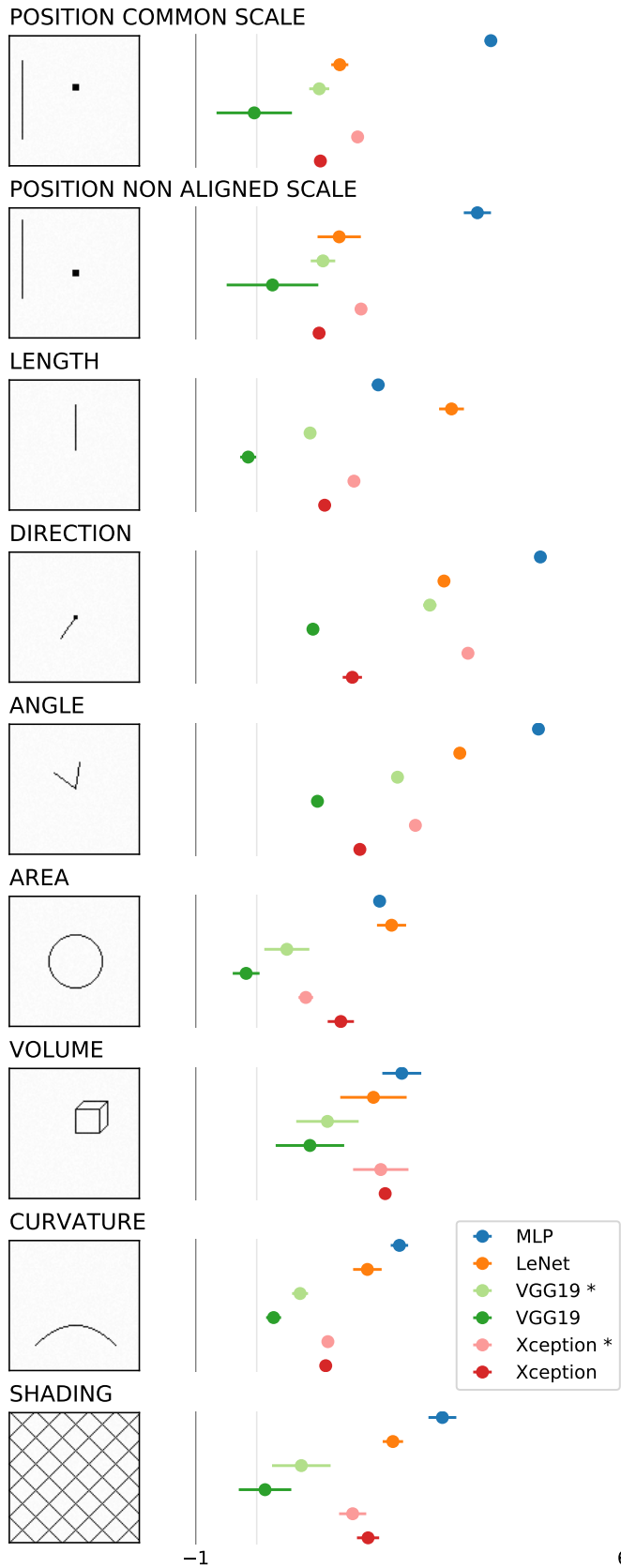


Fig. 3: **Elementary perceptual tasks results for most complex task parameterization.** *Left:* Example stimuli image. *Right:* MLAE and 95% confidence intervals for different networks. Lower MLAE scores are better. The * indicates ImageNet networks instead of being trained from scratch.

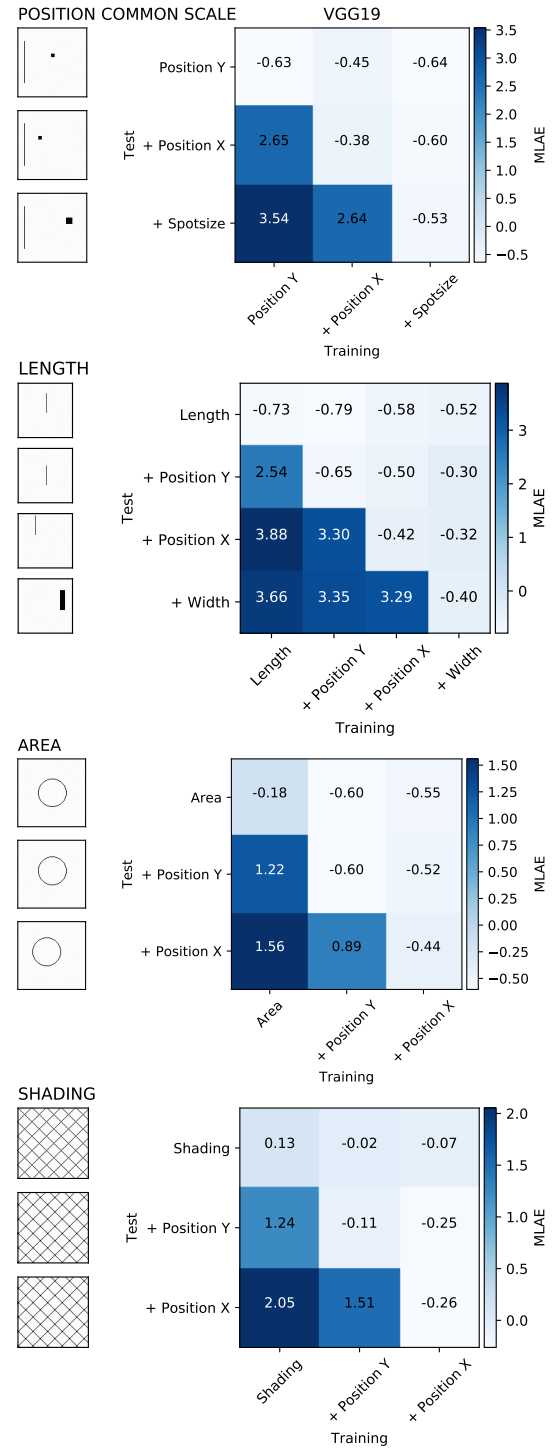


Fig. 4: **Cross-network variability for perceptual tasks.** VGG19 networks trained on one set of parametrizations (X-axis) while tested across different ones (Y-axis), for the top four performing encodings. Diagonal matrix entries represent networks trained and tested on the same parameterizations. Below diagonal entries are scenarios where the test data has more parameters than the training data; above diagonal entries have fewer. We measure the mean logistic absolute error (MLAE)—the lower the score, the better. VGG19 becomes only slightly less accurate as the parameterization becomes more complex; however, it is unable to generalize to unseen element translations as error increases rapidly. Note that all networks showed similar behavior.

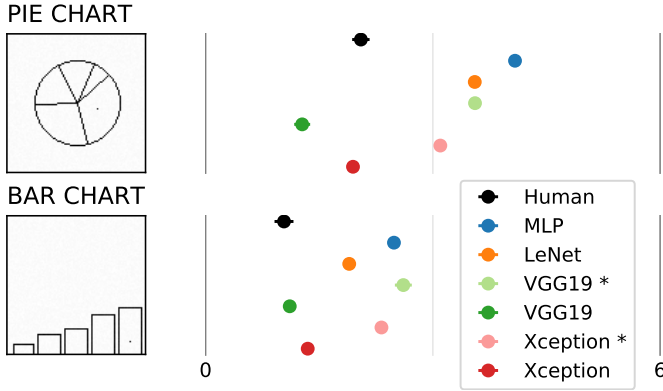


Fig. 5: **Computational results of the position-angle experiment.** *Left:* Our encodings of one data point as a pie chart and a bar chart. *Right:* MLAE and 95% confidence intervals for different networks (the lower, the better). VGG19* and Xception* are using ImageNet weights while all other networks were trained on the stimuli. We train all networks 12 times (4 times for VGG19 and Xception due to significantly longer training times). VGG19* and Xception* use ImageNet weights. Our results align with Cleveland and McGill’s human results, shown in black [7].

5 EXPERIMENT 2: POSITION-ANGLE

Cleveland and McGill measure how humans perceive the ratios of positions and angles through comparisons on bar charts and pie charts [7]. We create rasterized images following Cleveland and McGill’s proposed encoding and investigate computational perception of our networks (Fig. 1). These have five bar or pie sectors representing numbers that add to 100, where each is greater than three and smaller than 39. One required change is in the minimal differences between the values: Cleveland and McGill create stimuli where the differences between each number are greater than 0.1. However, as our networks only take 100×100 pixel images as input, we can only minimally represent a difference of 1 pixel.

Cleveland and McGill ask participants to estimate the ratio of the four smaller bars or sectors to the known and marked largest bar or sector. As such, we mark the largest quantity of the five in each visualization with a single pixel dot, then ask our networks to perform multiple regression and produce the four ratio estimates. Since the position of the largest element changes, we generate the targets such that the largest element is marked with 1 and the smaller elements follow counter-clockwise for the pie chart and to the right for the bar chart. Each of the bar and pie chart visualizations has 878,520 possible permutations.

5.1 Hypotheses

H2.1 Computed perceptual accuracy will be higher for bar charts than pie charts. Cleveland and McGill report that position judgments are almost twice as accurate (MLAE) as angle judgments in humans. Following our ranking of elementary perceptual tasks (Table 3), we see that our networks also judge position encodings more accurately than angles, and so our networks will be able to more easily judge bar charts than pie charts.

H2.2 Convolutional neural networks will learn to regress bar chart ratios faster than pie chart ratios in training. This follows directly from H2.1.

5.2 Results

Perceptual Accuracy. Our networks are able to regress the task ratios for bar charts and pie charts (Fig. 5). Cross-validation yields an average $MLAE = 2.176$ ($SD = 0.456$) for bar charts, and an average $MLAE = 3.296$ ($SD = 0.77$) for pie charts. This difference is statistically significant ($F_{1,110} = 86.061, p < 0.01$), and so we **accept H2.1**.

Post hoc comparisons show that this holds for most networks: MLP for pie charts 4.09 ($SD = 0.027$) and for bar charts 2.494 ($SD = 0.068$)

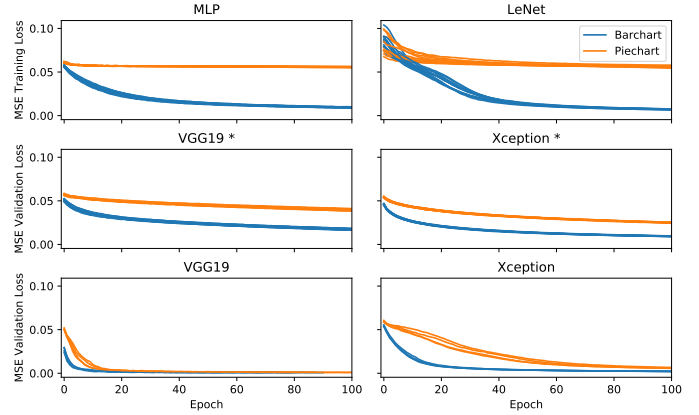


Fig. 6: **Training efficiency of the position-angle experiment.** Mean Squared Error (MSE) loss after each epoch during training, computed on previously-unseen validation data. We train all networks 12 times (4 times for VGG19 and Xception due to significantly longer training times). VGG19* and Xception* use ImageNet weights. All networks reduce MSE loss faster when learning bar charts compared to pie charts.

is significant ($t_{22} = 72.300, p < 0.01$); LeNet for pie charts 3.556 ($SD = 0.022$) and for bar charts 1.902 ($SD = 0.08$) is significant $t_{22} = 66.111, p < 0.01$; VGG19* with ImageNet weights for pie charts 3.561 ($SD = 0.047$) and for bar charts 2.601 ($SD = 0.113$) is significant $t_{22} = 25.919, p < 0.01$; Xception* with ImageNet weights for pie charts 3.094 ($SD = 0.046$) and for bar charts 2.315 ($SD = 0.032$) is significant $t_{22} = 46.329, p < 0.01$; Xception from scratch for pie charts 1.939 ($SD = 0.1$) and for bar charts 1.375 ($SD = 0.062$) is significant $t_{22} = 8.276, p < 0.01$; but the difference for VGG19 from scratch (pie charts 1.297 ($SD = 0.129$), bar charts 1.153 ($SD = 0.09$)) was not significant with $p < 0.05$. This outcome is in line with the elementary perceptual task results (Table 3), where VGG19 was the most successful network, where networks trained from scratch were more performant, and where angle was more difficult to learn than position.

Training Efficiency. We measure the MSE loss for all networks on previously-unseen validation data during training. We consider a network as converged when this validation loss does not decrease after 10 sequential epochs. Fig. 6 shows this MSE validation loss during the first twenty epochs for each condition, plotted across all cross-validation splits with overdrawn lines. The pie chart loss decreases more slowly, with the average loss over epochs being 0.052 ($SD = 0.015$) for pie charts and 0.037 ($SD = 0.018$) for bar charts. This difference is statistically significant ($F_{1,2238} = 20.656, p < 0.01$). Thus, we **accept H2.2**.

To all our networks, the bar chart is a superior visual encoding than a pie chart, in terms of accuracy and efficiency. Cleveland and McGill observe the same effect for accuracy during their human experiments.

6 EXPERIMENT 3: POSITION-LENGTH

Cleveland and McGill assess the perception of position and length across five designs of grouped and divided bar charts (Fig. 7). Both types of chart can show the same information, but the elementary perceptual task is different: a grouped bar chart always involves the judgment of positions along a common scale, while a divided bar chart requires length judgments in addition. Types 1, 2, and 3 involve the judgment of positions along a common scale while types 4 and 5 involve the measure of length. For each graph, two bars or bar segments were marked, and participants were asked to judge what percentage the smaller marked element was of the larger. From their experiment, Cleveland and McGill found type 1 to be the easiest and type 5 to be the hardest.

For data generation, we follow the same approach as in the original experiment. We generate ten value pairs using the following equation:

$$s_i = 10 \times 10^{(i-1)/12}, \quad i = 1, \dots, 10, \quad (2)$$

Then, we generate eight other random values in the range of 10 and 93. These boundaries were chosen such that the largest first-layer convolutional filter size in our networks (of 5×5 in LeNet) would see all content in our 100×100 pixel image. The paired quantities/elements are marked by a single pixel. We ask our networks to estimate the ratio of the smaller to the larger, which we model as a single value regression problem. For type 4, we follow Cleveland and McGill’s constraint that neither the top or the bottom of the marked quantities match, which forces estimations of length rather than position. This task has $9.20E + 16$ possible permutations—a challenging problem for the capacity of our networks, but one which, as Cleveland and McGill found, humans can reliably solve to within $\approx 6.5\%$ error [7].

6.1 Hypotheses

H3.1 Our networks can estimate all types equally well. A grouped bar chart involves judging a position while a divided bar chart most likely (if not type 2) requires length judgments. Our rankings of elementary perceptual tasks do not yield a strong preference for either across all networks.

H3.2 A trained multi-task network will work as well as individual trained networks. We train a multi-task network (labeled ‘multi’) from all five types. While we fix the number of trainable parameters to be the same as in the single task network, CNNs have a hierarchical structure which allows them to learn intermediate representations that are useful for multiple tasks.

6.2 Results

Perceptual Performance. We report the average MLAE for each type across our networks: for type 1 $MLAE = 3.956$ ($SD = 0.274$), for type 2 $MLAE = 3.952$ ($SD = 0.441$), for type 3 $MLAE = 4.349$ ($SD = 0.367$), for type 4 $MLAE = 3.668$ ($SD = 0.256$), and for type 5 $MLAE = 3.902$ ($SD = 0.253$). These distributions yield significance ($F_{4,25} = 2.815, p < 0.05$), but post-hoc comparisons show that only type 3 and type 4 differ ($t_{10} = 3.406, p < 0.01$). This means that the networks do not prefer a certain type on average, which leads us to **partially accept H3.1**. Further, we do not replicate the same type difficulty ordering as Cleveland and McGill in their human studies.

Overall, our networks’ performances are clearly worse than their human baselines. The problem space is much larger than in the elementary perceptual tasks, resulting in average errors of 12–20%. Further, the finding from the elementary perceptual tasks that position and length judgments had approximately equivalent rank is consistent with these findings.

Multi-task Network Performance. In Cleveland and McGill’s original position-length experiment, humans were asked to judge visualizations across types 1–5. In the last row of Fig. 7, we average the human performances of types 1–5 to create an average score across tasks. For our multi-task networks trained on all stimuli types, we record an average error across all classifiers of $MLAE = 4.358$ ($SD = 0.327$). Then, we compare against the average errors for all types, as reported above for perceptual performance. We reach significant differences ($F_{5,30} = 3.454, p < 0.05$). Post-hoc comparisons yield significant differences between the multi-task network and type 4 ($t_{10} = 3.716, p < 0.01$) and also to type 5 ($t_{10} = 2.467, p < 0.05$). Since the average MLAE is worse than all of types 1–5, and the distributions observe significant differences, we acknowledge that the multi-type task is harder for the networks than learning single types of encodings, and so we **reject H3.2**. One exception is VGG trained from scratch, which shows more promise, though performance has a wide variance across cross-validation sets.

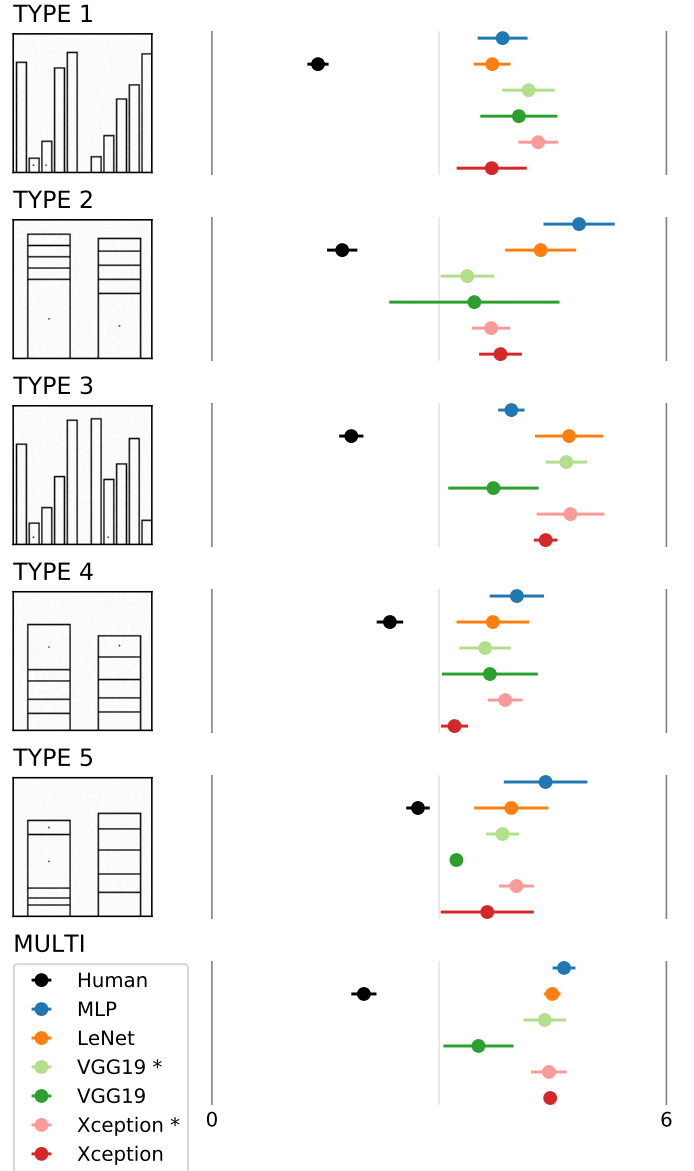


Fig. 7: **Computational results of the position-length experiment.** *Left:* Rasterized visualizations of type 1–5 for divided and grouped bar charts of Cleveland and McGill’s position-length experiment. *Right:* MLAE and 95% confidence intervals for different regressors estimating the value of marked quantities in the visualizations. The VGG19* and Xception* networks are using ImageNet weights. The top 5 rows represent networks trained on a single encoding while the last row shows ‘multi’ networks which were trained on a random stream of types 1–5. We visually compare against human performance from the original experiment.

7 EXPERIMENT 4: BARS AND FRAMED RECTANGLES AND WEBER’S LAW

Visual cues can help in converting graphical elements back to their real world variables. Cleveland and McGill introduced the bars-and-framed-rectangles experiment to compare the perceptual judgment of length and position along non-aligned scales. Fig. 8 shows both variations on the left. Without framing, it is difficult to judge which bar is larger (bottom). However, with a frame showing maximum length, this length judgment is converted into a position judgment along non-aligned scales, which simplifies the perceptual problem.

Cleveland and McGill theorize that judging the framed whitespace could be considered a length rather than a position judgment. Given this,

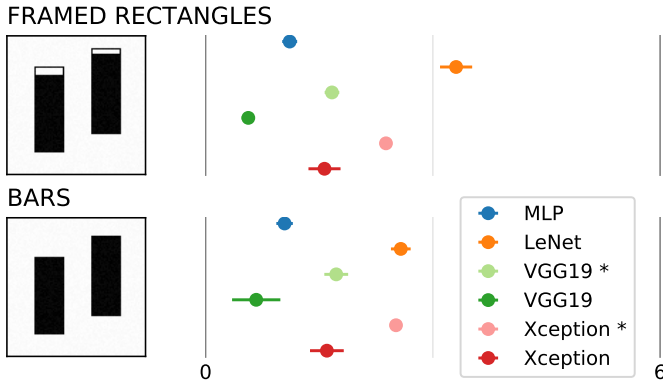


Fig. 8: **Computational results of the bars-and-framed-rectangles experiment.** *Left:* Visual encodings of two bars for length judgment (bottom) following Cleveland and McGill’s proposed experiment. Perceiving which bar is longer is much easier for humans when a frame is added (top). *Right:* For networks (trained from scratch, or * indicates ImageNet weights), there seems no significant difference between the encodings as reported by MLAE and 95% confidence intervals.

they relate the task to Weber’s Law, which states that the perceivable difference within a distribution is proportional to the initial size of the distribution [16]. For this experiment, Weber’s Law implies that humans can more easily measure the difference in the white scale since its initial size is small, whereas estimating the small change in lengths of the black bars is harder. The Just Noticeable Difference (JND) is higher when the initial stimuli is smaller in size.

We set up the bars-and-framed rectangles experiment as a two value regression task. We create rasterized visualizations of size 100×100 (Fig. 8), and ask our networks to estimate the sizes of the stimuli.

7.1 Point Cloud Experiment

We also create a random 2D point cloud version of Weber’s Law, in which the networks must estimate the number of added dots (up to 10) over an initial number of 10, 100, or 1,000 dots (Fig. 9). Each individual stimuli image has random dot placement. For 10 initial dots, a human would simply count all of the dots, but for 100 and 1,000 initial dots, this is a very difficult problem where a human is likely to simply give up. For a CNN, this problem is also very difficult: there are $\binom{100 \times 100}{10} = 2.73 \times 10^{33}$ possible locations for the 10 initial dots, which makes memorization untenable.

7.2 Hypotheses

H4.1 For bars and rectangles, the networks will improve with additional visual cues. The original bar and framed rectangle experiment shows how visual cues aid humans in mapping graphical elements to quantitative variables. This should be the same for feed-forward neural networks, as we are giving them more signal from which to learn.

H4.2 The networks will be unable to solve the point cloud experiment. This just-noticeable-difference problem has too many parameter variations to judge, though a human could solve the simplest version with 10 initial dots.

7.3 Results

Visual Cues. Our bars and framed rectangles regression task involves first identifying the smaller bar and then estimating the ratio of it to the larger. We observe varying performance for our networks. Averaged across networks: the framed rectangle encoding $MLAE = 1.982$ ($SD = 0.89$) and for the bars encoding 1.867 ($SD = 0.709$). This difference was not significant, and so we **reject H4.1**. VGG19 again is able to regress the length in both cases, for the framed rectangle encoding $MLAE = 0.595$ ($SD = 0.225$) and for the bar encoding $MLAE = 0.735$ ($SD = 0.410$), though with higher variance without

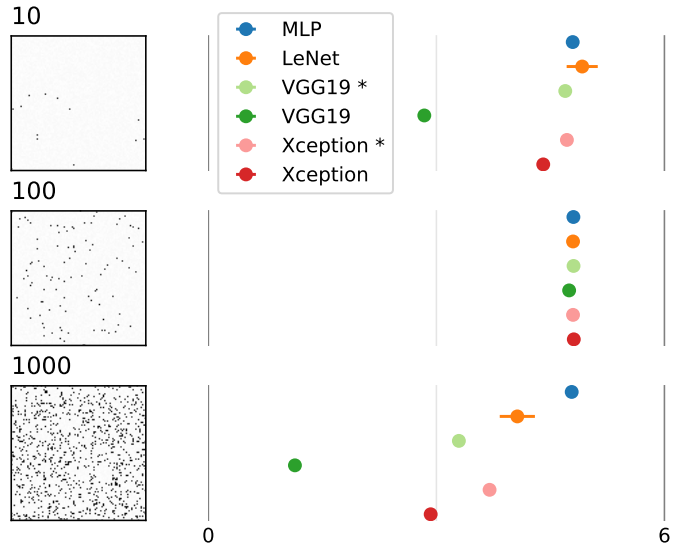


Fig. 9: **Computational results of the point cloud experiment.** *Left:* We create 2D point clouds with 10, 100, and 1000 initial dots. We then add up to 10 new dots. For humans, it is easier to count the new dots if there are initially 10 points but it is impossible to see how many dots are added when starting with 1000 dots. *Right:* We let our networks regress the number of added dots and report MLAE and 95% confidence intervals.

the added visual cues. That said, the difference in relation to the visual cue is not significant.

Weber’s Law. VGG is the only network able to solve the 10-dots version of the problem to within 10% error, which itself is surprising. Most networks achieve close to random performance, which is at $MLAE = 4.79$. Moving to 100 dots, no network succeeds and all networks achieve random performance. However, at 1,000 dots, the larger-capacity networks perform better, which again is surprising, with VGG able to solve this problem to a low error. This may be explained by the fact that, with this many dots, the problem is easier solved as a density estimation problem rather than as a counting problem. These results require further investigation, and so we only **partially reject 4.2**.

8 DISCUSSION

Across experiments, CNNs were able to regress at least some parameter variant of the graphical perception tasks to error rates of around 10%. This suggests that, for well-constrained tasks, we can use CNNs to predict measures. All our generated stimuli exhibit parameter variability on top of the added noise, which ranged from ≈ 20 permutations up to tens of orders of magnitudes of variability. We found no clear correlation between task parameterization and network capacity, and it appears that some tasks are easier for networks to learn. Area is one such task in which the multi-layer hierarchy of receptive fields can aid in task performance, whereas this architecture seems less useful for estimating direction.

Our cross-network and cross-parameterization experiments show that simple variations can significantly reduce network performance. Many of our parameter variations introduce translation to the mark in X or Y. CNNs are often said to be ‘translation invariant’, but this relies on the translation being present in the training data to begin with [18]. In effect, this asks the network to learn many versions of the elementary prediction problem across image positions. Many of our networks have the capacity to accomplish this task.

Our networks were able to solve the constrained position-angle experiment as well as a human. However, the networks were not able to solve the position-length experiment nearly as well, as the problem has many more permutations of stimuli. However, some tasks which

we did not expect to be solved, such as the 1000-point cloud JND task, were solvable by some network architectures (specifically, VGG19). This surprising result requires further investigation. More broadly, overall VGG19 appears to be a better architecture for solving graphical perception tasks, regularly outperforming Xception. We posit that this is because Xception dedicates more of its parameters to inception blocks, which attempt to provide the network with some scale invariance (a property useful for natural image object detection). However, for our 2D graphical perception tasks, this property is rarely useful.

Our networks trained on ImageNet were not better off than those trained from scratch on the perceptual tasks directly, performing worse overall. This may be unsurprising given that we ‘specialize’ these networks. This does not agree with prior comparisons between networks trained on natural images and the visual cortex, and that humans are able to solve all of these graphical perception tasks with the same visual system which views the natural world. Our experiments suggest that researchers looking to apply existing network architectures to elementary visualization problems should do so without using existing weights.

Understanding Real-world Visualizations. The application of CNNs to visualization understanding assumes an unconstrained task where the represented data values are unknown, e.g., scraping visualizations from the Web to measure attributes in bulk. In this situation, variability in chart parameterization is high—a simple *google search* for bar chart yields an incredible amount of variation. Our data suggests that networks not trained on a specific parameterization are unable to generalize, which means that the parameterization variability must be in the training data. Further, we see that task performance can drop if the number of parameters is higher than the network capacity (in a loose sense), which means that CNNs for understanding graphical data across visual designs must be large. In these respects, applying CNNs to understanding real-world visualizations remains a challenge.

New networks such as capsule networks, which aim to compartmentalize the learning of visual attributes like position, size, and orientation, hold greater promise for application to visualization [29]. Likewise, and in a different strategy, there are alternative generative approaches for learning probabilistic programs from visual stimuli, which again would explicitly represent these attributes [21]. Both of these approaches attempt to represent the fact that, at a high level, graphical perception is not a memory task, but a task requiring abstraction.

REFERENCES

- [1] I. Barany and V. H. Vu. Central limit theorems for Gaussian polytopes. *ArXiv Mathematics e-prints*, Oct. 2006.
- [2] J. Bertin and M. Barbut. *Semiologie graphique: les diagrammes, les reseaux, les cartes*. Mouton, 1967.
- [3] Z. Bylinskii, S. Alsheikh, S. Madan, A. Recasens, K. Zhong, H. Pfister, F. Durand, and A. Oliva. Understanding infographics through textual and visual tag prediction. *CoRR*, abs/1709.09215, 2017.
- [4] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand. Where should saliency models look next? In *European Conference on Computer Vision*, pp. 809–824. Springer, 2016.
- [5] M. Carpendale. Considering visual variables as a basis for information visualisation. 2003.
- [6] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pp. 1800–1807. IEEE Computer Society, 2017.
- [7] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.
- [8] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.
- [9] K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pp. 267–285. Springer, 1982.
- [10] J. Harper and M. Agrawala. Deconstructing and restyling d3 visualizations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST ’14, pp. 253–262. ACM, New York, NY, USA, 2014. doi: 10.1145/2642918.2647411
- [11] L. Harrison, D. Skau, S. Franconeri, A. Lu, and R. Chang. Influencing visual judgment through affective priming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2949–2958. ACM, 2013.
- [12] L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using weber’s law. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1943–1952, Dec 2014. doi: 10.1109/TVCG.2014.2346979
- [13] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.
- [14] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 203–212. ACM, 2010.
- [15] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.
- [16] A. S. Householder and G. Young. Weber laws, the weber law, and psychophysical analysis. *Psychometrika*, 5(3):183–193, 1940.
- [17] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890, 2016.
- [18] E. Kauderer-Abrams. Quantifying translation-invariance in convolutional neural networks. *arXiv preprint arXiv:1801.01450*, 2017.
- [19] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision*, pp. 235–251. Springer, 2016.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds., *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.
- [21] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- [22] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.
- [23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. doi: 10.1109/5.726791
- [24] D. Linsley, S. Eberhardt, T. Sharma, P. Gupta, and T. Serre. What are the visual features underlying human versus machine vision? *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 2706–2714, 2017.
- [25] J. Mackinlay. Applying a theory of graphical presentation to the graphic design of user interfaces. In *Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software*, pp. 179–189. ACM, 1988.
- [26] T. Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2015.
- [27] D. Pineo and C. Ware. Data visualization optimization via computational modeling of perception. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):309–320, Feb 2012. doi: 10.1109/TVCG.2011.52
- [28] M. Ricci, J. Kim, and T. Serre. Not-So-CLEVR: Visual Relations Strain Feedforward Neural Networks. *ArXiv e-prints*, Feb. 2018.
- [29] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. *CoRR*, abs/1710.09829, 2017.
- [30] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017.
- [31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.
- [33] A. Treisman and S. Gormican. Feature analysis in early vision: evidence from search asymmetries. *Psychological review*, 95(1):15, 1988.
- [34] D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan. Perception of elementary graphical elements in tabletop and multi-surface environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’07, pp. 473–482. ACM, New York, NY, USA, 2007. doi: 10.1145/1240624.1240701
- [35] L. Wilkinson. *The grammar of graphics*. Springer Science & Business Media, 2006.
- [36] Q.-S. Xu and Y.-Z. Liang. Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1):1–11, 2001.
- [37] D. L. Yamins and J. J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356, 2016.