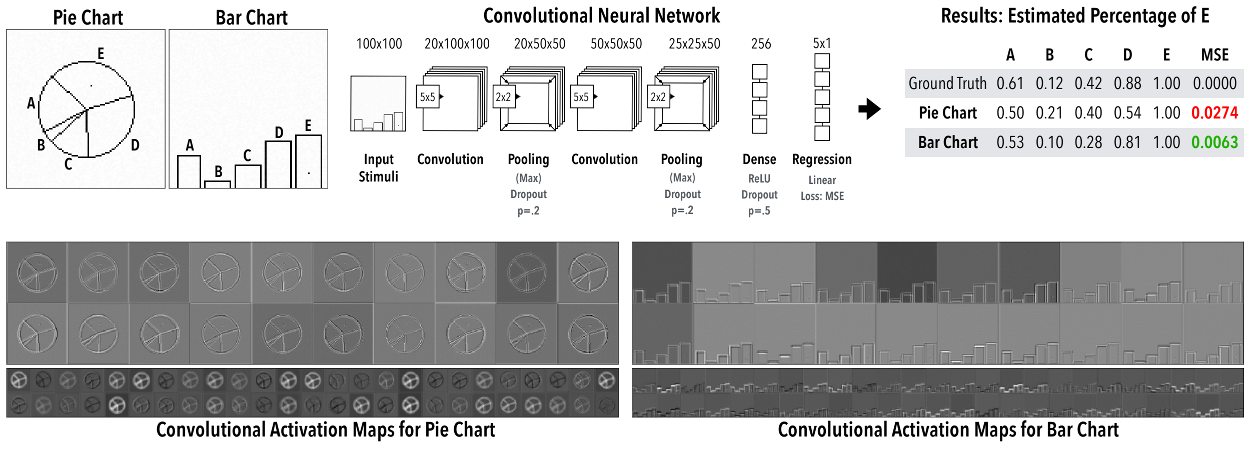
To appear in IEEE Transactions on Visualization and Computer Graphics

Evaluating 'Graphical Perception' with CNNs

Daniel Haehn, James Tompkin, and Hanspeter Pfister


Pie Chart
Bar Chart
A
B
C
D
E
Convolutional Neural Network
100x100
20x100x100
20x50x50
50x50x50
25x25x50
256
5x1
5x5
2x2
5x5
2x2
Input Stimuli
Convolution
Pooling (Max) Dropout p=.2
Convolution
Pooling (Max) Dropout p=.2
Dense ReLU Dropout p=.5
Regression Linear Loss: MSE
Results: Estimated Percentage of E
Convolutional Activation Maps for Pie Chart
Convolutional Activation Maps for Bar Chart

	A	B	C	D	E	MSE
Ground Truth	0.61	0.12	0.42	0.88	1.00	0.0000
Pie Chart	0.50	0.21	0.40	0.54	1.00	0.0274
Bar Chart	0.53	0.10	0.28	0.81	1.00	0.0063

Fig. 1: **Computing Cleveland and McGill's Position-Angle Experiment using Convolutional Neural Networks.** We replicate the original experiment by asking CNNs to assess the relationship between values encoded in pie charts and bar charts. We find that CNNs can predict 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. For visualization, how do CNNs perform when applied to graphical perception tasks? 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 network architectures on five different visualization tasks and compare to existing and new human performance baselines. While under limited circumstances CNNs are able to meet or outperform human task performance, we find that CNNs are not currently a good model for human graphical perception. We present the results of these experiments to foster the understanding of how CNNs succeed and fail when applied to data visualizations.

Index Terms—Machine Perception, Graphical Perception, Deep Learning, Convolutional Neural Networks.

1 Introduction

Convolutional neural networks (CNNs) have been successfully applied to a wide range of visual tasks, most famously to natural image object recognition [40, 41], 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 human visual cortex, even though they do not incorporate many details of biological neural networks or model higher-level abstract or symbolic reasoning [18, 31, 50]. 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 [39], with recent works targeting the systematic evaluation of the limits of feed-forward convolutional neural networks for both image recognition problems [2] and for visual relation problems [22, 36].

In visualization, there is increasing research interest in the computational analysis of graphs, charts, and visual encodings [15, 23, 34], for applications like information extraction and classification, visual question answering ("computer, which category is greater?"), or even design analysis and generation [45]. One might turn to a CNN for these tasks. However, computational analysis of visualizations is a more complex task than natural image classification [24], requiring the identification, estimation, and relation of visual marks to extract information. For instance, we take for granted the human ability to generalize an understanding of length to a previously unseen chart design, or to estimate the ratios between lengths, yet for a CNN these abilities are in question, with no clear mechanism for concept abstraction.

Our goal is to better understand the abilities of CNNs for visualization analysis, and so we investigate the performance of current off-the-shelf CNNs on visualization tasks and show what they can and cannot accomplish. As computational visualization analysis is predicated upon an understanding of elementary perceptual tasks, we consider the seminal *graphical perception* settings of Cleveland and McGill [10]. This work describes nine reasoning tasks, such as position relative to a scale, length, angle, area, and shading density, and measures human graphical perception performance on bar and pie chart quantity estimation. We reproduce Cleveland and McGill's settings with four different neural network designs of increasing sophistication (MLP, LeNet, VGG, and Xception), and compare their performance to human graphical perception. For this task, we collect new human measures for each elementary task, for the bars and frames rectangles setting, and for a Weber's law point cloud experiment. Further, as CNNs trained on natural images are said to mimic early human vision, we investigate whether using pre-trained natural image weights (via ImageNet [27]) or weights trained from scratch on elementary graphical perception tasks produces more accurate predictions.

First, we find that CNNs can more accurately predict quantities than humans for nine elementary perceptual tasks, but only if their training data includes similar stimuli. Second, that our networks can estimate bar chart lengths and pie segment angles with accuracy similar to humans, and that our networks trained more easily on bar charts. Third, that our networks were largely unable to estimate length ratios

- *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.*

across five bar chart designs, unlike humans, and that bar chart design type had largely no effect on CNN performance but a significant effect on human performance. Fourth, that framing bars make it no easier for our CNNs to estimate just noticeable differences in length, unlike for humans. Fifth, that some CNNs can solve a difficult Weber's law problem that is beyond human ability. Practically, we find that networks trained from scratch on visualizations perform better than using pre-trained natural images weights, and that current Xception networks perform worse than older VGG networks.

A second goal of our work is to help frame the visualization community's discussion of CNNs as it builds towards future computational visualization analysis applications. For this, our findings suggest that CNNs are not currently a good model for human graphical perception, and as such their application to specific visualization problems may be possible but needs care. We discuss this in more detail with respect to designing CNNs for analyzing visualizations. Further, toward this goal, we accompany this paper with open source code and data, both to enable reproduction studies and to spur new machine perception systems more adept at graphical perception: `http://vcglab.org/perception`

2 Related Work

Graphical Perception. Cleveland and McGill [10, 11] coined the phrase *graphical perception* to describe "the visual decoding of information encoded on graphs". To understand encodings in visualizations, they define *elementary perceptual tasks* as mental-visual stimuli, and rank their perceptual difficulty to humans. From these definitions, the authors perform the *position-angle* experiment to compare bar charts and pie charts, and the *position-length* experiment where users judge relations between encoded quantities in grouped and divided bar charts. The authors then use this to redesign a statistical map via *bars and framed rectangles* and Weber's law [17], 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 [19], with similar results. Harrison et al. repeated the Cleveland-McGill experiments while observing viewer emotional states, again with similar results [16]. Talbot et al. delve deeper into the impact of specific bar chart design variations on human prediction [43]. While we focus on Cleveland and McGill's work due to its repeated reproduction, many works investigate human perception to visual encoding [3, 8, 32, 33, 44, 46, 47].

Cleveland and McGill's definition of graphical perception judiciously excludes a human subject and a visual decoding method, leaving the door open for machine perception such as with CNNs. Their quality standard is only that the machine must decode *information*, which leaves narrow and well-defined applications like our experiments approachable. However, machine graphical perception must meet human breadth in capability to be generally useful in a human world.

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 into layers, which inspired computational systems based on multilayer neural networks. Fukushima and Miyake developed the early Neocognitron quantitative model [14], which ultimately led to the work of Hinton, Bengio, and LeCun [29] and today's GPU-powered deep neural networks. Such networks have been developed with many architectures attempting to model properties useful to visual reasoning, like translation invariance or part hierarchies. Across these, the error behaviors of CNNs indicate that they process images in a different way to humans even though they are loosely biologically inspired [2, 13, 42].

Computational Visualization Analysis. Pineo et al. create a computational model of early human vision based on neural networks, then optimize flow visualizations for comprehension [34]. Their simulations show that visualization perception triggers neural activity in higher-level areas of cognition, which the authors suspect is supported by low-level neurons performing elementary perceptional tasks. Other work tries to parse infographics by finding higher-level saliency models [4, 6, 7], or tries to parse text and key visual elements from visualizations using both classic and deep-learning-based computer vision [5, 15, 26, 35, 38] (to name but a few).

The field of visual question answering (VQA) has begun to combine image and text feature vectors for visualizations. Kafle et al. extend stacked attention networks with dynamic encodings to support different bar chart designs, plus present a benchmark dataset [23]. However, the ability of CNNs to solve visual relation problems like VQA has been called into question [36]. Kahou et al. attempt to answer visual questions across five chart types. However, the authors state that the task poses "a significant machine learning challenge" with no tested system able to meet human-level performance [24].

Our work takes a step back. None of these works investigate the building blocks of visualization: elementary perceptual tasks such as position, length, and angle estimation. To produce useful visualization analyzers, we must be able to computationally predict human responses in these tasks. We investigate whether this is possible with CNNs.

3 Experimental Setup

We compare CNNs to human baselines across five experiments:

- E1. We use Cleveland and McGill's elementary perceptual tasks to directly estimate quantities for visual marks (position, length, direction, angle, area, volume, curvature, and shading) (Section 4).
- E2. We reproduce Cleveland and McGill's position-angle experiment that compares pie charts to bar charts (Section 5).
- E3. We reproduce Cleveland and McGill's position-length experiment that compares grouped and divided bar charts (Section 6).
- E4. We assess the bars and framed rectangles setting from Cleveland and McGill, where visual cues aid perception (Section 7).
- E5. We conduct a Weber's law point cloud experiment (Section 8).

First, we describe the commonalities across all our experiments. In each, we measure whether different CNNs can predict values from low-level visual marks. We formulate these measurement tasks as logistic regression rather than classification problems, so that we can estimate continuous variables such as directions and angles. Given a stimuli image, 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 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 measurement labels.

3.1 Networks

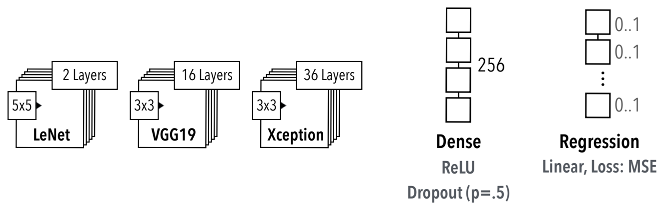
Multilayer Perceptron. As a baseline, we use a multilayer perceptron (MLP). The MLP does not have the convolutional layers which help CNNs solve visual tasks, and so we include it to tests whether a CNN is really needed to solve our simple graphical perception tasks (Fig. 2). Our MLP contains a layer of 256 perceptrons that 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 test three CNN architectures of increasing sophistication. Each has more layers than the last, which is an indicator for the network's capacity to hierarchically represent information. Each network also has more trainable parameters than the last, which is an indicator for how much information the network is able to learn overall. While larger networks need more data to train, we expect them to perform better than simpler networks.

Our smallest CNN is the traditional LeNet-5 with 2 layers, which was designed to recognize hand-written digits [30]. Next, we use the VGG19 network with 16 layers, which achieved 90% top 5 performance in the 1000-class ImageNet object recognition challenge in 2014 [40]. Finally, we use the Xception network with 36 layers [9], which achieved

Table 1: **Network Training.** We use different CNN 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 (SGD)	
		Learning rate	0.0001
MLP	2,560,513	Momentum	Nesterov
LeNet + MLP	8,026,083	Value	0.9
VGG19 + MLP	21,204,545	Batch size	32
Xception + MLP	25,580,585	Epochs	1000 (Early stop)



(a) Feature Generation (b) Multilayer Perceptron

Fig. 2: **Network Architecture.** We use a multilayer perceptron (MLP) to 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.

95% top 5 performance on ImageNet in 2017 and was also designed to solve the 17,000-class JFT object recognition challenge [20]. Xception includes state-of-the-art architecture elements: residual blocks to allow it to be deeper, and depth-wise separable convolutions (or Inception blocks) to separate spatial from cross-channel correlations for more efficient parameter use. All three networks have as their last layers an MLP architecture equivalent to our baseline, and so they act as earlier image and feature processors for this final regressor. Table 1 lists the number of trainable parameters per network.

For *VGG19* and *Xception*, we train all network parameters on elementary perceptual tasks (*from scratch*); and we use network parameters previously trained on the ImageNet object recognition challenge but retrain the parameters in the final MLP layers (*fine tuning*). We know that humans are able to perform graphical perception tasks, and so maybe these pre-trained parameters that mimic early-layer human vision features are useful for the task. However, parameters trained from scratch are unlikely to mimic human features, as the network has only seen visualization tasks and not natural images (i.e., growing up not in Flatland [1], but in Visland).

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 validation loss does not decrease for ten epochs. Each epoch trains the network upon every stimuli, with model updates after every mini-batch of 32 stimuli.

Environment. We run all experiments on NVIDIA Titan X and Tesla V100 GPUs. We use Python scikit-image to generate the stimuli and use Keras with TensorFlow to train the networks.

3.2 Data

Image Stimuli and Labels. We create our stimuli as 100×100 binary images, rasterized without interpolation. We develop a parameterized stimuli generator for each elementary task, with the number of possible parameter values differing per experiment (we summarize these in Table 1 of the supplemental material). Before use, we scale the generated images to the range of -0.5 to 0.5 for value balance. Then, we add 5% random noise (uniformly distributed between -0.025–0.025) to each pixel to introduce variation that prevents the networks from simply 'remembering' each individual image. In supplemental material Section 2, we visually compare how our stimuli vary from Cleveland and McGill's original stimuli for E1–5, and justify any differences.

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 experiment reproducibility, we perform repeated random sub-sampling validation, also known as Monte Carlo cross-validation [49]. We run every experiment separately twelve times (four times for the 'from scratch' networks due to significantly-longer training 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:

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

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.

Error Distributions/Confidence Intervals. We check for normality in our error distributions using the D'Agostino-Pearson test: 1.14% of our networks did not pass. These were typically from the smaller MLP or LeNet networks (see supplemental material for example error distributions). As such, we broadly assume normality of errors and follow Cleveland and McGill in presenting 95% confidence intervals, computed via bootstrapping (with 10,000 rather than 1,000 samples for a more accurate estimate).

Confirmatory Data Analysis. To accept or reject our hypotheses under this normality, 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. Lower MSE values are better.

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, and we also train and evaluate networks with stimuli from across the encoding types. These single decision-making networks better mimic judgments that a human would be able to make.

3.4 Human Baselines

We take human baseline measurements for the position-angle (E2) and position-length (E3) experiments from Cleveland and McGill [10], which had 51 participants. For the position-length experiment, we are also able to take human baseline measurements from Heer and Bostock's crowdsourced reproduction of Cleveland and McGill's experiments [19], which had 50 participants. Each participant in both experiments reviewed 10 stimuli for each condition.

For the three remaining experiments (E1,E4,E5), we use Amazon Mechanical Turk to crowdsource new human baselines from 25 participants. Each participant was shown 10 stimuli for each experiment condition (nine for E1, two for E4, and three for E5), with three stimuli per condition presented as practice stimuli. This totaled 182 HITs per participant, with each HIT worth \$0.06. Average HIT time was 27 seconds. 85 HITs total were rejected for out of range values. Participants were recruited from the US, with Master Worker or better qualification. As in Cleveland and McGill, participants were requested to perform "a quick visual judgment and not try to make precise measurements, either mentally or with a physical object such as a pencil or your finger."

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 [10, 11]. These tasks are the low-level building blocks for information visualizations (Figure 3): 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. We generate multiple versions of each elementary perceptual task, which 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. Each variation increases the size of the space of possible images for the network to predict (Table 1; supplemental material). Since empirical evidence suggests that CNNs can 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 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 can 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.

4.2 Results

Midmean random performance for all tasks is $MLAE = 4.8$, save for direction (4.6) where the 0–360 wrap improves random performance.

Overall Accuracy. The tested CNNs and MLP can 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 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 2. 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 task 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, suggesting 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 if 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 slight variations in visual encodings can confuse CNNs, making it difficult to generalize the measurement of quantities to unseen examples dissimilar from the training data.

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 [10]. 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 with minimum scale difference of 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

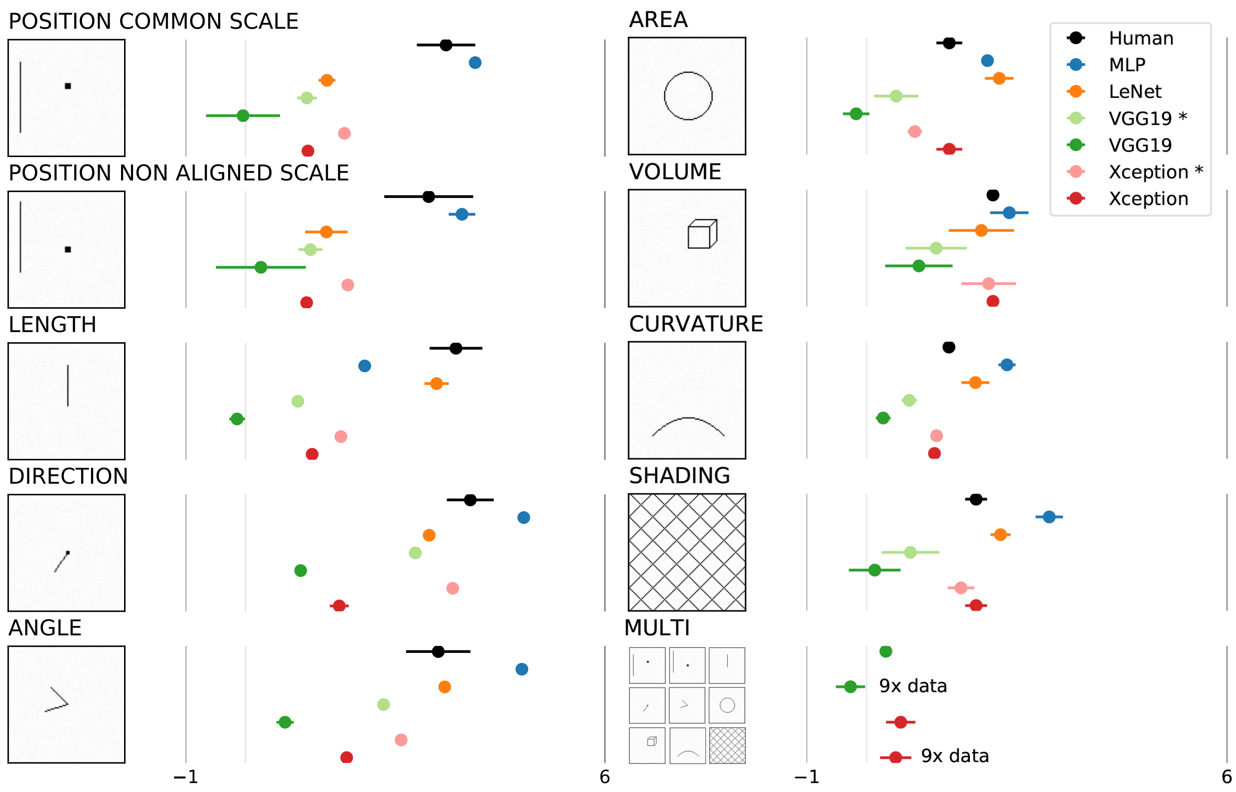


Fig. 3: **Elementary perceptual tasks results for the most complex task parameterization.** In each column: *Left:* Example stimuli image. *Right:* MLAE and bootstrapped 95% confidence intervals for different networks. Lower MLAE scores are better. The * indicates fine-tuned ImageNet weights instead of weights trained from scratch. Bottom right shows 'multi' VGG19 and Xception networks trained on all perceptual tasks, combined with optional 9× increase of training data.

Table 2: **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 [10]. VGG19 performs best overall, while VGG19 * and Xception * networks using ImageNet weights yield identical rankings.

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

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 2), 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 Performance. Midmean random performance is at $MLAE = 4.7$. 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 (0.068) 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

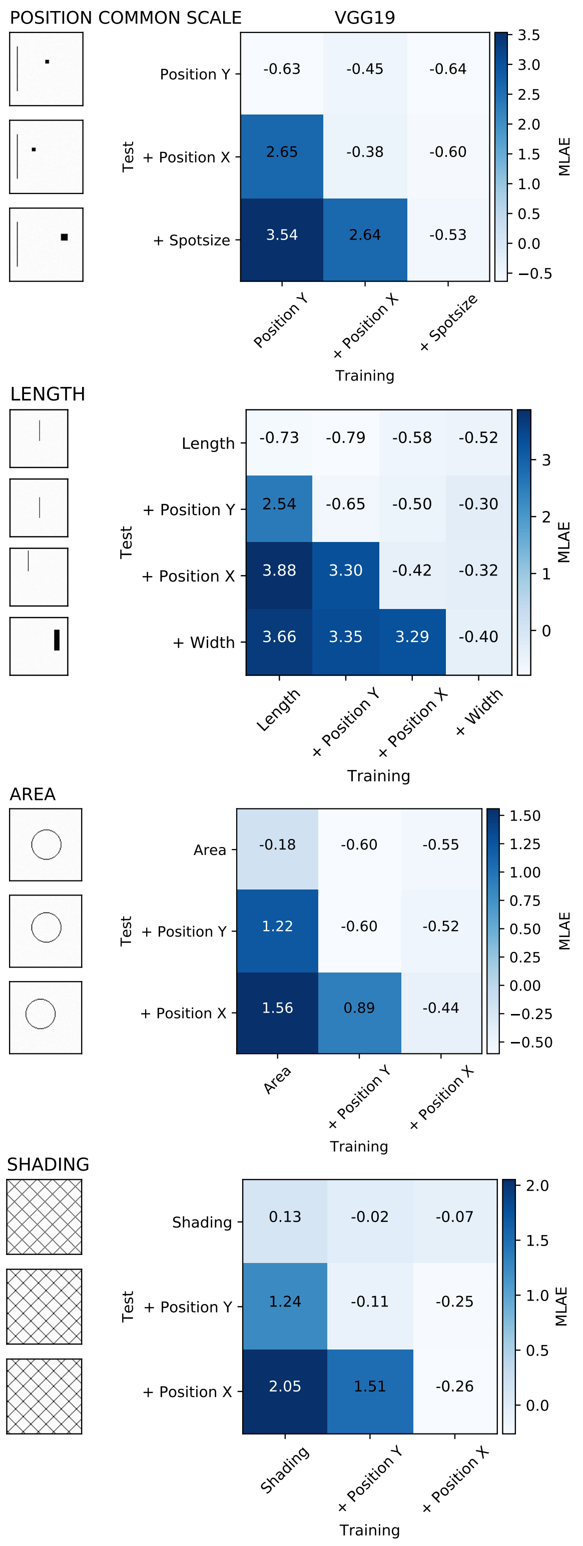


Fig. 4: **Cross-network variability for perceptual tasks.** VGG19 networks trained on one set of parametrizations (X-axis) while tested across different ones (Y-axis), for the top four performing encodings. Diagonal matrix entries represent networks trained and tested on the same parameterizations. Below 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.

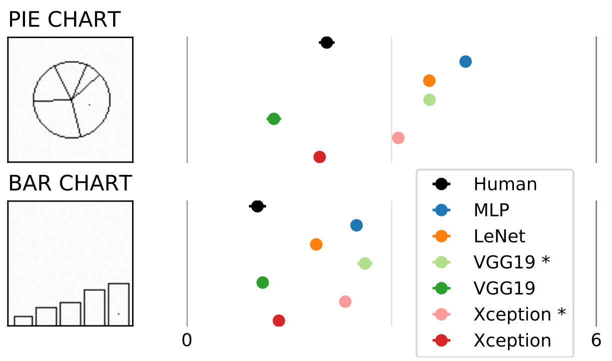


Fig. 5: **Computational results of the position-angle experiment.** *Left:* Example stimuli. *Right:* MLAE and bootstrapped 95% confidence intervals (the lower, the better). VGG19 * and Xception * fine tune ImageNet weights, with all other networks trained from scratch. We show Cleveland and McGill's human results in black [10].

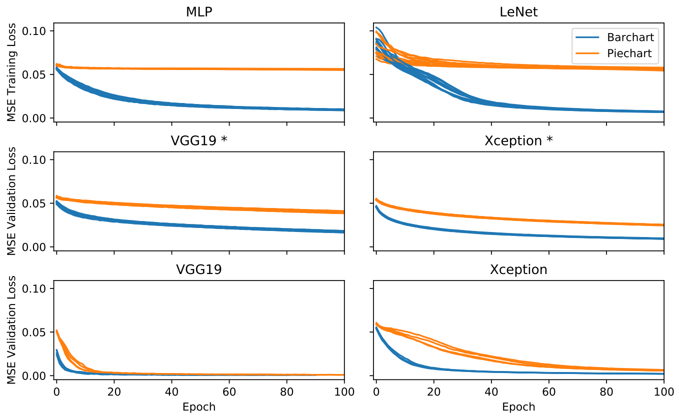


Fig. 6: **Training efficiency of the position-angle experiment.** Mean Squared Error (MSE) loss after each epoch during training, computed on previously-unseen validation data. We train all networks 12 times (4 times for VGG19 and Xception due to significantly longer training times). VGG19 * and Xception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

with the elementary perceptual task results (Table 2), where VGG19 was most successful, 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. Figure 6 shows 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 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 also requires length judgments. Types 1, 2, and 3 involve relating the judgment of positions along a common scale while types 4 and 5 involve relating 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. Cleveland and McGill ordered the types from easiest (type 1) to hardest (type 5).

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

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

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, forcing estimations of length rather than position. This task has 9.20×10^{16} possible permutations—a challenging problem for the capacity of our networks, but one that Cleveland and McGill found can be solved reliably by humans with $\approx 6.5\%$ error [10].

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. Midmean random performance is at $MLAE = 4.7$. Average MLAE across our networks is: type 1 $MLAE = 3.956$ ($SD = 0.274$), type 2 $MLAE = 3.952$ ($SD = 0.441$), type 3 $MLAE = 4.349$ ($SD = 0.367$), type 4 $MLAE = 3.668$ ($SD = 0.256$), and type 5 $MLAE = 3.902$ ($SD = 0.253$). This yields significance ($F_{4,25} = 2.815, p < 0.05$), but post-hoc comparisons show that only types 3 and 4 differ ($t_{10} = 3.406, p < 0.01$). Thus, our networks do not prefer a certain type on average, and so we **partially accept H3.1**.

Overall, our networks' performances are clearly worse than their human baselines. The problem space in these visual relation tasks 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 the 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 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. We test whether there is insufficient training data for this task by providing five times more training data; however, this decreases performance. Thus, we **reject H3.2**. One promising exception is VGG trained from scratch, though performance has a wide variance across cross-validation sets.

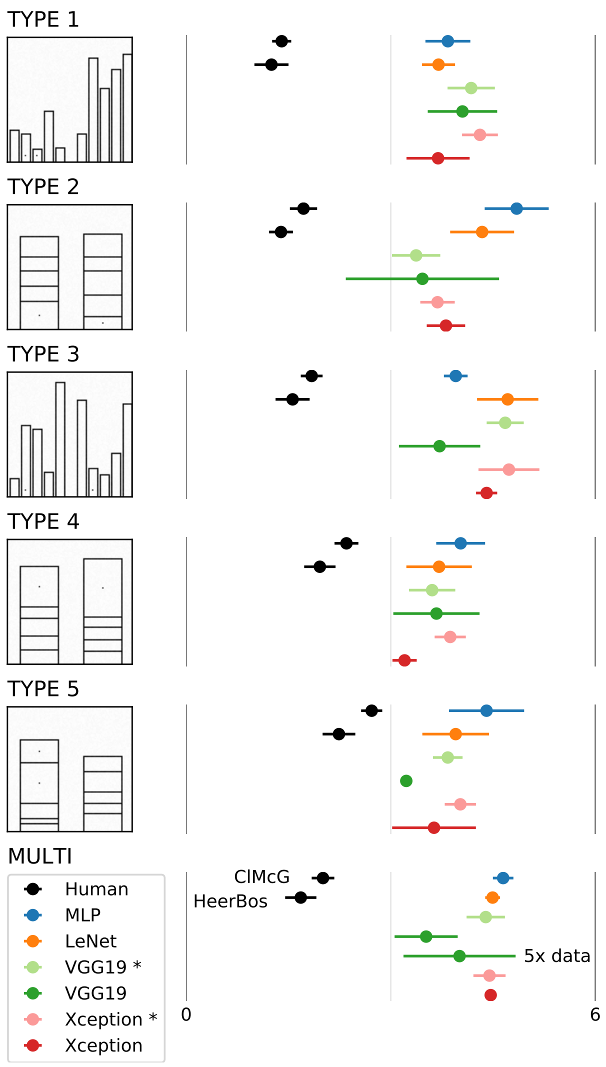


Fig. 7: **Computational results of the position-length experiment.** *Left:* Type 1–5 stimuli for divided and grouped bar charts (as per Cleveland and McGill). *Right:* MLAE and bootstrapped 95% confidence intervals of our networks. Star * denotes networks using ImageNet weights. The last row shows 'multi' networks trained on a random stream of types 1–5 (plus VGG19 with 5× training data). We include human performance (black) from the original experiment (ClMcG, top) and from Heer and Bostock's crowdsourced studies (HeerBos, bottom).

7 Experiment 4: Bars and Framed Rectangles

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 can be 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, they relate the task to Weber's Law: the perceivable difference within a distribution is proportional to the initial size of the distribution [21]. 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 stimulus is smaller in size.

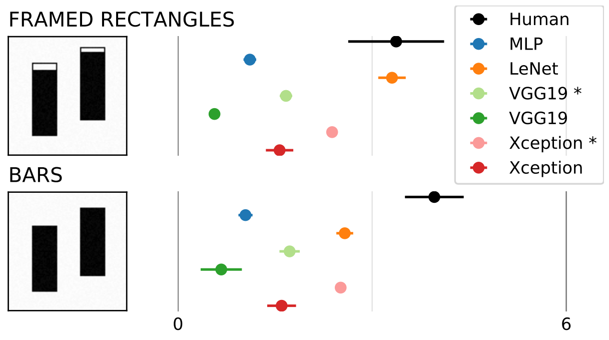


Fig. 8: **Computational results of the bars-and-framed-rectangles experiment.** *Left:* Stimuli of two bars for length judgment (bottom) following Cleveland and McGill's setting. Perceiving which bar is longer is significantly easier for humans when a frame is added (top). *Right:* For networks (trained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

We set up the experiment as a two value regression task (Fig. 8). Each bar length varies randomly by between 1 and 12 of 60 pixels, with each bar undergoing an individual vertical shift of up to 20 pixels. There are 132 different labels over 50,160 possible stimuli.

H4.1 Networks performance will improve with additional visual cues. The original bar and framed rectangle setting 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 give them more signal from which to learn.

7.1 Results

Midmean random performance in this task is equal to $MLAE = 4.8$. 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 can 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 the added visual cues. The difference in relation to the visual cue is not significant.

Further, our user study provides evidence that humans can more easily measure the frame rectangles. Participants were able to estimate the bar length with less error when frames are present $MLAE = 3.336$ ($SD = 0.828$) than without this visual aid $MLAE = 3.928$ ($SD = 0.42$). This difference is significant $F_{1,48} = 9.765, p < 0.01$.

8 Experiment 5: 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 stimulus image has random dot placement. For 10 initial dots, given a brief glance at the stimuli, a human can approximately estimate the number of the dots, but for 100 and 1,000 initial dots, this is a difficult problem where a human is likely to randomly guess. For a CNN, this problem is also difficult: there are $\binom{100\times100}{10} = 2.73 \times 10^{33}$ locations for the 10 initial dots, which makes memorization untenable.

H5.1 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.

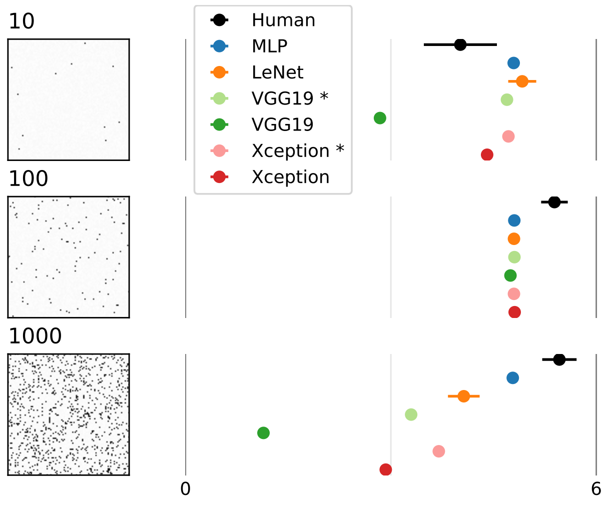


Fig. 9: **Computational results of the point cloud experiment.** *Left:* We create 2D point clouds with 10, 100, and 1000 initial dots. Then, we add up to 10 new dots. For humans, it is possible to estimate how many dots are added if there are initially 10 points, but it is impossible to see how many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

8.1 Results

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 midmean random performance, which is at $MLAE = 4.8$. Humans given a quick glance are better, solving the problem to within 17% error or $MLAE = 4.00$. Moving to 100 dots, no network succeeds, and all networks achieve random performance. For 100 and 1,000 dots, human performance is actually worse than random at 42% and 44% error respectively ($MLAE = 5.39$ and 5.46), as the participant response distributions skew towards estimating higher numbers of dots (we include response histograms in the supplemental material). At 1,000 dots, the larger-capacity networks perform better, which again is surprising, with VGG able to solve this problem to a low 3% error. This is explained by the fact that, with this many dots, the problem is easier solved as a summation problem rather than as a counting problem, which is a problem better suited to the multi-layer pooling architecture of CNNs. As such, we only **partially reject 5.1**.

9 Discussion

Performance Across Experiments. In the elementary perceptual tasks, CNNs were able to regress at least some parameter variant to error rates of around 3–10%. This suggests that, for well-constrained tasks, we can use CNNs to directly predict quantities from individual visual marks or shapes. Area is one such task in which the multi-layer hierarchy of receptive fields and pooling layers can aid in solving this task via summation, whereas these help less for prediction in our direction task. In general, our humans were less proficient at these direct estimation tasks, as they require precise geometric reasoning. CNNs solve this problem in a different way to humans, as they interpolate from similar training examples within the learned representation.

However, the CNNs approach is not able to solve the visual relation task of the position-length experiment where it must identify and compare multiple bars. The problem has many more permutations of stimuli than the capacity of any of our networks [36]. Conversely, this visual relation task is relatively simple for humans given our ability to abstract the concept of length to identify bars and compute their ratio. This suggests that new network designs are needed to solve these problems in generalizable ways and not via exhaustive training.

Finally, some tasks that we did not expect to be solved, such as the 1000-point cloud JND task, were solvable by at least one network

(VGG19). In principle, estimating the number of points added to the point cloud can be computed exactly by summing over the stimuli and subtracting the base number of points (10, 100, 1,000). This is a task that is virtually impossible at a glance for a human, but one that is made possible due to the CNNs layer aggregation methodology.

Architecture Surprises and Generalizability. We see high variance in performance across our networks. The MLP is usually least able to solve the tasks as it contains no explicit visual processing convolution layers. Next, while it has convolutional layers, the LeNet often does not have the network capacity to solve the task. Next, we find that VGG19 is the best architecture for solving graphical perception tasks, regularly outperforming the newer Xception. This is surprising because Xception has more parameters and is deeper, giving it stronger ImageNet natural image performance. Further, in our elementary perceptual task (E1) multi networks (Figure 3), even with this extra capacity, Xception was less able to exploit the 9× increase in training data than VGG.

So why is VGG19 consistently better? Recent works have discovered a similar effect in comparisons with the more complex ResNet50 and InceptionV3 networks for classification under simple geometric transformations [13]. VGG19 is better able to generalize to image translations because it better anti-aliases the input and feature map signals (in the Nyquist-Shannon sense) though the pooling and stride subsampling operations across its layers [2].

This property helps in our E1 tasks as we vary the X and Y location of the visual marks and is a useful invariance or generalizability for visualization tasks in general. Networks that do not preserve shiftability as well, like Xception, must have the capacity to learn translation invariance through the data [25]. For instance, through data augmentations which translates the stimuli artificially. This is a less efficient use of each trainable parameter, which also helps to explain the less efficient training on the 'multi' network with nine times the data.

Cross-parameterization Generalizability. That said, our cross-network and cross-parameterization experiments (Fig. 4) show that, even for VGG, this ability only goes so far to interpolate between training stimuli to new test stimuli. Without seeing any examples of X or Y translation, VGG performance deteriorates rapidly (for the other networks, too; see supplemental). This applies even to simple design variations like stroke width changes. Imagine scraping visualizations from the Web—a simple web search for 'bar chart' yields high design variation. We can hardly expect to standardize visualization designs for computational analysis, and so this means that CNN training data for general applications must include representation from across the design space [23]. Thankfully, generating visualization designs with parametric models is relatively simple, unlike for natural images.

Further, we note that task performance can drop if the number of parameters is higher than the network capacity (in a loose sense), meaning 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 [24].

From Scratch vs. Fine Tuned. Our networks fine-tuned on ImageNet were not better than those trained from scratch on the perceptual tasks, performing worse overall. This may be unsurprising given that our from-scratch networks are 'specialized'. This confounds prior comparisons made between networks trained on natural images and the visual cortex, given that humans are also able to solve graphical perception tasks with the same visual system that sees the world.

Practical Insight Summary: Across our experiments, we gained insights with practical application for other researchers or practitioners investigating machine graphical perception:

1. CNN architecture performance on natural images is not a good predictor for performance on graphical perception tasks.
2. While the general network capacity trend of 'more is better' is borne out by our experiments, other factors like generalization power through invariances are important for visualization tasks.
3. VGG19 or similar architectures are reasonable starting points.
4. Training weights from scratch is a better strategy for visualization task performance than fine tuning natural image weights.
5. Be aware when using natural image augmentations like zoom, rotation, or skew, as these have specific meanings for visualizations.
6. For training sets, we can use parametric models to explore large visualization design spaces, where data augmentation is replaced by parameter and design variations (e.g., stroke width).

10 Conclusions

We set out to investigate how current CNNs perform on graphical perception tasks, and our findings are mixed. In the constrained settings of the elementary perceptual tasks of experiment 1, CNNs perform better than humans and were able to more accurately estimate quantities directly from visual marks on images. For the CNN, these tasks require learning to predict from training stimuli with relatively minor differences (which, for it, is easy), whereas for the human these tasks require making precise geometric estimates (which, for us, is hard). In other experiments, such as the position-length experiment, CNNs cannot complete the task. These tasks require identifying the bars of interest and then measuring the ratios of their lengths. Such visual relations are much harder for CNNs to predict as the space of outcomes is much larger, requiring exhaustive training [36]. In contrast, humans can generalize their concept of length to the new task from few examples. Finally, in the point cloud task, which is largely impossible for humans for 1,000 points, we see that VGG19 can solve this task to a high accuracy through its aggregation over layers.

Overall, the variation in our findings suggest that CNNs are not currently a good model for human graphical perception, in that they do not predict human performance and in that their successes and failures are at odds with human ability. This implicitly shows that the respective mechanisms for graphical perception are not comparable. As such, researchers and practitioners should be careful when applying and drawing conclusions from these models in their work.

Future Work. In general, we are optimistic about machine graphical perception. Our findings suggest that this requires approaches that are different from the recent architectural developments like residual and inception blocks included in Xception to aid natural image classification. Networks that explicitly preserve geometric invariances, like scale, rotation, or translation, are potentially useful for graphical perception tasks [12, 48]. For instance, in the elementary angle task, rotation invariance would factor out the overall rotation and leave only the angle estimation problem; however, this must be controlled as, in the direction task, rotation invariance would remove the signal we wish to estimate. New capsule networks also hold promise as they include specific architectural mechanisms to compartmentalize the learning of visual attributes like position, size, and orientation [37]. Likewise, given that most visualization designs are easily described procedurally, there is promise in investigating generative approaches for learning probabilistic programs from visual stimuli [28]. Again, these would explicitly represent visual attributes. In general, approaches that attempt to represent higher-level abstractions are a key requirement for machine graphical perception to develop beyond a memorization and interpolation task. We release our open source code and data to help spur new machine perception systems more adept at graphical perception: `http://vcglab.org/perception`

Acknowledgments

We thank Junkyung Kim, Matthew Ricci, Daniel Pineo, Thomas Serre, John Tsotsos, Li Zhaoping, and our reviewers for helpful discussions and feedback. We thank the staff of the Harvard Human Research Protection Program for fast IRB approval and the Harvard Research Computing crew for their substantial support. We also thank the Boston Fire Department for extinguishing flames in our data center, which we promise was not caused by us training over 2,500 models for this paper. This research is supported in part by grants NSF IIS-1447344 and NIH 5U01CA198935.

References

[1] E. A. Abbott. *Flatland: Romance of Many Dimensions*. 1885.

[2] A. Azulay and Y. Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *CoRR*, abs/1805.12177, 2018.

[3] J. Bertin and M. Barbut. *Semiologie graphique: les diagrammes, les reseaux, les cartes*. Mouton, 1967.

[4] 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. *arXiv preprint arXiv:1709.09215*, 2017.

[5] 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.

[6] Z. Bylinskii, N. W. Kim, P. O'Donovan, S. Alsheikh, S. Madan, H. Pfister, F. Durand, B. Russell, and A. Hertzmann. Learning visual importance for graphic designs and data visualizations. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software & Technology*, 2017.

[7] 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.

[8] M. Carpendale. Considering visual variables as a basis for information visualisation. 2003. doi: 10.5072/PRISM/30495

[9] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pp. 1800–1807. IEEE Computer Society, 2017.

[10] 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.

[11] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.

[12] T. Cohen and M. Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 2990–2999, 2016.

[13] L. Engstrom, D. Tsipras, L. Schmidt, and A. Madry. A rotation and a translation suffice: Fooling CNNs with simple transformations. *CoRR*, abs/1712.02779, 2017.

[14] 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.

[15] 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

[16] 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.

[17] 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

[18] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

[19] 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.

[20] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[21] A. S. Householder and G. Young. Weber laws, the weber law, and psychophysical analysis. *Psychometrika*, 5(3):183–193, 1940.

[22] 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.

[23] K. Kafle, S. Cohen, B. Price, and C. Kanan. DVQA: Understanding data visualizations via question answering. In *CVPR*, 2018.

[24] S. E. Kahou, A. Atkinson, V. Michalski, Á. Kádár, A. Trischler, and Y. Bengio. FigureQA: An annotated figure dataset for visual reasoning. *CoRR*, abs/1710.07300, 2018.

[25] E. Kauderer-Abrams. Quantifying translation-invariance in convolutional neural networks. *arXiv preprint arXiv:1801.01450*, 2017.

[26] 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.

[27] 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.

[28] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[29] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[30] 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

[31] 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.

[32] 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.

[33] T. Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2015.

[34] 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

[35] J. Poco and J. Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. *Computer Graphics Forum (Proc. EuroVis)*, 2017.

[36] M. Ricci, J. Kim, and T. Serre. Not-So-CLEVR: Visual Relations Strain Feedforward Neural Networks. *ArXiv e-prints*, Feb. 2018.

[37] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. *CoRR*, abs/1710.09829, 2017.

[38] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. Revision: Automated classification, analysis and redesign of chart images. In *ACM User Interface Software & Technology (UIST)*, 2011.

[39] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017.

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[42] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[43] J. Talbot, V. Setlur, and A. Anand. Four experiments on the perception of bar charts. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2152–2160, Dec 2014. doi: 10.1109/TVCG.2014.2346320

[44] A. Treisman and S. Gormican. Feature analysis in early vision: evidence from search asymmetries. *Psychological review*, 95(1):15, 1988.

[45] F. Viégas, M. Wattenberg, D. Smilkov, J. Wexler, and D. Gundrum. Generating charts from data in a data table, 09 2018. US 20180088753 A1.

[46] 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

[47] L. Wilkinson. *The grammar of graphics*. Springer Science & Business Media, 2006.

[48] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5028–5037, 2017.

[49] Q.-S. Xu and Y.-Z. Liang. Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1):1–11, 2001.

[50] D. L. Yamins and J. J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356, 2016.