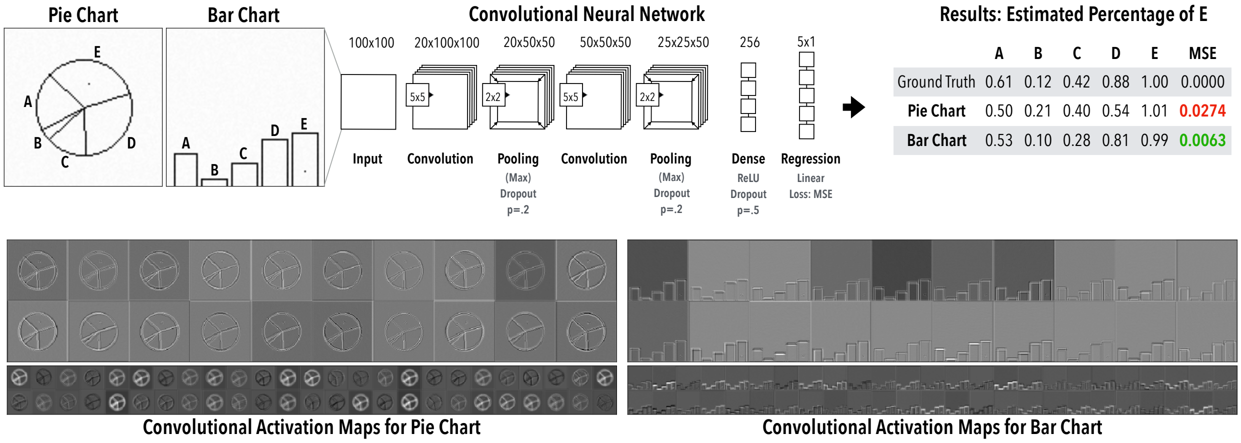
Evaluating ‘Graphical Perception’ with CNNs

Daniel Haehn, James Tompkin, and Hanspeter Pfister



	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	1.01	0.0274
Bar Chart	0.53	0.10	0.28	0.81	0.99	0.0063

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 [20, 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 [14, 24, 36]. 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 [10, 30], with recent works targeting the systematic evaluation of the visual perception limits of CNNs [18, 28].

One fundamental application of human vision is to understand data visualizations. This is a task unlike natural image processing but 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, 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.

As such, 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. Against human perception, we pit four neural networks: 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 [21]) 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.xxxx/TVCG.201x.xxxxxxx

land and McGill [7]: 1) Nine elementary perceptual tasks with increasing parameteric complexity, e.g., length estimation with fixed x, then with varying x, then with varying width, including cross-network evaluations testing the generalizability of networks to unseen parameters; 2) The position-angle experiment, which compares judgements 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, as an 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 which 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 [13] 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 [15], with similar results. Harrison *et al.* [12] 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 their 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 perceptional 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, 11, 19, 29]. 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 (though few remember their infancy). 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 exist 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 the elementary perceptual reasoning tasks themselves.

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 a separate data, created using the same stimuli generator function but with unseen ground-truth measurements (Section 3.2).

3.1 Networks

Multilayer Perceptron. As a baseline, we use a multilayer perceptron (MLP), but without the 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) (Fig. 2). 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 compare different convolutional neural networks (CNNs) with both 'trained from scratch' weights and pre-trained weights on a database of natural images (1000-class ImageNet [21]). These networks are the traditional LeNet-5 with 2 layers, which was designed to recognize hand-written digits [23]; the VGG19 network with 19 layers, which was designed to solve the ImageNet object recognition challenge [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 [16]. Each of these networks has 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 have 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 which mimic human vision, and then to see whether they are more or less useful than networks trained from scratch.

Optimization. All network 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.

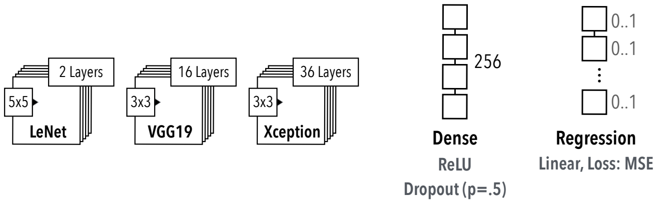
Environment. We run all experiments on Tesla X and Tesla V100 graphical processing units. We use the KERAS framework with a TensorFlow backend to train the networks, and use the scikit-image library to generate the stimuli.

3.2 Data

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

Table 1: **Network Training.** We use different feature generators as input to a multilayer perceptron which performs linear regression. This results in different sets of trainable parameters. As a baseline, we also train the MLP directly on the visualization images without any additional feature generation.

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



(a) Feature Generation

(b) Multilayer Perceptron

Fig. 2: **Network Architecture.** The multilayer perceptron (MLP) in our experiments has 256 neurons which are activated as rectified linear units (ReLU). We use Dropout regularization to prevent overfitting. As output, we perform linear regression for continuous variables. The MLP can learn to represent the visualizations directly, but we also learn features generated by LeNet (2 conv. layers, filter size 5×5), VGG19 (16 conv. layers, filter size 3×3), or Xception (36 conv. layers, filter size 3×3) to test different model complexities.

−0.025–0.025) to each pixel to introduce variation which prevents the networks from simply 'remembering' each different image.

Each stimuli image also has an associated ground truth label representing the parameter set which generated the image, e.g., the length in pixels of a bar. As before, we scale these labels to the range of 0.0 to 1.0, which represent the maximum and minimum values that this parameter can take.

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. We run every experiment seperately 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 as a presentation metric:

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

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. JT: Which 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 learn 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.

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, and so 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 prevent this. We also evaluate generalizability by asking a network previously trained upon one task parameterization to answer questions about the same type of task stimuli but with more complex parameterization, 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 parameterization, plus we also train and evaluate a networks on stimuli across the different types. This single decision-making software better mimics the judgements that a human would be able to make.

4 Experiment: 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, shading (or ink density), and color saturation. As human color perception is complex, and because Cleveland and McGill perform no experiments with it, for now we leave it for future work.

For the remaining nine tasks, we create visualizations as 100×100 raster images, and test whether each of our networks is able to regress absolute values from the images. As discussed in Section 3.3, we generate multiple parameterizations for each elementary perceptual task to allow us to increase the number of parameters that the networks must estimate, and to measure performance as we move closer towards a general representation of visual marks. 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 size of the spot size (Table 2). These parameterizations are still simple—each increase is only slightly more complex for a human to solve—but it increases the number of possible images for the network to 'learn'.

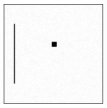
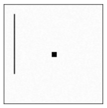
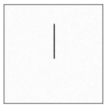
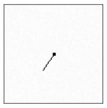
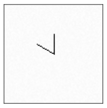
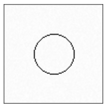
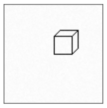
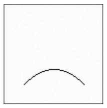
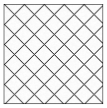
4.1 Hypotheses

H1.1 The CNNs tested will be able to regress quantitative variables from graphical elements. We parametrize different visual encodings (Table 2) and test whether the CNNs can measure them, and relate the results to accuracies obtained by humans on similar tasks.

H1.2 CNN perceptual performance will depend on network architexture. 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.

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 investigate whether this order is also relevant for computing graphical perception.

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
	*Position Common Scale*	
	Position Y	60
	+ Position X	3,600
	+ Spot Size	216,00
	*Position Non-Aligned Scale*	
	Position Y	600
	+ Position X	36,000
	+ Spot Size	216,000
	*Length*	
	Length	60
	+ Position Y	2,400
	+ Position X	144,000
	+ Width	864,000
	*Direction*	
	Angle	360
	+ Position Y	21,600
	+ Position X	1,296,000
	*Angle*	
	Angle	90
	+ Position Y	5,400
	+ Position X	324,000
	*Area*	
	Radius	40
	+ Position Y	800
	+ Position X	16,000
	*Volume*	
	Cube Sidelength	20
	+ Position Y	400
	+ Position X	8,000
	*Curvature*	
	Midpoint Curvature	80
	+ Position Y	1,600
	+ Position X	64,000
	*Shading*	
	Density	100
	+ Position Y	2,000
	+ Position X	40,000

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 by interpolating between different training data points, and so perform on variations of a similar perceptual task. We create visual representations of the elementary perceptual tasks with different variability, and expect that networks will be able to generalize when presented with slight task variations.

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 ($SD = 0.845$). From these results, we **accept H1.1**.

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. For human performance, we report the ranking of Cleveland and McGill [7]. 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)

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 ($SD = 0.407$).

Across tasks, we compare the average regression performances for our networks and report the effect of the network 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 more tunable parameters 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). However, 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. In sum, we **partially accept H1.3**. Finally, we note that the rankings between networks using ImageNet weights are identical, which suggests that the information about elementary perceptual tasks gained from those 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 so 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**.

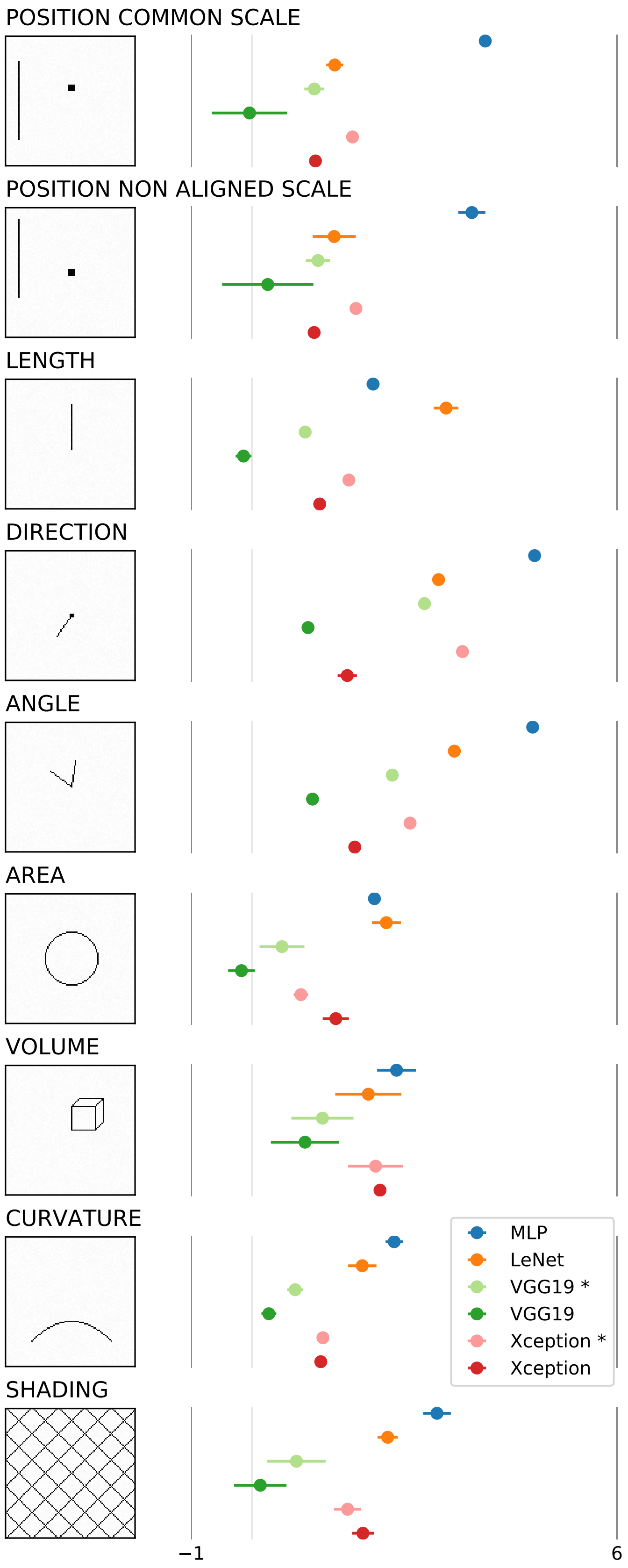


Fig. 3: **Elementary perceptual tasks results for most complex task parameterization.** *Left:* Example stimuli image. *Right:* MLAE and 95% confidence intervals for different networks. The * indicates networks which use ImageNet weights up until the MLP, rather than being trained from scratch.

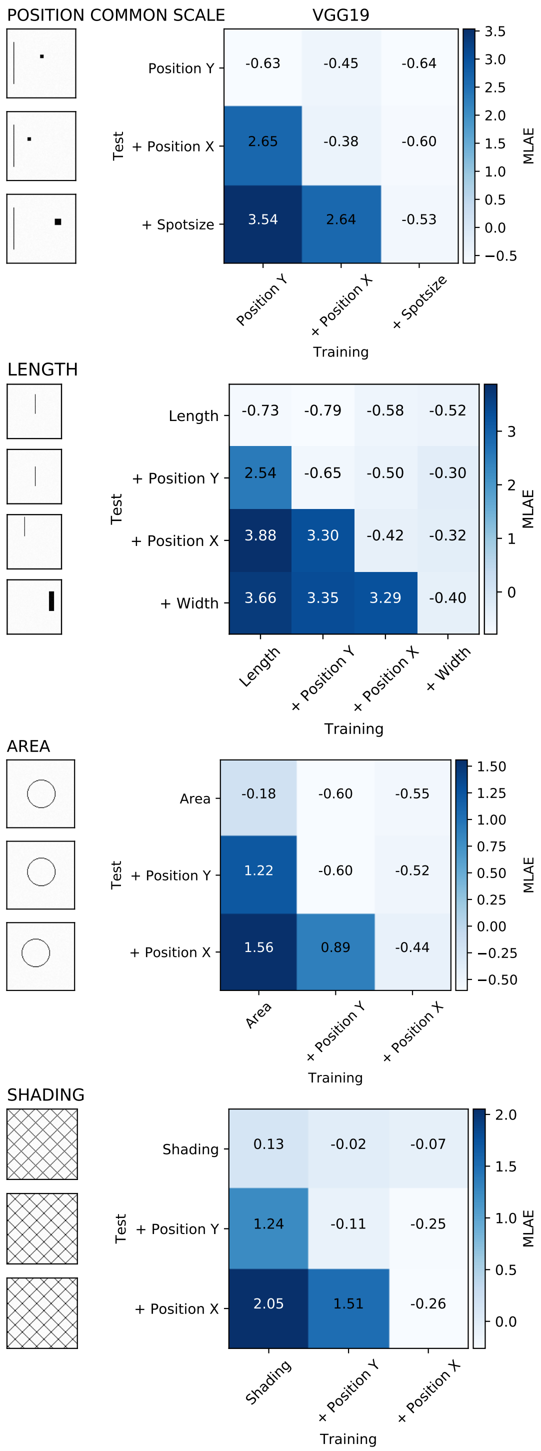


Fig. 4: **Cross-network variability for perceptual tasks.** VGG19 networks trained on different parametrizations (X-axis) tested across different parameterizations (Y-axis), for the top four performing encodings. Diagonal matrix entries represent networks trained and tested on the same parameterizations. Below diagonal entries are scenarios where the test data has more parameters than the training data; below 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.

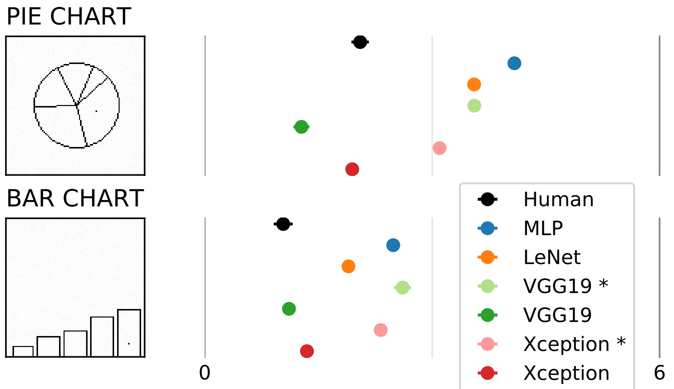


Fig. 5: **Computational results of the position-angle experiment.** *Left:* Our encodings of one data point as a pie chart and a bar chart. *Right:* MLAE and 95% confidence intervals for different networks. VGG19 * and Xception * are using ImageNet weights while all other networks were trained on the stimuli. We mimmick the original experiment of Cleveland and McGill and compare against their human results [7].

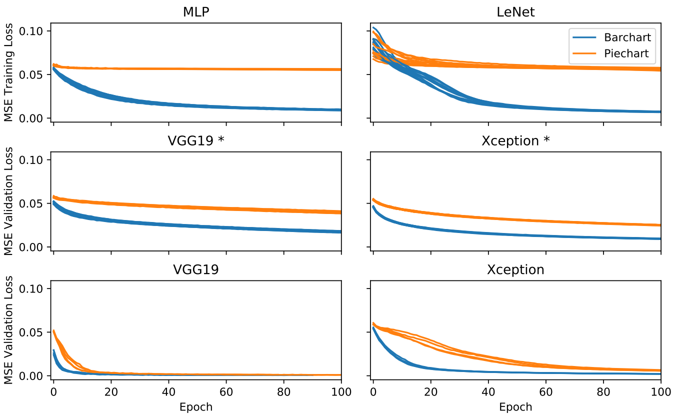


Fig. 6: **Training efficiency of the position-angle experiment.** Mean Squared Error (MSE) loss during training of our networks computed on previously unseen validation data after each epoch. The regressors estimate quantities in pie charts and bar charts. We train all networks 12 times (4 times for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

5 Experiment: Position-Angle

Cleveland and McGill measure human perception of quantities encoded as positions and as angles through their position-angle experiment [7]. The actual experiment compares pie charts versus bar charts since these map down to elementary position and angle judgement. We create rasterized images mimicking Cleveland and McGill's proposed encoding and investigate computational perception of our four networks.

We follow the data generation according to Cleveland and McGill and generate datasets of 5 numbers which add to 100. The numbers fulfill their proposed requirements of being greater than 3 smaller than 39, with differences between values being greater than 0.1. Similar to Cleveland and McGill, we create pie chart and bar chart representations (Fig. 5, left). We create these visualizations as 100×100 pixel raster images. We then mark the largest quantitiy of the five in each visualization with a single pixel dot. The regression task, again similar to the experiment bei Cleveland and McGill, is to estimate what value each quantity is in relation to the marked largest. Since the position of the largest element changes, we generate the targets in such fashion that the largest element is marked with 1 and the other quantities follow counter-clockwise for the pie chart and to the right for the bar chart. To be successful, the networks essentially first have to find the marked quantity, have the 'rolling' encoding figured out, and then estimate the quantities properly. We generate the pie chart and bar chart visualizations with $878,520$ possible permutations each which renders this regression task as a decent problem.

5.1 Hypotheses

We proposed two hypotheses entering the elementary perceptual task experiment:

- **H2.1 Computed perceptual performance is better using bar charts than pie charts.** Cleveland and McGill report that position judgements are almost twice as accurate as angle judgements. This renders bar charts superior to pie charts and should also be the case for convolutional neural networks.

- **H2.2 Convolutional neural networks can learn position faster than angles.** We assume that regressing bar charts is easier than understanding pie charts. Following our ranking of elementary perceptual tasks (Table 3), we suspect that our networks learn encodings of positions faster than angles resulting in more efficient training and faster convergence.

5.2 Results

Perceptual Performance. Our networks are able to perform the regression task for bar charts and for pie charts (Fig. 5). We evaluate over 56 runs for each condition *visual encoding* (12 runs per network, but only 4 for VGG19 and Xception due to higher training times), which yields an average $MLAE = 2.176$ for bar chart ($SD = 0.456$), and 3.296 ($SD = 0.77$) for pie chart. This difference is statistically significant ($F_{1,110} = 86.061, p < 0.01$) and leads us to **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 * 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 * for pie charts 3.094 ($SD = 0.046$) and for bar charts 2.315 ($SD = 0.032$) is significant $t_{22} = 46.329, p < 0.01$, and Xception for pie charts 1.939 ($SD = 0.1$) and for bar charts 1.375 ($SD = 0.062$) is significant $t_{22} = 8.276, p < 0.01$. The difference for VGG19 (pie charts 1.297 ($SD = 0.129$), bar charts 1.153 ($SD = 0.09$)) was not significant with $p < 0.05$. This is not surprising since VGG19 is a very powerful network which can adapt to seemingly any visual encoding as seen in our ranking for elementary perceptual tasks (Table 3).

Training Efficiency. We measure the MSE loss for all networks on previously unseen validation data during training. We count a network as converged when this validation loss does not decrease after 10 sequential epochs meaning that each network and even each run stops after a varying number of training epochs. To measure the training efficiency, we look at the MSE validation loss during the first twenty epochs of 56 runs for each condition. Visually inspected, the pie chart loss decreases slower (Fig. 6). The average loss in this period is for pie charts 0.052 ($SD = 0.015$) and for bar charts 0.037 ($SD = 0.018$). This difference is statistically significant ($F_{1,2238} = 20.656, p < 0.01$). We therefor conclude that networks train more efficiently and faster when learning bar charts and **accept H2.2**.

It seems that the visual encoding of bar charts is superior to pie charts in terms of performance and efficiency. This is interesting since Cleveland and McGill observe the same effect during their human experiment and conclude that the perceptual task of estimating position is easier for humans than the estimation of angles. Our ranking of elementary perceptual tasks yields a low score for angles and the related encoding of directions while position ranks in the mid to top.

6 Experiment: Position-Length

The position-length experiment by Cleveland and McGill involves the perception of five types of visualizations [7]. The visualizations are either grouped bar charts or divided bar charts (Fig. 7. Both types of graphs can show the same information but the elementary perceptual task is different. According to the theory by Cleveland and McGill, a grouped bar chart always involves the judgement of positions along a common scale. On the other hand, a divided bar chart might require length judgements in addition. Figure 7 shows different charts of both types on the left. As identified by Cleveland and McGill, types 1, 2, and 3 involve the judgement of positions along a common scale while types 4 and 5 involve the measure of length. This means that this experiment does not just compare grouped versus divided bar charts but rather comparing position and length judgements.

For data generation we follow the same approach as in the original experiment. We generate pairs from ten values generated using Cleveland and McGill's equation

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

with two different values for each pair. We then generate 8 other random values in the range of 10 and 93. These boundaries were chosen because we create rasterized visualizations using 100×100 pixel and want to stay in frame for a convolutional filter size of 5×5 in LeNet. We then visually encode the ten values as type 1–5. The paired quantities get marked by a single pixel. These are our quantities of interest and the task is to estimate the ratio of the smaller to the larger. We model this task as a single value regression. For type 4, we follow Cleveland and McGill's constraint that neither the top or the bottom of the marked quantities match to force length estimations rather than position.

We model this experiment as a single value regression task and ask networks to estimate the percentage of the smaller to the larger marked quantity in each visual encoding. The network first has to find the two marked quantities, then identify the smaller one, and finally estimate the ratio in comparison to the larger quantitiy. The 8 random quantities (which should be ignored by the network) push the number of possible permutations to a massive $9.20E+16$ and creates a very challenging problem.

Finally, we also train 'multi' networks which include all five types resulting in an even bigger challenge.

6.1 Hypotheses

We proposed two hypotheses entering the elementary perceptual task experiment:

- **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 judgements. Our rankings of elementary perceptual tasks do not yield a strong preference for either across all networks.
- **H3.2 Trained 'multi' networks work as well as individual trained networks.** Convolutional neural networks have massive numbers of trainable parameters. Their complexity allows them to learn different types of visual encodings in one training session.

6.2 Results

Perceptual Performance. In the original position-length experiment, Cleveland and McGill report that types 1-5 were post-ordered by their perceptual difficulty with type 1 being the easiest and type 5 the hardest. We report the average MLAE for each type across our networks (and total 56 runs per type) as follows: 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 comparions show that really only type 3 and type 4 differ ($t_{10} = 3.406, p < 0.01$). This

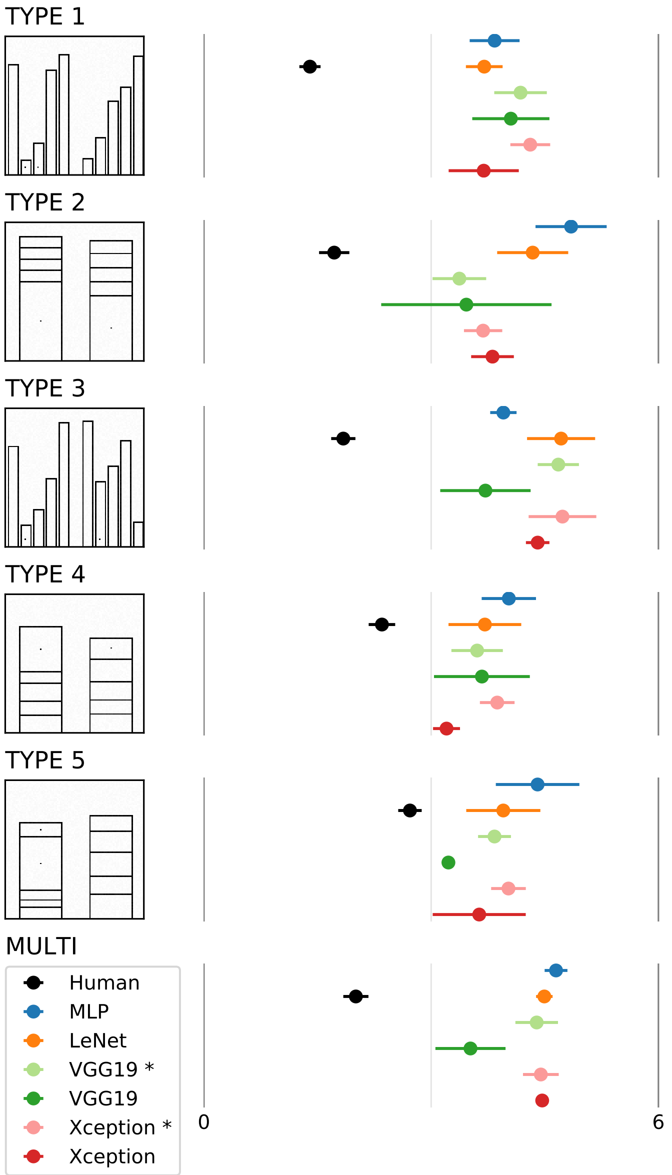


Fig. 7: **Computational results of the position-length experiment.** *Left:* Rasterized visualizations of type 1–5 for divided and grouped bar charts of Cleveland and McGill's position-length experiment. *Right:* MLAE and 95% confidence intervals for different regressors estimating the value of marked 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.

means that the networks do not prefer a certain type on average which leads us to **partially accept H3.1** and we do not replicate the same pattern as Cleveland and McGill in their human studies even though our networks' performance is clearly worse than their human baseline.

'Multi' Network Performance. In Cleveland and McGill's original position-length experiment, humans were asked to judge visualizations of types 1-5. In the bottom of Figure 7 we average the human performances of types 1-5 to create a 'multi' human. Similarly, when training our networks, we first limit each training to one stimuli. However, a 'multi' network which learns types 1-5 simultaenously is closer to the human in the original user study. For

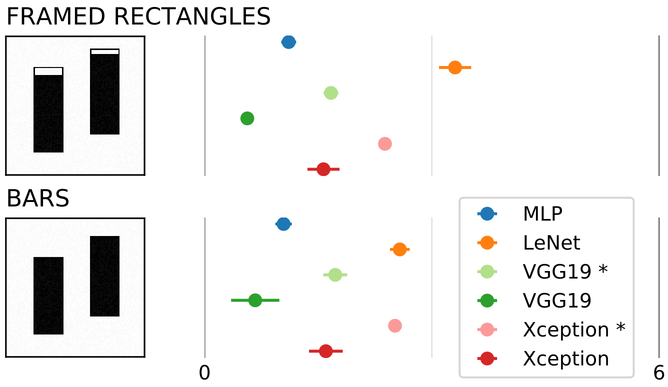


Fig. 8: **Computational results of the Bars-and-Framed-Rectangles experiment.** Log absolute error means and 95% confidence intervals for the *bars-and-framed-rectangles experiment* as described by Cleveland and McGill [7]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network, as well as feature generation using the VGG19 and Xception networks trained on ImageNet.

our 'multi' experiment, we record an average error across all classifiers of $MLAE = 4.358$ ($SD = 0.327$). We then 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' experiment and type 4 ($t_{10} = 3.716, p < 0.01$) and also to type 5 ($t_{10} = 2.467, p < 0.05$). Since our the average MLAE is worse than all of type 1–5, and the distributions observe significant differences, we acknowledge that the 'multi' task is harder for the networks than learning single types of encodings. This might relate to us fixing all other parameters of the experiment such as the number of early stopping epochs and the size of training data, however, we **reject H3.2**.

7 Experiment: Bars and Framed Rectangles

Visual cues can help converting graphical elements back to their real world variables. Cleveland and McGill introduced the bars-and-framed-rectangles experiment which compares the perceptual judgement tasks of length and position along non-aligned scales [7]. Figure 8 shows both variations on the left. It is not easy to judge which bar is larger in the bottom picture which involves a length judgement. However, when adding a little frame as a reference, this length judgement is transferred to a position judgement along non-aligned scales. Based on this little frame, it is easy to see that the right bar is slightly larger than the left since the whitespace in the top of the frame is smaller than the one on the left.

As Cleveland and McGill explain in their theories, judging the whitespace is actually also a length judgement rather than a position [7]. They now relate this task to Weber's Law which states that the perceivable difference within a distribution is proportional to the initial size of the distribution [17]. Here, it means that humans can easily measure the difference in the white scale since its initial size is small while estimating the small change in lengths of the black bars is not easily perceivable. The Just Noticable Difference (JND) is higher when the initial stimuli is smaller in size (here the white bars).

We mimmick the bars-and-framed rectangles experiment as a two value regression task. We create rasterized visualizations of size 100×100 as shown in Figure 8 and let our networks estimate the sizes of the stimuli.

7.1 Hypotheses

We proposed two hypotheses entering the elementary perceptual task experiment:

- **H4.1 Classifiers can leverage 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 since they are inspired by the visual system.

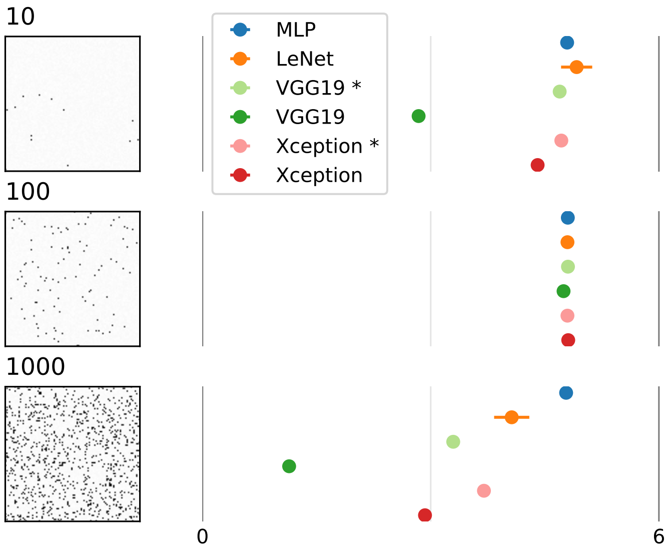


Fig. 9: **Computational results of the Weber-Fechner's Law experiment.** Log absolute error means and 95% confidence intervals for the *bars-and-framed-rectangles experiment* as described by Cleveland and McGill [7]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network, as well as feature generation using the VGG19 and Xception networks trained on ImageNet.

- **H4.2 Weber's law can be transferred to computational perception.** Cleveland and McGill confirmed Weber's law based on the bar and framed rectangle experiment. For humans, the ability to perceive change within a distribution is proportional to the size of the initial distribution.

7.2 Point Cloud Experiment

We conduct an additional experiment testing whether Weber's law applies to convolutional neural networks for graphical perception. For this, we generate three 2D point clouds as base stimuli – each is created randomly by activating 10, 100, or 1000 pixels in a 100×100 raster image. We then additionally activate from 1 to 10 random pixels within the initial distribution but by carefully only selecting inactive pixels. We show examples for this in Figure 9 (left). This means that the number of additional points is harder to identify if they are added to the 1000 pixel set while the 10 pixel set allows to easily count. We then let our networks solve a regression task to estimate the number of added points.

7.3 Results

8 Results and Discussion

Graphical Perception by CNNs. In all experiments, CNNs were able to regress visual encodings to their quantitative variables reasonably with error rates comparable to humans. This suggests that future work can enable full-blown understanding of different chart types, e.g. a classifier which can identify one type of bar charts.

Understanding Infographics by CNNs. While elementary perceptual tasks can be learned by CNNs, it seems to be a very challenging task to have CNNs 'understand' information visualizations which come in all variations. A simple *google search* for barchart yields an incredible amount of variations.

Stimuli Variability. All our generated stimuli exhibit a certain variability which ranges from a very low number of permutations of 20, for the most simple volume elementary perceptual task, to millions, for the position-length experiment. We additionally add random noise to each stimuli to ensure that the CNNs do not just memorize images. We do not observe a direct correlation between variability and 'perceptability' by our networks which suggests that the networks do not just memorize images. We also perform a direct noise and no-noise comparison (see supplemental) without any significant effect.

Concept Learning. Our cross-network experiments show that simple variations throw off the networks and result in high error rates. This suggests that the networks are not actually learning concepts but rather learn slight variations of pixel values. While we try to counteract this with our variability settings, it seems that this is the way the networks work.

Transfer Learning using ImageNet. Classifiers trained on imagenet are tuned towards natural images. While VGG19 and Xception perform better than the shallower LeNet, their full performance only develops when training from scratch. This shows how natural images are truely different than infographics.

Anti-aliasing. To keep things simple, we choose to create rasterized visualizations without interpolation. However, there might be a bias from networks trained on natural images (such as VGG19 * and Xception *, with ImageNet weights) which prefer smoother and not so prominent edges. We compare anti-aliased stimuli against the original ones without any significant effect (see supplemental).

9 Conclusions

Future work: allow insights for infovis for machines

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] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[11] 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

[12] 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.

[13] 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

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

[15] 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.

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

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

[18] 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.

[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 *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[21] 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.

[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] M. J. Seo, H. Hajishirzi, A. Farhadi, and O. Etzioni. Diagram understanding in geometry questions. In *AAAI*, pp. 2831–2838. AAAI Press, 2014.

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