How Can Deep Neural Networks Aid Visualization Perception Research? Three Studies on Correlation Judgments in Scatterplots

Fumeng Yang
fy@northwestern.edu
Northwestern University
Evanston, IL, USA

Yuxin Ma
mayx@sustech.edu.cn
Southern University of Science and Technology
Shenzhen, China

Lane Harrison
lharrison@wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

James Tompkin
james_tompkin@brown.edu
Brown University
Providence, RI, USA

David H. Laidlaw
david_laidlaw@brown.edu
Brown University
Providence, RI, USA

ABSTRACT

How deep neural networks can aid visualization perception research is a wide-open question. This paper provides insights from three perspectives—prediction, generalization, and interpretation—via training and analyzing deep convolutional neural networks on human correlation judgments in scatterplots across three studies. The first study assesses the accuracy of twenty-nine neural network architectures in predicting human judgments, finding that a subset of the architectures (e.g., VGG-19) has comparable accuracy to the best-performing regression analyses in prior research. The second study shows that the resulting models from the first study display better generalizability than prior models on two other judgment datasets for different scatterplot designs. The third study interprets visual features learned by a convolutional neural network model, providing insights about how the model makes predictions, and identifies potential features that could be investigated in human correlation perception studies. Together, this paper suggests that deep neural networks can serve as a tool for visualization perception researchers in devising potential empirical study designs and hypothesizing about perpetual judgments. The preprint, data, code, and training logs are available at `https://doi.org/10.17605/osf.io/exa8m`.

CCS CONCEPTS

• **Human-centered computing** → *Visualization design and evaluation methods*; *Information visualization*; *Empirical studies in visualization*; **User models**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

deep neural networks, visualization features, predictive modeling, perception, scatterplots

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3581111

ACM Reference Format:
Fumeng Yang, Yuxin Ma, Lane Harrison, James Tompkin, and David H. Laidlaw. 2023. How Can Deep Neural Networks Aid Visualization Perception Research? Three Studies on Correlation Judgments in Scatterplots. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3544548.3581111

1 INTRODUCTION

Understanding visualization perception is fundamental to designing visualizations and building visual analytics systems. Research in this area has been difficult because many interrelated factors can influence visualization perception. Conversely, deep neural networks appear to be promising model architectures for image data even if they do not model biological human perception. They can be optimized to recognize complicated patterns from natural images to produce classifications [27], extract infographic components [19, 59], or attempt graphical perception on elementary visual encodings [23]. These observations provoke a research question—How can deep neural networks aid visualization perception research?

To explore this intersection, we revisit one research topic in visualization perception: how people discriminate between two scatterplots for larger linear correlation, quantified by the Pearson correlation coefficient (r). Studies model perceptual precision using Just-Noticeable Difference (JND) [26, 89], but these models provide little information on where JNDs do not describe judgment behavior [26, 43]. Later studies model binary judgments [117], motivated by speculation that participants may compare correlation using visualization features as a proxy [26, 43] (e.g., a prediction ellipse [117]). These features were often defined from expert hypotheses [35, 74, 117] yet provided limited explanations as well as prediction accuracy [117].

With this exemplar topic, we conduct three studies that explore how deep convolutional neural networks might bring new inspirations to visualization perception research. The first study assesses twenty-nine deep convolutional neural network architectures in predicting participants' judgments, and finds that a subset of the architectures holds at least the same accuracy as the best-performing regression analyses in prior studies (Sec. 4). The second study shows that these models display better generalizability than prior models to two other judgment datasets (Sec. 5): one doubling data points

and the other including outliers. The third and final study extracts and categorizes features learned by a convolutional neural network model, and finds these features qualitatively reflect visualization features that may influence participants' judgments (Sec. 6). Finally, we speculate that these approaches can feasibly be applied to other visualizations and tasks to aid empirical studies and automatic design (Sec. 7): the predictive models may inspire automatic evaluation, and the features extracted by the models provide clues to theorizing visualization perception and designing new models.

The specific contributions of this research include:

- Quantitative performance measurements of twenty-nine convolutional neural network architectures for predicting participants' correlation comparison judgments in scatterplots, compared with the best-performing regression analyses using factors previously proposed in the literature;
- Quantitative evidence that some neural network models display better generalizability for correlation comparison judgments in scatterplots, compared with the regression analyses from prior studies;
- Qualitative evidence that deep convolutional neural network models trained on correlation judgment data can provide interpretable and novel visualization features, which may aid research in construing how people make perceptual judgments in scatterplots;
- Insights, limitations, and challenges of using these neural network models, including the possibility of building generalizable models for visualization perception studies;
- Three correlation comparison judgment datasets, each with 20,160 scatterplot pair images, varying in the number and distribution of points presented as well as participant performance.

2 BACKGROUND

2.1 Artificial neural networks

Inspired by biological neural networks [65, 104], artificial neural networks were designed to capture trends and patterns hidden in large data corpora. They comprise a set of inter-connected neuron layers that transform input data into different representations, and deep neural networks use multiple layers. Convolutional neural networks (CNNs) use banks of learned filters within convolutional layers that operate over windows of the previous layer. These filters extract features from input images for tasks like classification.

Existing research attempts to answer whether an artificial (convolutional) neural network can model a biological visual system [48, 50, 56, 63, 78–80, 116] (e.g., by comparing the extracted features [18, 37, 57, 94]). These works motivate this research. One difference is that we consider artificial neural networks as a means for predicting perceptual judgments, while the underlying operations may (or may not) be comparable to their biological counterparts.

2.2 Machine learning & visualization

Machine learning techniques, especially deep neural networks, were progressively applied to vision-related fields [12, 56] such as data visualization (see [108] for a survey). One compelling application is to retrieve values and labels from charts [38, 40, 66, 75, 82, 97] to improve visualization design [91] or perform Visual Question Answering (VQA) [39, 83, 84]. Deep neural networks were used to extract features from infographics [59] to automate design processes [11, 14], generate recommendations [32, 54, 61, 118, 121], and highlight visual salience [10]. Alternatively, visualizations are effective means to explain and interpret a neural network (see [30] for a survey).

Previous research explored how convolutional neural networks might apply to visualizations. Several convolutional neural networks were applied to two-value ratio judgment in elementary visual encodings, showing limited performance [23]. Similar research used convolutional neural networks to evaluate the effectiveness of graph visualizations [21]. Other research employed deep generative models to quantify and reveal various scatterplot features [36, 103]. Relevantly, neural networks are found to partially recognize Gestalt patterns [45], an important guideline for visualization perception.

On user modeling, previous studies explored sequential models to predict user click and navigation behaviors in visual search tasks [8, 76], requiring hand-engineered features. Other studies used neural networks to regress aesthetics and memorability scores of infographics [19] or learn a similarity metric for generic correlation perception and other visual quality in scatterplots [62, 92, 114]. The last category is the most related to this research, both starting with a neural network optimized from visualization images and participant judgments. However, the cited studies ended with the initial models and their predictions. This research explores far more territories: we factorize model architectures, demonstrate generalizability, and interpret the learned features.

2.3 Correlation perception & scatterplots

Early studies examine the potential factors that affect how people estimate linear correlation (r) in scatterplots [7, 16, 51, 55, 69]. Rensink and Baldridge [89] first systematically model correlation perception in scatterplots using Just-noticeable Difference (JND), a property to describe perceptual precision. A later study extended this approach to eight other visualizations and regarded JND as a metric for visualization evaluation [26]. Soon afterward, the modeling approach was advanced by introducing Bayesian data analysis [43, 74]. In addition to modeling JNDs, other research obtained weak evidence that participants used visualization features as proxies in this comparison task, and manually extracted a set of visualization features to explain participant judgments [117]. Other studies on scatterplots investigated the effects of visual marks [58, 64, 70, 106], computable features [113], mean estimation [22], indices-based correlation [95], robust regression [13, 58], and cluster perception [109].

The studies cited above found that correlation perception in visualizations may follow a systematic order that is receptive to a variety of factors like correlation coefficients [26, 89], visualization forms and features [117]. They also establish a perspective that visualization perception can be modeled. However, those JND models provide little information beyond this specific measure of perceptual precision, like where JNDs are unidentifiable (e.g., [26, 43]). Similarly, those visualization feature models have limited prediction accuracy and inadequate explanations, especially when the

initial hypotheses about the features cannot be proved. For comparison, this research also constructs four regression analyses based on the above literature. The first two regressions use correlation coefficients as predictors (cf. [26, 87, 89]), and the other two use visualization features (e.g., prediction ellipse) (cf. [69, 117]).

2.4 Two sets of open questions

The literature on machine learning and visualization is nascent and raises many questions. For example, given the numerous machine/deep learning techniques, how can we navigate which, if any, are appropriate to investigate visualization perception data? What additional values or properties can we expect from machine/deep learning techniques compared to existing modeling approaches (e.g., regression analysis)? What limitations and challenges might we encounter?

Our current research seeks to answer these questions in the context of correlation perception in scatterplots, which faces another set of open questions. First, existing models have limited accuracy in predicting people's judgments—is it possible to improve the accuracy as an improvement of model performance? Second, researchers have to collect new data and fit new models for any new hypothesis about design parameters—is it possible to generate a model of peoples' judgments that transfers across multiple visualization designs? Third, visualization features defined from expert hypotheses (e.g., [35, 117]) could be an incomplete set—is it possible to extract perceptual features from visualization images automatically? Since deep neural networks often perform well on feature extraction and prediction, we anticipate they might provide additional insight into this second set of open questions. In doing so, we also explore answers and uncover additional challenges for the first set of open questions.

3 METHODOLOGY

As our research constitutes three studies, this section outlines the methodology shared across the studies, including the human subject experiments to collect human judgments and the setups to train neural network models.

3.1 Preliminaries

This research is at the intersection of deep learning and visualization perception, so we begin with preliminaries to align readers from different backgrounds.

Following the cited studies [26, 89, 117], this research focuses on (linear) correlation comparison judgments in scatterplots. Given two side-by-side scatterplots, participants must choose which they perceive as showing more correlated data (see Fig. 1). The dichotomous responses make it possible to model the judgments as a *classification* problem. The input is raster visualization images, each combining two scatterplots. The label is participants' `Left` or `Right` judgments. The output is predictions of participant judgments.

This research collects and uses three sets of participant judgments, varying in the data properties reflected by the scatterplots (see Fig. 1). The first dataset replicates the previous studies [26, 89, 117] and constitutes scatterplots of 100 data points, denoted by Scatterplots100. The second dataset constitutes scatterplots of 200 data points, denoted by Scatterplots200. The third dataset constitutes scatterplots of 100 data points with five outliers, denoted by Scatterplots95+5. Further motivation for choosing these parameters is provided in Sec. 5.1 that covers the generalizability study.

Each dataset is randomly divided into training, validation, and test sets. The training set is used to estimate the parameters, the validation set determines when to stop training to avoid overfitting, and the test set provides an unbiased evaluation of a model.

To avoid confusion, henceforth, the term *regression (analysis)* is a method, *model* alludes to a trained (or fitted) instance with the parameters estimated, *architecture* refers to the structure, and *neural network* (or *regression*) is used when differentiating *model* and *architecture* is unnecessary.

3.2 Present research

The three studies are formalized as three research questions on correlation comparison judgments in scatterplots. They show how deep neural network models could aid a visualization perception study from three perspectives.

Study 1 asks do deep neural networks better predict participants' correlation judgments? Presented in Sec. 4, this study focuses on prediction. We assessed the prediction accuracy of a set of neural network architectures in predicting participants' judgments from the Scatterplots100 dataset. We compared them to the best-performing regression analyses based on factors proposed in the literature. A subset of neural network architectures has comparable prediction accuracy to the regression analyses.

Study 2 asks do the models better generalize predictions to other related datasets?[1] Presented in Sec. 5, this study focuses on generalization, showing the models may provide predictions for new data before human judgment is collected. We applied the top-performing models from Study 1 to predict participants' judgments in two new datasets (Scatterplots200 and Scatterplots95+5). We compared their prediction accuracies with the corresponding models trained from scratch on the new data. Two neural network architectures outperform all the others in predicting new data.

Studies 1 and 2 suggest that compared to the previous works, these neural network models may better capture the features in visualization images to predict human judgments. As such, **Study 3 asks what features a convolutional neural network model learns to predict participants' perceptual judgments?** Presented in Sec. 6, this study focuses on exploration. We first conducted an error analysis to examine the bias of a selected neural network model. We then analyzed and interpreted the visual features learned by this model. This study identifies both previously proposed and new visualization features, yielding new clues about correlation perception in scatterplots.

The generalizable predictions might aid in automatic design and evaluation (e.g., inputting them with images of new designs), and the features learned may offer clues to correlation perception and model improvement. Together, compared to years of the previous studies, the three studies show that deep neural network models

[1] Model generalizability is evaluated using the same estimated parameters. This definition is stricter than where refitting new data is allowed.

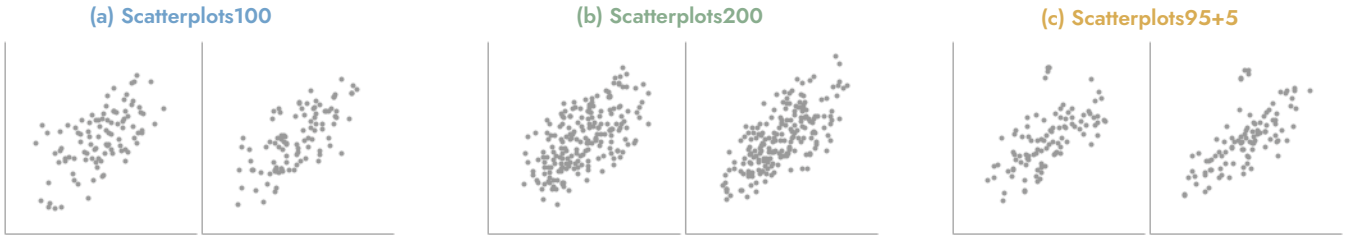

Figure 1: Example visualization images from the three datasets. We ask participants, "please choose the scatterplot that appears to show the more correlated data." In each pair, the correlation coefficients are 0.6 (left) and 0.69625 (right), respectively.

optimized from one judgment dataset potentially aid multiple difficulties in visualization perception research, including prediction, generalization, and interpretation [12].

3.3 Pilot studies

We started with pilot studies to probe experimental setups. Among a series of pilot studies, we detail one that used the 19,000 correlation comparison judgments in scatterplots collected by Yang et al. [117]. We reproduced their visualization images and trained a set of neural network models to help decide the visualization parameters of scatterplots and design our human subject experiments. Following this pilot study, in this research, each scatterplot is 150 × 150 pixels, and a scatterplot pair is 308 × 154 pixels. These allow a reasonable batch size (32) to be fit into a 16GB GPU for the largest architecture used. The visualization images are also large enough for participants to view and compare correlation in two scatterplots. Each dot in a scatterplot is 2 pixels in radius, colored in opaque gray (■#999999). As this set of 19,000 judgments (images) seemed adequate, we decided to collect roughly the same number of judgments. We also determined the data splitting strategy, learning rate, batch size, the number of epochs (for Study 1), and other hyperparameters for training the neural networks (see Sec. 3.5) based on this pilot study.

3.4 Human subject experiments

To start, we collected the three judgment datasets from three human subject experiments. These experiments shared the same design and procedure, summarized in Fig. 2 and described as follows.

Experimental design The experiments tasked participants with choosing the scatterplot that shows the higher correlation between a pair (see Fig. 1). Because the relationship between JNDs and correlation coefficients (r) was not of interest in this research, they were systematically sampled as follows. In each pair, one always had a fixed r out of six possible values (see Fig. 2 line 1), and the other approached this fixed r from either above ($r + \Delta r$) or below ($r - \Delta r$). For each approach and fixed r, we generated eight pairs and varied the perceptual distance between each pair. The perceptual distance was measured by JND of r (see Fig. 2 line 2).[2] For example, suppose JND is 0.1 when r = 0.5, we generated two scatterplot pairs for 2 JNDs: (0.5, 0.5 + 0.2) and (0.5, 0.5 − 0.2). The exception is that we only considered non-negative correlation and set any negative r to 0. Last, because these 2-JND pairs were unchallenging due to the large difference in correlation (see Appx. F), we also used them as attention checks. Each participant accomplished 6 (r levels) × 8 (JND levels) × 2 (approaches) = 96 judgments, randomized and split into two sessions of 48 judgments.

To collect Scatterplots100, we first estimated JNDs through a pilot study using the previous staircase method [26, 89, 117]. We then fine-tuned the JNDs and tested them in a sequence of pilot studies until the proportion of judgments selecting higher correlation was similar across different r levels (see Appx. B). This eliminated judgment skewness within a dataset. For Scatterplots200 and Scatterplots95+5, we assumed the same JNDs to suppress variance across the datasets, but expected participants' judgments are proportionally different (see Appx. B).

Generating stimuli We adapted the previous algorithm [26] to generate the datasets and then rendered scatterplots using `d3.js` in a browser. For a given correlation coefficient r, this algorithm first generated an initial set of random points (x_i, y_i) and then transformed the y−coordinates to meet the given r. The transformation was smaller if a point was closer to the means of x and y. We manipulated the initial set to generate Scatterplots95+5 by placing five points near the mean of x but 3.5 standard deviations away from the mean of y, resulting in five outliers (see Fig. 1c). This algorithm otherwise generated all datasets in the three experiments with minor modifications (e.g., the number of points). Each dataset was generated onsite, and thus different participants were shown different images. We recorded each dataset to later reproduce the visualization images for training deep neural network models.

Procedure After the consent and an overview, participants first viewed a grid of scatterplots at 8 different r levels as training. They then practiced 16 judgments, with feedback on if they selected the higher correlation. They then accomplished the two main sessions with an optional break in the middle, followed by a demographics questionnaire. Between any two judgments, there were 500 milliseconds of a blank white screen to eliminate visual aftereffects. A 1-minute video to illustrate the experiment is available in supplementary materials.

Participants Based on the experimental design and planned sample size, we decided to use 210 participants for one experiment.[3] All

[2]We consider only population-level, average JNDs here.

[3]The experimental design allows 96 judgments per participant, and the pilot studies show that 19,000 judgments seem sufficient. We hoped to meet 20,000 judgments and rounded the participant numbers to the nearest 10. As such, 20,000 judgments ÷ 96 judgments per participant ≈ 210 participants.

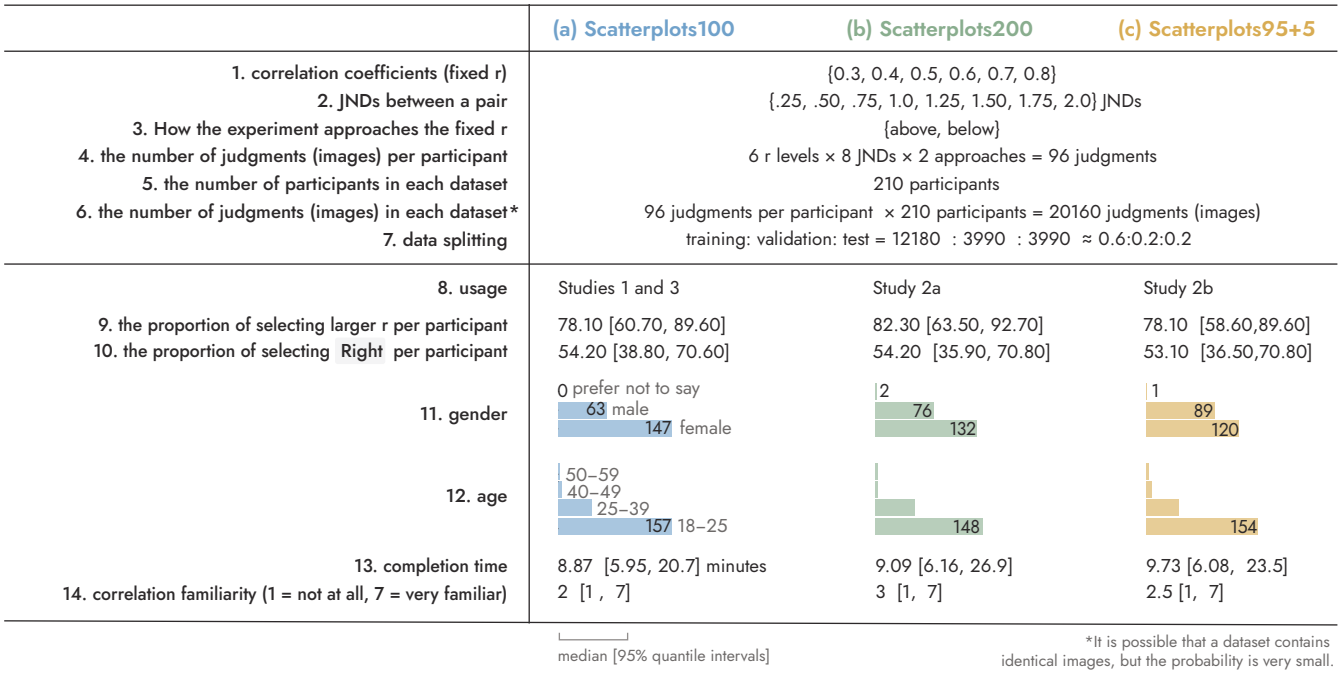

Figure 2: The summaries of the three datasets from the human subject experiments. The proportion of selecting the scatterplot on the right side (Right) is about 54.2% (cf. 50%). To account for this bias, we decided on a guessing threshold of 55%, slightly higher than the empirical observation.

participants were recruited from `Prolific.co`[4] and each was paid 1.60 USD for their time (see Fig. 2 line 13). We first recruited more than 210 participants, and then dropped participants if they failed to select higher correlation in more than 55% of judgments (the guessing threshold, see Fig. 2) or if they failed in more than half of the twelve attention checks. After this step, we had about 205 to 215 participants. We continued to drop extra participants randomly or recruit additional participants until we met 210. The pilot and earlier participants were excluded from later experiments. In total, we had 210 × 3 = 630 unique participants and collected 20,160 × 3 = 60,480 judgments (images), roughly balanced in Left and Right judgments. We used all judgments in training or testing the models.

3.5 Training neural network models

With the datasets collected, this section outlines the shared methods to create and train neural network models. The later sections report the results.

Visualization images We reproduced each scatterplot pair as a `.png` file using `Python 3.8.8`, `Matplotlib 3.4.1`, and an `SVG` backend. We manually adjusted the configuration (e.g., paddings) to match a screenshot of the same pair originally rendered by `d3.js` in the human subject experiments (see a comparison in Appx. A). Each reproduced image is 308 × 154 pixels, with a single channel.

[4]The experiments were conducted in early August 2021, when participant pool on `Prolific.co` was leaned towards young females. Our data quality is not affected by this, because the judgment distributions are similar to the previous datasets [26, 117].

Data splitting We partitioned each dataset (i.e., Scatterplots100, Scatterplots200, or Scatterplots95+5) into the training, validation, and test sets in a ratio close to 0.6:0.2:0.2. That is, we randomly and respectively assigned 58, 19, and 19 judgments from each participant to the three folds.

Implementation We used `Python 3.7.11`, `PyTorch 1.10.0`, `PyTorch lightning 1.5.10`, and `torchvision 0.11.3`, and trained all the neural network models on `Google Colab` [1] with a 16GB Tesla P100 GPU. We always adopted the implementation from `torchvision`, modifying the input and output layers, or noted exceptions. We implemented the regression analyses using `R 4.0.5`, `lme4 1.1.26` [4], `Rstan 2.21.1` [100], `CmdStanR 0.3.0` [20], `brms 2.15.0` [9], and `tidybayes 2.3.1` [42].

Measure For the first two studies, we used **prediction accuracy** on the same test sets (i.e., the proportion of **correctly predicting participants' left or right judgment**) as the only measure for simplicity; other measures are correlated with prediction accuracy. Because the training of a neural network model was progressive, we selected the checkpoint with the smallest loss on the validation test as the final model to best describe the relationship between the input and output [60]. We used cross-entropy as the loss function or noted exceptions. The third study is a qualitative exploration.

Training Each judgment corresponds to an image and a label (Left or Right judgment). We initialized and trained a neural network model from scratch, applied batch normalization (batch size = 32, see Sec. 3.3 above), used an SGD optimizer with a momentum of 0.9, and employed a scheduler to decrease the initial learning

rate (0.001) after every five epochs at a rate of 70% (i.e., gamma = 0.7). We trained each model for 50 epochs in Study 1 and 20 epochs in Study 2, as we later noticed that most models converged early. We trained (or fitted) all the regression and neural network models based on the same training and validation sets, and applied each model to predict the judgments in the corresponding test set. We repeated the data splitting, initialization, training, and testing processes ten times to get ten samples of prediction accuracy. We then calculated the means and 95% bias-corrected and accelerated (BCa) bootstrap confidence intervals (CIs) [15, 17]. All code, datasets, results, and training logs are provided in supplementary materials.

4 STUDY 1: PREDICTING PERCEPTUAL JUDGMENTS

The first study focuses on predicting perceptual judgments. We ask **do deep neural networks better predict participants' correlation judgments** for correlation comparison in scatterplots? This study assessed a set of neural network architectures using Scatterplots100 in comparison with four regression analyses based on factors previously identified in the literature.

4.1 Previous regression analyses

The literature shows that correlation coefficients and visualization features may influence correlation comparison in scatterplots. To compare with these, we replicated their modeling approaches (e.g., [117]) by constructing four logistic regressions. Logistic regression is suitable for dichotomous responses, and it was used to model correlation comparison judgments in scatterplots [117]. We implemented both frequentist and Bayesian approaches.

Logistic regression (r) The first two regressions take pairs of correlation coefficients as the predictors (input), which are known factors that strongly affect correlation judgments [26, 89, 117]. They are also strongly correlated with the previously proposed visualization features [117]. In Wilkinson-Rogers-Pinheiro-Bates's notation [4, 81, 112], the regression formula is $LeftRight \sim r_{\text{RIGHT}} + r_{\text{LEFT}} + (1|ParticipantID)$. $LeftRight$ denotes the Left and Right judgments. r_{RIGHT} and r_{LEFT} denote the correlation coefficients of the two datasets; they are correlated with each other (collinearity), making the model coefficients *unidentifiable*, but the model itself is valid for prediction [67]. The $(1|ParticipantID)$ term denotes that each participant has a random intercept to account for the similarity in the judgments from the same participant [99]. **Bayesian logistic regression** shares the same notation and formula, but it follows a Bayesian approach and uses weakly informative priors. Bayesian statistics are emerging for human-computer interaction research (e.g., [43, 44]) and are robust to random errors (e.g., outliers). For these reasons, we included a Bayesian logistic regression.

Logistic regression (ellipse) uses the area of the prediction ellipse (see Appx. G for examples) as the predictors, a top-performing visualization feature in explaining correlation comparison judgments in scatterplots [117]. Similarly, the regression formula is $LeftRight \sim ellipseArea_{\text{RIGHT}} + ellipseArea_{\text{LEFT}} + (1|ParticipantID)$, and we constructed both frequentist and Bayesian models for comparison. We also had experimented with other visualization features based on the literature [69, 117] and a mixture model of multiple features. We found *ellipseArea* performed the best, and the mixture model failed to converge.

These four regressions are categorically different from the neural networks below. The first two use exact correlation coefficients that describe the scatterplots; the other two use the best knowledge about correlation judgments in scatterplots to make predictions. Both are *not* provided with visualization images. On the contrary, the neural networks operate on the same set of images but are *not* provided with *a priori* knowledge about correlation. The four regressions calibrate our expectation for a "good" neural network model. If the prediction accuracy is lower, then its estimate of correlation is likely poor, or the model has overfitted to noisy human judgments. If the accuracy is higher, the stimuli themselves may provide new visualization features that help predict human judgments.

4.2 Deep convolutional neural networks

We surveyed both the computer vision and visualization literature and collected a set of 30 neural network architectures, of which 29 are convolutional neural networks designed to solve human vision tasks. We started with simpler architectures and fewer trainable parameters, and gradually considered depth, width, image resolution, and invariance.

By experimenting with past and state-of-the-art architectures, we show which architectures can provide better predictions and learn lessons about model selection and architecture design.

Common approaches

Multilayer perceptron (MLP) [111] operates on all pixels simultaneously without convolution. Haehn et al. used an MLP as a baseline to test whether a convolutional neural network was necessary to solve graphical perception [23]. Following this logic, our MLP contains three layers of 4096, 4096, and 2 perceptrons, without any dropout layer.

AlexNet [49] is an architecture that first achieved state-of-the-art performance for image classification [2]. It is the simplest convolutional neural network in this section, consisting of five convolutional layers and two fully-connected hidden layers. This architecture is deeper than LeNet [52] used by Haehn et al. [23] and Giovannangeli et al. [21] for solving visualization tasks.

VGG [98] succeeded AlexNet by increasing depth to 11–19 convolutional layers with small convolutional kernels to extract more image features collectively. These are labeled VGG-11, VGG-13, VGG-16, and VGG-19. In previous studies, VGG-19 was most accurate for two visualization tasks: judging two-value ratios and counting dots added to scatterplots [23]. Similarly, VGG-16 performed well on counting edges and degrees in graph visualizations [21].

ResNet uses skip or residual connections to allow neural networks to be even deeper [28], with variants widened by a factor k to improve performance [119]. A ResNet-18 was trained to predict memorability and aesthetics scores of infographics and visualizations [19], and an altered version was used as a perceptual quality metric for

generic correlation perception in scatterplots [114]. We evaluated ResNet-18, ResNet-50, ResNet-152, and Wide ResNet-50-2.

DenseNet increases connectivity between layers rather than deepening or widening the network [33]. This design results in a high capacity but fewer learned features. We evaluated DenseNet-121, DenseNet-161, DenseNet-169, and DenseNet-201.

EfficientNet uniformly scales width, depth, and image resolution through a compound coefficient ϕ to improve training efficiency and accuracy [102]. Scaling up resolution allows it to extract fine-grained features from input images. We adapted the `Pytorch` implementation from Melas-Kyriazi et al. [68] and evaluated EfficientNet-B0, EfficientNet-B2, EfficientNet-B4, and EfficientNet-B6.

Variants and alternatives

Antialiased CNNs are more robust to input translations [120]. A regular convolutional neural network is more likely to produce incorrectly-different predictions for image features that have translated, which might affect perceptual judgment prediction. We denoted these variants by (antialiased).

FiLM modules condition neural network layers on additional inputs [77]. We used these inputs to inform the neural networks of participant IDs, much like the logistic regressions using participant IDs in random intercepts. In training, each input image and judgment has an added 210-dimensional one-hot vector representing which one of the 210 participants made that judgment. We attached FiLM modules to a feature extractor, and used 15 modules as they were the most accurate in the pilot studies. We denoted these variants by FiLM.

Bayesian CNNs introduce probabilistic distributions to neural network parameters, making them more robust to over-fitting [96]. We constructed Bayesian variants based on the implementation by Shridhar et al. [96], using the evidence lower bound (ELBO, a common loss function for probabilistic inference) and an Adamax optimizer [46].

VCC is our design to improve the training efficiency of VGG and integrate FiLM modules and Bayesian CNNs. VGG was trained slowly due to having over a hundred million parameters. Thus, we modified VGG-11 and created three VCCs. We added one convolutional layer with 128 filters to the first two convolutional layers of VGG-11 to create VCC-4, the first four to create VCC-5, and the first six excluding the third max-pooling layer to create VCC-7. All VCCs had one fully-connected layer as the classifier, resulting in fewer than 5 million parameters.

Finally, **VAE**s (or variational autoencoders) attempt to compress data into a latent space through an encoder/decoder structure [47]. VAEs were used to extract features in infographics and predict memorability and aesthetics scores [19], summarize representations in scatterplots [36], and code patterns in visualizations [54, 121]. We adopted the β−VAE architecture [29] from Jo and Seo [36], who also extracted correlation features from scatterplots. We used 64 latent features and trained for 100 epochs with an Adamax optimizer [46]. The classifier had one layer connecting the 64 features to the output.

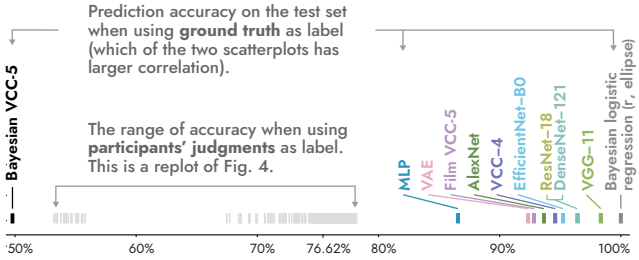

Figure 3: The sanity check shows that most of the neural networks can learn ground truth labels (i.e., which of the two scatterplots actually has higher correlation).

4.3 Training and interpreting

As remarked in Sec. 3.5, we initialized and trained ten models for each regression/architecture. We report the means and 95% confidence intervals of prediction accuracy in Fig. 4. To help interpret these results, we explain the sanity check and thresholds as follows.

Sanity check Human perceptual judgments are often noisy. If a model cannot learn the ground truth labels (i.e., which of the two scatterplots *actually* has a higher correlation coefficient) as an easier problem, it is impractical to expect them to predict participants' judgments. We therefore also trained the neural network models on the ground truth labels for a sanity check. To simplify this process, we only trained one model for the simplest architecture from one category. The uncertainty in accuracy should not exceed those from predicting participants' judgments (shown in Fig. 4). As such, we selected the two Bayesian logistic regressions (the frequentist logistic regressions did not converge), MLP, AlexNet, VGG-11, VCC-4, ResNet-18, DenseNet-121, EfficientNet-B0, VAE, and Bayesian VCC-5, reporting the results in Fig. 3.

The results show that most models can predict the ground truth (e.g., accuracy>90%); they can process the scatterplot images to learn correlation comparison. One exception is MLP, which shows an accuracy lower than others but higher than any models trained on participants' judgments. This indicates that MLP can learn correlation comparison from the images but may need more training epochs. Another exception is Bayesian VCC-5, showing an accuracy of around 50%; Bayesian VCC-5 cannot process these image.[5]

Thresholds We also note two bounds to help interpret the results from predicting participants' judgments. The first one is 55%, the guessing threshold for removing inattentive participants (see Fig. 2). Above this threshold, we surmise that a model learned features that help make a prediction. Below this threshold, a model likely fails to extract information from data. The second bound is the proportion of judgments where participants select the scatterplot of higher correlation; the average proportion is 76.62% for Scatterplots100. We surmise that surpassing this threshold is evidence that a model learns features that help correctly predict those judgments where participants do not select the scatterplot of higher correlation

[5]We speculate that the reason might be VCC-5 is not suitable for this dataset, but probabilistic weights prevent overfitting an inappropriate model. The latter two studies in this paper provide some limited evidence for this speculation.

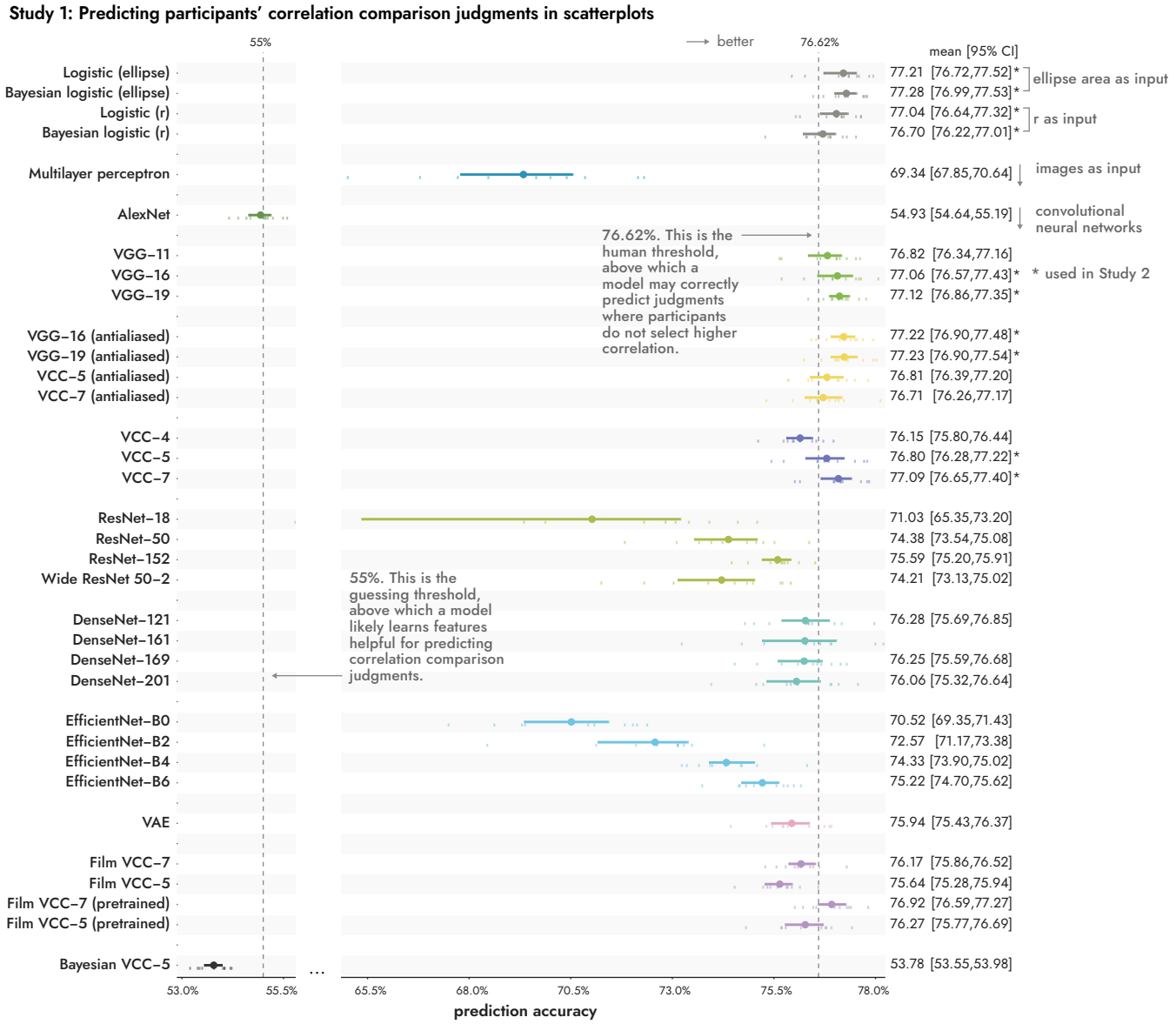

Figure 4: Study 1 shows a subset of neural network architectures has comparable prediction accuracy to regression analyses based on previously identified factors. We repeat the training, validation, and test processes ten times and report the means and 95% confidence intervals of prediction accuracy.

(e.g., when a feature is deceptive to humans [74], the model agrees with human judgments). We term this a "human" threshold.

4.4 Results

Do deep neural networks better predict participants' correlation judgments?

As shown in Fig. 4, VGGs, VCCs, and their variants (i.e., FiLM modules, Antialiased CNNs) have **comparable prediction accuracy** to the four regressions, outperforming the others. However, neither variant appears to improve prediction accuracy further. For FiLM VCC-7, it may indicate that participants' effects, while improving the training fit of a model, may not improve its test prediction; or the residual blocks (see below) inside FiLM modules [120] cause drawbacks. For Antialiased CNNs, we would expect improvements where translation equivariance is desired (such as if the visualization axes and points vary globally in a location within the image), but this appears not to be the case within this dataset. Additionally, the VGG-like architectures slightly improve prediction accuracy as the depth increases, and generally display less uncertainty in prediction accuracy.

Looking at others, the three architectures designed for large-scale image analysis—ResNet, DenseNet, and EfficientNet—achieved middling prediction accuracies and displayed more uncertainty. Both ResNet and DenseNet may worsen as the depth increases, while EfficientNet improves as the scaling factor increases. The literature shows that ResNet is less stable than VGG [71, 107]. Human behavioral data is often quite noisy, likely affecting ResNet more as a consequence. VAE also shows middling prediction accuracy, suggesting that its encoder learned representative features, but additional features might be necessary for improving accuracy. MLP appears able to make predictions based on 1D sequential pixels. It may reach a comparable accuracy after more training epochs, suggesting that convolutional layers are helpful. AlexNet and Bayesian VCC-5 seem unable to learn from these images. AlexNet can be trained to predict the ground truth, but it seems to struggle with noisy judgments, or the large kernels (11) in its first convolutional layer may have wiped out the fine-grained features in the scatterplots.

Among the four regressions, Logistic regression (ellipse) and Bayesian logistic regression (ellipse) yield slightly better prediction accuracy, corroborating with the literature that this feature better explains correlation comparison judgments in scatterplots [117]. The Bayesian approaches could be slightly less or more accurate, likely depending on the priors and fitting process.

Insights These results indicate that **most of the optimized architectures can predict participants' correlation comparison judgments in scatterplots**, and **a subset has comparable prediction accuracy to the best-performing models based on the factors and features proposed in the literature**. More complex architectures may be necessary for better prediction accuracy, but not necessarily a result of increasing depth, width, or image resolution. These quantitative performance measurements also provide insights into which architectures might provide better predictions for other visualizations and tasks.

5 STUDY 2: GENERALIZING THE PREDICTIONS

This study focuses on generalizing the model predictions to other related datasets [5]. In empirical studies, the generalizability provides hypothetical results without collecting human judgment and refitting the model. In particular, for deep convolutional neural networks, this generalizability will also address the concern about their sensitivity to small input changes [3]. As such, Study 2 asks **do the models better generalize predictions to other related datasets?** We examined two other datasets of correlation comparison judgments in scatterplots: increasing the number of data points (Scatterplots200) and presenting outliers (Scatterplots95+5) (see Fig. 1). Both vary in the data presented to participants and their judgment performance (see Appx. B).

5.1 Motivation

Study 2a: Scatterplots200 Previous studies of correlation perception often have a fixed number of data points [26, 89, 117]. As increasing the number of data points likely preserves certain scatterplot features, participants' judgments are probably similar [55]. Also, if a neural network model appeals to the features varying with the number of points (e.g., memorizing the coordinates), it may not generalize to such a new dataset. We therefore collected Scatterplots200, using the same experimental protocol as Scatterplots100 but increased the number of data points in each scatterplot to 200 (see Sec. 3.4 above).

Study 2b: Scatterplots95+5 Studies on correlation and cluster perception also suggest that noise and outliers in scatterplots likely affect participants' perceptions [85, 86, 93, 105]. To a neural network model, a change in pixel distributions might also affect its prediction. We therefore collected Scatterplots95+5, using the same protocol as the other two datasets. The difference is that five points were constantly located around 3.5 standard deviations from the mean of y (see Sec. 3.4 above). We chose five points because participants may ignore one or two points, but too many points would also present a comparable cluster, and five out of one hundred (5%) appear to be an acceptable threshold.

5.2 Training and interpreting

Therefore, we assessed the ten top-performing regressions (and architectures) from Study 1: the four logistic regressions, VGG-16, VGG-19, VCC-5, VCC-7, VGG-16 (antialiased), and VGG-19 (antialiased). In both Studies 2a and 2b, we repeated the data splitting and testing processes ten times. For each regression/architecture, we applied each of their ten models from Study 1 (e.g., the same estimated parameters) to predict each of the ten test sets, producing $10 \times 10 = 100$ samples of prediction accuracy.

To help understand the results, for each regression/architecture, we also trained ten models from scratch using the corresponding training and validation sets as analogies (we recomputed prediction ellipse). When compared to these analogies, a generalizable model should display the least decline in prediction accuracy. We calculated the means and confidence intervals of prediction accuracy and comparison, and reported them in Fig. 5.

5.3 Results

Do the models better generalize predictions to other related datasets?

In both Studies 2a and 2b, the VGG-19 and VGG-19 (antialiased) models are **the most generalizable** to the new datasets; they clearly outperform the others, especially those regression models. VGG-16 (antialiased) also has compelling performance. The models of other architectures (e.g., VCCs) were comparable when predicting a specific dataset (e.g., Scatterplots100 from Study 1), but they show notable declines in generalizability, likely due to overfitting. These neural network models operate on visualization images, informed about the changes in input.

The two regressions using correlation coefficients do not know the changes in visualizations. The other two regressions using *ellipseArea* know summarized changes. They essentially learned a difference threshold to predict the `Left` and `Right` judgments (see Appx. B). The optimal threshold varies with datasets; therefore, the previous models seem less generalizable to the new datasets. Bayesian logistic regression (ellipse) on Scatterplot95+5 seems an exception, but with much more uncertainty in prediction accuracy.

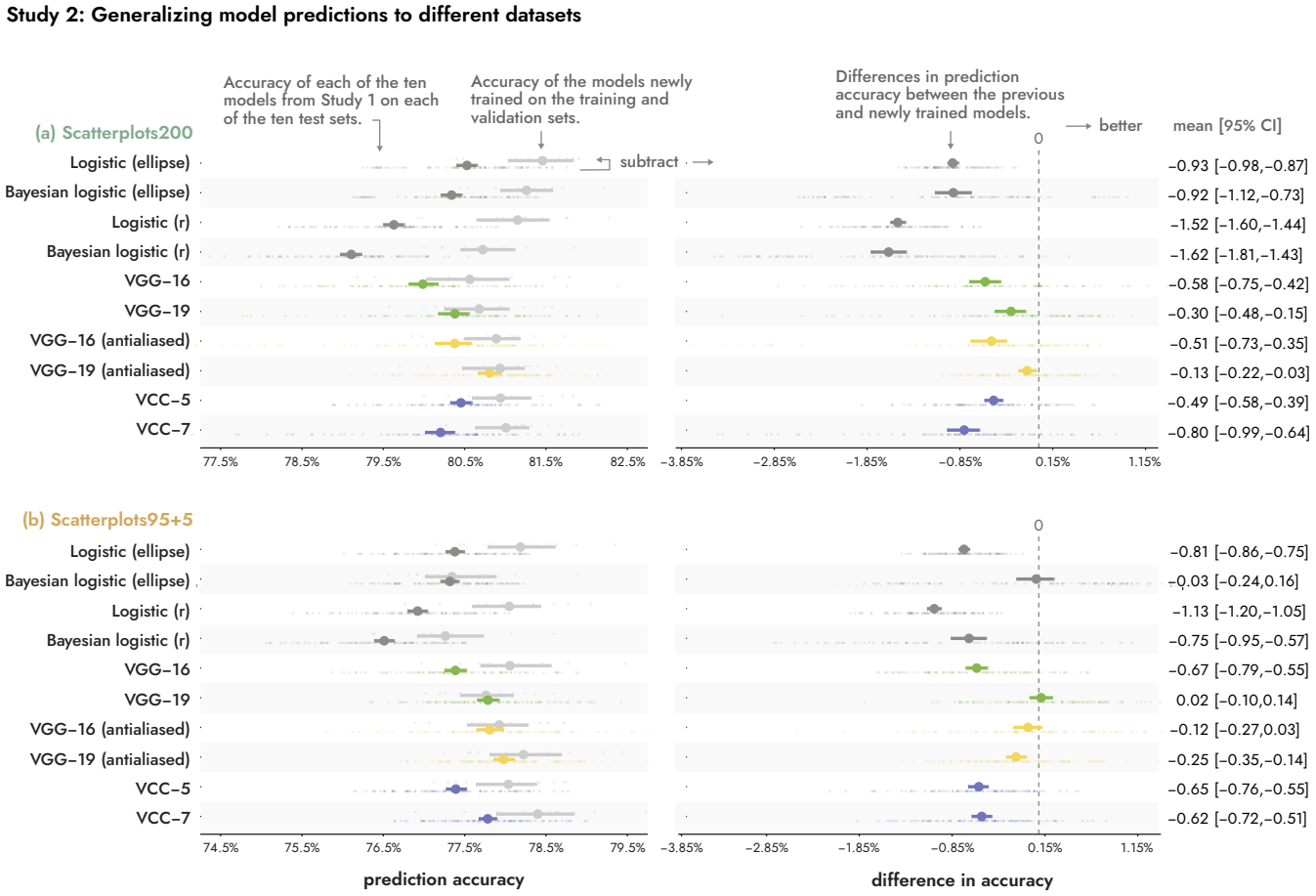

Figure 5: Studies 2a and 2b show VGG19 and VGG19 (antialiased) have the best generalizability on (a) Scatterplots200 and (b) Scatterplots95+5. Taking the top-performing regressions and neural network architectures from Study 1, we applied each of their ten models (i.e., the same estimated parameters) to predict each of the ten test sets, and compared them with those newly trained on the training and validation sets.

Between Studies 2a and 2b, the results are similar. The prediction accuracies are higher than those of Study 1; this is because participants selected scatterplots of higher correlation more often in these two datasets (see Appx. B). In other words, the "human" thresholds are higher (80.76% and 77.46%). However, most models from Study 1 show declines in prediction accuracy, compared to their analogies trained from scratch.

Insights This study partly answers an open question to correlation perception in scatterplots: **a deep convolutional neural network model like VGG-19 and VGG-19 (antialiased) could provide adequate predictions for new scatterplot designs without additional data** (i.e., collecting new human judgments). This is helpful in theorizing results or designing a new experiment. For example, we can use these models to infer human judgments of different visualizations or datasets, which might be used to automatically assess and optimize different design parameters based on model predictions of when humans may make mistakes. This study also suggests that these deep neural network models are not very sensitive to small input changes (e.g., a pixel change in dot size) in scatterplots. However, we are unclear about the causes of declines in their prediction accuracy. We do not know what a model learned and how these led to their prediction and generalization. These skepticisms invite Study 3 below.

6 STUDY 3: EXPLORING THE LEARNED FEATURES

Study 3 asks what features a convolutional neural network model learns to predict participants' correlation comparison judgments in scatterplots? This study shows an exploration of interpreting perceptual judgments in visualizations. We first conducted an error analysis to examine prediction bias and then extracted the features learned by a model. Because VGG-19 is representative of several architectures and shows the best prediction accuracy and generalizability above, we centered on a VGG-19 model trained on Scatterplots100 from Study 1. We reported the features of other models in Appx. D and supplementary materials.

6.1 Study 3a: Error analysis

We first conducted a small error analysis to examine the bias in the model predictions. We calculated the portion of Left and Right predictions. Of all correct predictions, 56.17% were Right, and among all incorrect predictions, 52.41% were Right. We then visually inspected the relationship between participants' judgments, the model predictions, and the confidence of the model (measured by probability) as follows.

The model makes more correct predictions as JND increases. This is prominent—the task is getting easier, and participants are more likely to make unambiguous (less noisy) judgments. We note that one JND is likely to be a threshold for the model. Above one JND, the model always predicts that participants select scatterplots of higher correlation ■ ■. Below one JND, the model predicts that participants select lower correlation sometimes ■ ■.

The model often has a high confidence score (e.g., >.7), indicating sufficient training. It is more confident in a correct prediction but can be very confident in an incorrect prediction. When its confidence is very low (e.g., [.5,.55]), the model appears to assign the two labels randomly.

Insights These results suggest that the model was sufficiently trained to predict participants' correlation comparison judgments in scatterplots; it also behaves reasonably and does not have strong biases in its predictions. These results support the following study of learned features.

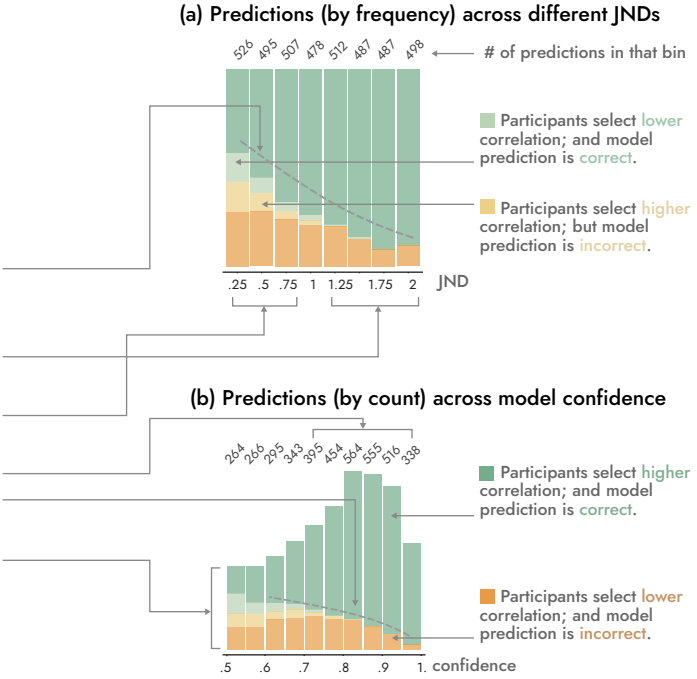

Figure 6: Study 3a reports an error analysis of a VGG19 model from Study 1, showing no strong bias.

6.2 Study 3b: Extracting features learned by a deep neural network model

Explaining a convolutional neural network model is an open problem. Here we resorted to a prevalent technique—*feature visualization* [72, 73].[6] This technique generates images that maximize the activation of a neuron, a filter (a channel), or a layer. The resulting images are considered *features* that the model "looks for," showing how the model builds up its understanding [73]. We chose filter-based optimization because it yielded the most interpretable results of better visual quality. We then generated feature visualizations for each filter in each convolutional layer for comprehensiveness.

Methods We used an SGD optimizer (steps = 2,048, learning rate = 0.15) with FFT parameterization and generated four diverse images (weight = 200) in the decorrelated space to avoid high-frequency artifacts [73]. We decided the hyperparameters based on the convergence and interpretability of the resulting feature images. We did not apply any transformation or preprocessing to preserve location and orientation information. We based our implementation on the `lucent` library [101] and obtained a total of 22,016 images for the VGG-19 model (e.g., 512 filters × 4 feature images per filter = 2,048 images for the last convolutional layer).

To analyze the resulting feature images, we combined k-means algorithms with a manual process to cluster the images for each convolutional layer. We first recursively applied k-means algorithms based on the Manhattan distance between image summaries (i.e., binning pixels) and gradually increased the number of clusters (usually 2 or 3 but up to 4). We terminated the recursion and expansion until the new clusters did not appear to share similar features. We then manually inspected each image in each cluster and relocated them into different clusters. This clustering process can be erroneous, as one image could ambiguously belong to more than one cluster, but the resulting clusters generally conceptualize features and describe their distributions. We consider each cluster a feature learned by the model.

Results We select one image from each cluster and report the proportion of the images in a cluster to all generated for that layer in Fig. 7. A larger cluster means that this layer learns redundant features. We omit non-interpretable clusters (e.g., all white images) here but reported them in Appx. D. We provide all the feature images of this and other models (see the beginning of Sec. 6) in supplementary materials.

Overall, we observe that the later layers learn more detailed features than the early layers. Max-pooling operations extract more features (e.g., Layer 10 vs. 14) while convolution operations aggregate features (e.g., Layer 14 vs. 17). These observations are consistent with prior findings and the nature of these operations.

❶ The first convolutional layer does not appear to learn any features, and the early convolutional layers seem to learn one or two ❷ 2D Gaussian kernels. This *Gaussian kernel* feature penetrates all the layers and evolves into ❸ the *ribbon* feature, which may be used to contrast different parts of the input (e.g., left and right).

As early as Layer 7, the model partially recognizes the *dot* feature. At Layer 14, the model is responsive to the *dot* feature, which disappears shortly and reappears at ❹ Layer 27. After Layer 27, ❺ this *dot* feature gradually dominates, meaning that these layers redundantly learned this feature. The early layers also appear to

[6] There are other approaches to explain a neural network model based on attribution in one input image. The results are less informative than feature visualizations. See Appx. E for an example.

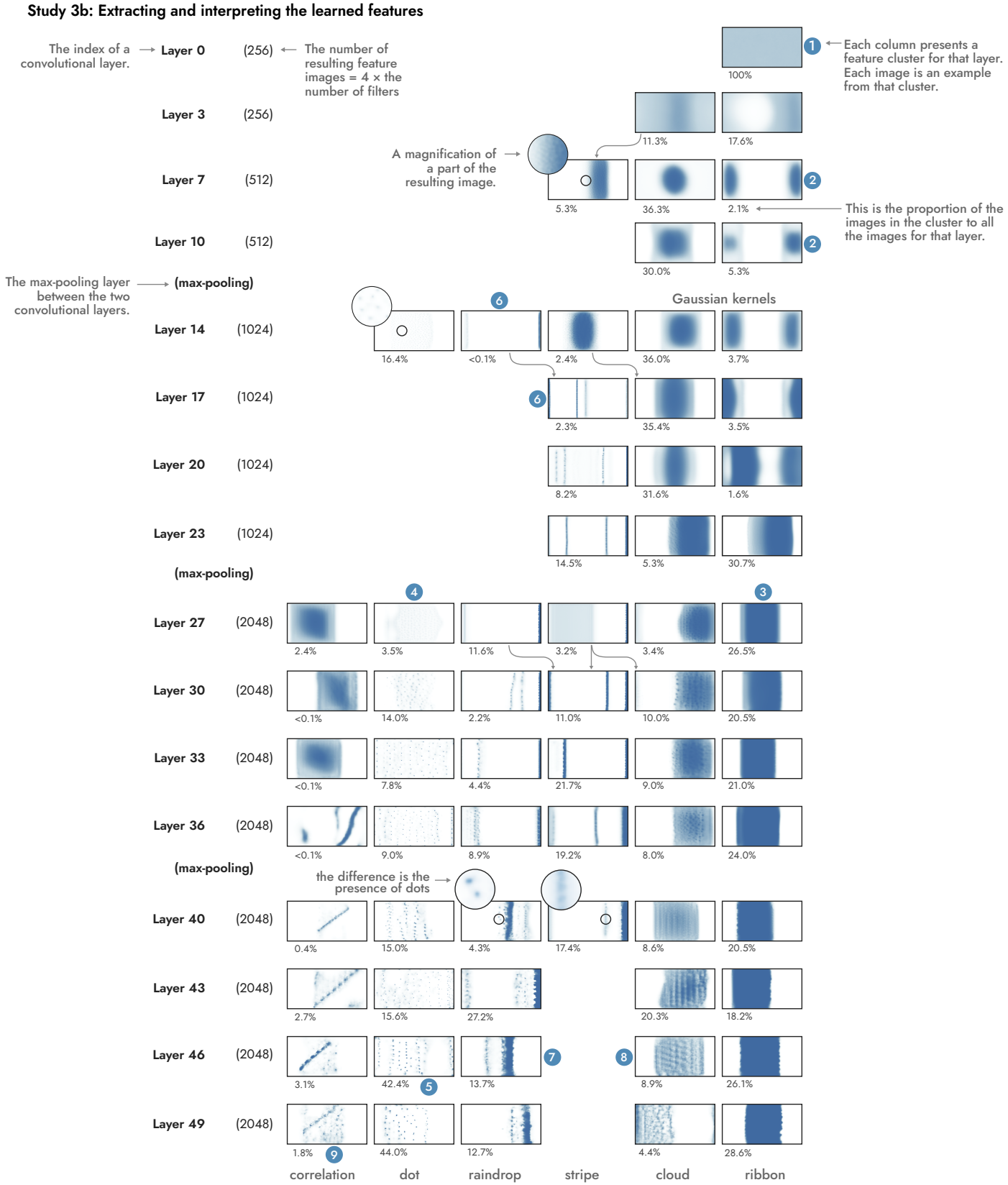

Figure 7: Study 3b extracts the features learned by a VGG19 model. This model was trained on Scatterplots100 in Study 1. Larger images are available in supplementary materials.

learn ❻ the *stripe* feature, which is preserved throughout all the subsequent convolutional layers. This feature is later combined with the *dot* feature and yields ❼ the *raindrop* and ❽ *cloud* features. Finally, we also note ❾ the *correlation* feature, reflected as one rotated Gaussian kernel in the middle layers (e.g., Layer 27, the first column) and a straight line with a slope in the later layers (e.g., Layers 40 - 49).

6.3 Discussion

What features does a convolutional neural network model learn to predict correlation judgments in scatterplots?

It appears that the VGG-19 model learns ***Gaussian kernel***, ***dot***, ***stripe***, ***ribbon***, ***correlation***, and **other local features** when trained to predict participants' correlation comparison judgments. The *Gaussian kernel* feature may allude to an estimation of pixel density. It also corroborates that prediction ellipses are an approximation or intermediate of the perceptual features that participants might look for [87, 88]; however, other more detailed features in the scatterplots would improve prediction accuracy. The *correlation* feature may indicate that the cloud shape and regression slope influence both the model predictions and participant judgments. The *dot* feature may imply that the model counts pixels or dots, which can be construed as some numerosity or magnitude estimation [53, 103, 115]. The *ribbon* feature may imply a comparison of the two scatterplots. The *Gaussian kernel* feature may explain that a prediction ellipse does not provide enough information; additional density information is necessary for model generalizability. Other features like *cloud* and *raindrop* may imply an estimation of dot entropy, which were previously considered plausible visual proxies for correlation comparison judgments in scatterplots [87, 88].

Most features share vertical patterns, which might relate to the data generation process and the definition of Pearson's correlation. The data generation process transforms y-coordinates towards the target Pearson's correlation coefficient (r), which is defined by the vertical distance to the regression line when x-coordinates are fixed. These vertical features may be caused by the model trying to attribute error to vertical distance.

We summarize the features of other models (see Appx. D) as follows. VGG-19 models trained on Scatterplots200, Scatterplots95+5, or ground truth share similar features but vary in their distributions. For example, the proportion of the *correlation* feature is much higher, especially for Scatterplots95+5 and ground truth. VGG-19 (antialiased) results in similar but less detailed features, consistent with the blurring operations in this architecture. VCC does not yield any features related to local properties (e.g., *dot*), but instead has only features related to location (e.g., *ribbon*). ResNet-18 is responsive to fine features in its early layers and generally yields features with higher spatial frequency, which corroborates prior findings that ResNet tends to learn non-robust features [3, 71, 107].

Insights This study shows that **a model directly trained on the visualization images and participant judgments provides interpretable visualization features.** This feature analysis and the auxiliary gradient analysis (see Appx. E) reveal clues to model behavior. It may extract and compares density information to make a prediction, while other fine features may provide additional information to improve prediction accuracy. These features may also partially explain why the model seems generalizable on Scatterplots200 and Scatterplots95+5: the features extracted are similar, and the model does not seem to memorize pixel coordinates to make predictions.

This study also provides partial answers to one open question for visualization perception: we show an approach to extract potential perceptual features systematically [35]. Several of the features are consistent with prior studies, validating our approach. More broadly, we also demonstrate how a single neural network model optimized on one dataset can reflect findings about visual features from a series of prior works. Beyond validating previously hypothesized features, the models also suggest new features of interest. These features may provide inspiration to theorize correlation perception in scatterplots and to design new models. For example, we can first compute the most important features (e.g., perhaps they are the *correlation* and *dot* features) and extract them from visualization images, following up with a logistic regression based on them.

7 GENERAL DISCUSSION

First, a reminder that this research does not attempt to suggest that deep neural networks are a biologically plausible model of human vision, and they are likely not [6]. Nor do we encourage a shortcut to study visualization perception without rigorous human-subject experiments; in the absence of adequate prior knowledge and caution, mindlessly fitting complex models to complex data will undermine the scientific community and engender false discoveries.

7.1 What benefits *might* DNNs bring?

In comparison to previous regression analyses, we show that deep convolutional neural networks, which process visualization images directly, can provide comparable prediction, better generalization, and new interpretable features. They do not require explicit researcher hypotheses (which can be fallacious) or eye-tracking data (which studies have shown can be irrelevant to participants' judgments [34]). These models might help us in forming ideas about current perceptual judgement data, including the design of future human subject experiments.

Of course, these models do not solve all problems we are curious about in visualization perception, but this direction of starting with visualization images and building our understanding backward seems promising. It may help researchers further ideate about fundamental theory for visualization perception and tasks. Besides deep convolutional neural networks, other pixel-based models may also be feasible; deep convolutional neural networks are a convenient choice. Similarly, our goal here is not to discourage factor-based regression analyses. They are essential for systematically investigating the effects of small sets of experimental factors (e.g., mark orientation [58] and point size).

7.2 Generalizability and future opportunities

Studies 2 and 3 suggest that the neural network models might be generalizable to other correlation perception datasets. As some models maintain performance with subtle changes in axes and

points, they might be able to predict scatterplots of *slightly* different parameters (e.g., dot color, marker size, fewer dots) without additional training or new data. However, we stress that inferring further generalization to other visualizations or tasks is challenging. As such, our research invites many possible follow-ups. One study could include scatterplots of negative correlation or different slopes. Human judgments are likely similar [26], but it may be more challenging to train a model to predict both positively and negatively correlated data. Another study could gradually drop dots or outliers and observe changes in human and model behaviors; such a procedure may reveal dots that are key to model prediction.

Our approach is possibly generalizable to other visualization perception studies. It does not require *a priori* hypotheses, and the number of images (judgments) used is middling among contemporary research. For comparison, Harrison et al. collected ~300,000 judgments [26], and Jardine et al. collected ~5,000 judgments [35]. It is useful to explore how model performance varies with sample size (e.g., to find a minimal sample size with acceptable performance).

7.3 Limitations

We acknowledge the limitations in our choices of task, design, and models. We considered only one task and three different datasets. Other visualization tasks and visual channels, such as estimating the mean, colors, or mark shape, could lead to different human and model performance. Our results might be specific to our data generation process (e.g., how we generated the five outliers). We only considered convolutional neural networks and investigated a small subset of all previously proposed architectures. There are many other deep learning architectures to explore, like using CapsuleNet to model hierarchical relationships in visualizations [90], generative adversarial networks (GAN) to model numeric responses, and the recent proposal of vision transformers (ViT) to preserve spatial information [24]. Similarly, further exploration of probabilistic neural networks may help address the limited sample size and noise in the data. We also recognize the limited interpretability of a neural network model but anticipate that future advances in interpretable machine learning and explainable artificial intelligence will help overcome this problem (e.g., [30, 31, 110]) and provide more insights into visualization perception.

The long training and computation time might be a practical limitation, although they largely depend on image size and hardware. Most models were sufficiently trained within two to four hours, but ten repeats cost more than a day. Computing feature visualizations for all filters may take a few days for a large architecture like VGG-19. Previous works show that channel activation is often power law distributed [31]. It might be sufficient only to investigate top-activated (e.g., 3%) neurons/filters to reduce computation time.

7.4 Predictive modeling for visualization

In principle, a model predicting human judgments can help design new visualizations by performing an automatic evaluation, optimizing the choice of graphical encoding, or informing designers of a misleading case. Given our assessment of a set of deep neural network architectures in Study 1, researchers can use our results for a preliminary model selection to identify a likely effective model and avoid repetitive efforts.

In practice, the model generalizability shown in Studies 2 and 3 is limited and only hints at the possibility to build more generalizable models (e.g., using very large corpora of visualization images and human judgments on different tasks). While researchers might have to collect such datasets, designers and practitioners could use the trained models or fine-tune the models on a small set if they generalized. Our study and previous studies (e.g., [21, 23]) show the effectiveness and limitations of current architecture designs. This points to a need to design neural network architectures and explanation methods that are *specific* to visualizations. Visualization images have different properties and patterns than natural images. They contain more abstract and repetitive features, and they rely on spatial information more than texture to convey information. Previous studies also show that fine-tuning neural network models pre-trained on natural images is not as effective as training them from scratch [23]. From vision scientists' perspective, humans may extract the same statistics information from visualization and natural images [25], but there seem to be one or more adaptation processes between them [41]. These suggest that a possible solution is to consider the shared and different statistics information between natural and visualization images and how people perceive and process this information. This will likely remove the redundancy in learned features and result in architectures with fewer parameters but improved prediction accuracy. Our studies provide evidence and insights for these open problems and hope to inspire future work on predictive modeling for visualization.

8 CONCLUSION

This manuscript reports insights from using deep convolutional neural networks for visualization perception research through three studies of correlation comparison judgments in scatterplots. The first study assessed a collection of thirty neural network architectures, showing that the deep convolutional neural network models can have equivalent prediction accuracy compared with the best-performing regression analyses in the literature. The second study applied the trained models to two related yet different datasets, showing that deep neural networks have better generalizability. The third study computed features learned by a convolutional neural network model, and revealed how the model builds its understanding and extracts image features to make a prediction; these features provide new clues to correlation perception. Together, this series of three studies show the emerging prospect of using deep neural networks to predict, generalize, and construe perceptual judgments for visualization perception research.

ACKNOWLEDGMENTS

This research was supported by NSF IIS-2107409, NSF IIS-1815587, the National Natural Science Foundation of China (No. 62202217), and NSF 2127309 to the Computing Research Association for the CIFellows Project. The authors thank Brown Visual Computing Group (especially Daniel Ritchie) and Kaiyu Zheng for their feedback, Jing Qian, Rocket Drew, and Yuan (Charles) Cui for their help with the manuscript, and Mi Feng for inspiring this work. The authors also thank Steve Haroz, Alex Kale, and Christie Nothelfer for a discussion on the perception of natural and visualization images at the VIS×VISION Slack workspace.

REFERENCES

[1] 2017. Colaboratory - Google Research. https://colab.research.google.com/ Online; accessed 15 Feb 2022.

[2] Md Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C. Van Esesn, Abdul A. S. Awwal, and Vijayan K. Asari. 2018. The history began from AlexNet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164* (2018). http://arxiv.org/abs/1803.01164

[3] Aharon Azulay and Yair Weiss. 2019. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research* 20, 184 (2019), 1–25. http://jmlr.org/papers/v20/19-519.html

[4] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software* 67, 1 (2015), 1–48. https://doi.org/10.18637/jss.v067.i01

[5] Moshe Ben-Akiva and Denis Bolduc. 1987. *Approaches to model transferability and updating: the combined transfer estimator.* Département d'économique, Université Laval.

[6] Jeffrey S. Bowers, Gaurav Malhotra, Marin Dujmović, Milton Llera Montero, Christian Tsvetkov, Valerio Biscione, Guillermo Puebla, Federico Adolfi, John E. Hummel, Rachel F. Heaton, and et al. 2022. Deep Problems with Neural Network Models of Human Vision. *Behavioral and Brain Sciences* (2022), 1–74. https://doi.org/10.1017/S0140525X22002813

[7] David M. Boynton. 2000. The psychophysics of informal covariation assessment: Perceiving relatedness against a background of dispersion. *Journal of Experimental Psychology: Human Perception and Performance* 26, 3 (2000), 867–876. https://doi.org/10.1037/0096-1523.26.3.867

[8] Eli T. Brown, Alvitta Ottley, Helen Zhao, Quan Lin, Richard Souvenir, Alex Endert, and Remco Chang. 2014. Finding Waldo: Learning about Users from their Interactions. *IEEE TVCG* 20, 12 (2014), 1663–1672. https://doi.org/10.1109/TVCG.2014.2346575

[9] Paul-Christian Bürkner et al. 2017. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software* 80, 1 (2017), 1–28. https://doi.org/10.18637/jss.v080.i01

[10] Zoya Bylinskii, Nam Wook Kim, Peter O'Donovan, Sami Alsheikh, Spandan Madan, Hanspeter Pfister, Fredo Durand, Bryan Russell, and Aaron Hertzmann. 2017. Learning Visual Importance for Graphic Designs and Data Visualizations. In *ACM UIST*. 57–69. https://doi.org/10.1145/3126594.3126653

[11] Zhutian Chen, Yun Wang, Qianwen Wang, Yong Wang, and Huamin Qu. 2020. Towards Automated Infographic Design: Deep Learning-based Auto-Extraction of Extensible Timeline. *IEEE TVCG* 26, 1 (2020), 917–926. https://doi.org/10.1109/TVCG.2019.2934810

[12] Radoslaw M. Cichy and Daniel Kaiser. 2019. Deep Neural Networks as Scientific Models. *Trends in Cognitive Sciences* 23, 4 (2019), 305–317. https://doi.org/10.1016/j.tics.2019.01.009

[13] Michael Correll and Jeffrey Heer. 2017. Regression by Eye: Estimating Trends in Bivariate Visualizations. In *ACM CHI*. 1387–1396. https://doi.org/10.1145/3025453.3025922

[14] Victor Dibia and Cagatay Demiralp. 2019. Data2Vis: Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks. *IEEE CGA* 39, 5 (2019), 33–46. https://doi.org/10.1109/MCG.2019.2924636

[15] Thomas J. Diciccio and Joseph P. Romano. 1988. A Review of Bootstrap Confidence Intervals. *Journal of the Royal Statistical Society: Series B (Methodological)* 50, 3 (1988), 338–354. https://doi.org/10.1111/j.2517-6161.1988.tb01732.x

[16] Michael E. Doherty, Richard B. Anderson, Andrea M. Angott, and Dale S. Klopfer. 2007. The perception of scatterplots. *Perception & Psychophysics* 69, 7 (2007), 1261–1272. https://doi.org/10.3758/BF03193961

[17] Pierre Dragicevic. 2016. Fair Statistical Communication in HCI. In *Modern Statistical Methods for HCI*, Judy Robertson and Maurits Kaptein (Eds.). Springer International Publishing, Cham, 291–330. https://doi.org/10.1007/978-3-319-26633-6_13

[18] Marin Dujmović, Gaurav Malhotra, and Jeffrey Bowers. 2020. What do adversarial images tell us about human vision? *bioRxiv* (2020). 10.7554/eLife.55978.

[19] Xin Fu, Yun Wang, Haoyu Dong, Weiwei Cui, and Haidong Zhang. 2019. Visualization Assessment: A Machine Learning Approach. In *IEEE VIS short paper*. 126–130. https://doi.org/10.1109/VISUAL.2019.8933570

[20] Jonah Gabry and Rok Češnovar. 2020. CmdStanR: the R interface to CmdStan. https://mc-stan.org/users/interfaces/cmdstan

[21] Loann Giovannangeli, Romain Bourqui, Romain Giot, and David Auber. 2020. Toward automatic comparison of visualization techniques: Application to graph visualization. *Visual Informatics* 4, 2 (2020), 86–98. https://doi.org/10.1016/j.visinf.2020.04.002 Visualization Meets AI workshop at PacificVis.

[22] Michael Gleicher, Michael Correll, Christine Nothelfer, and Steven Franconeri. 2013. Perception of average value in multiclass scatterplots. *IEEE TVCG* 19, 12 (2013), 2316–2325.

[23] Daniel Haehn, James Tompkin, and Hanspeter Pfister. 2019. Evaluating 'Graphical Perception' with CNNs. *IEEE TVCG* 25, 1 (2019), 641–650. https://doi.org/10.1109/TVCG.2018.2865138

[24] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. 2020. A survey on visual transformer. *arXiv preprint* (2020). arXiv:2012.12556

[25] Steve Haroz and Kwan-Liu Ma. 2006. Natural visualizations. (2006).

[26] Lane Harrison, Fumeng Yang, Steven Franconeri, and Remco Chang. 2014. Ranking Visualizations of Correlation Using Weber's Law. *IEEE TVCG* 20, 12 (2014), 1943–1952. https://doi.org/10.1109/TVCG.2014.2346979

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *IEEE ICCV*. 1026–1034. https://doi.org/10.1109/ICCV.2015.123

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE CVPR*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[29] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*. https://openreview.net/forum?id=Sy2fzU9gl

[30] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. 2019. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE TVCG* 25, 8 (2019), 2674–2693. https://doi.org/10.1109/TVCG.2018.2843369

[31] Fred Hohman, Haekyu Park, Caleb Robinson, and Duen Horng Polo Chau. 2020. Summit: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations. *IEEE TVCG* 26, 1 (2020), 1096–1106. https://doi.org/10.1109/TVCG.2019.2934659

[32] Kevin Hu, Michiel A. Bakker, Stephen Li, Tim Kraska, and César Hidalgo. 2019. VizML: A Machine Learning Approach to Visualization Recommendation. In *ACM CHI*. 1–12. https://doi.org/10.1145/3290605.3300358

[33] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely Connected Convolutional Networks. In *IEEE CVPR*. 2261–2269. https://doi.org/10.1109/CVPR.2017.243

[34] Laurent Itti and Christof Koch. 2001. Computational modelling of visual attention. *Nature reviews neuroscience* 2, 3 (2001), 194–203. https://doi.org/10.1038/35058500

[35] Nicole Jardine, Brian D. Ondov, Niklas Elmqvist, and Steven Franconeri. 2020. The Perceptual Proxies of Visual Comparison. *IEEE TVCG* 26, 1 (2020), 1012–1021. https://doi.org/10.1109/TVCG.2019.2934786

[36] Jaemin Jo and Jinwook Seo. 2019. Disentangled Representation of Data Distributions in Scatterplots. In *IEEE VIS short paper*. 136–140. https://doi.org/10.1109/VISUAL.2019.8933670

[37] Kamila Jozwik, Nikolaus Kriegeskorte, Radoslaw Martin Cichy, and Marieke Mur. 2019. Deep convolutional neural networks, features, and categories perform similarly at explaining primate high-level visual representations. (2019). https://doi.org/10.32470/CCN.2018.1232-0

[38] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeong-in Hwang, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. 2017. ChartSense: Interactive Data Extraction from Chart Images. In *ACM CHI*. 6706–6717. https://doi.org/10.1145/3025453.3025957

[39] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. DVQA: Understanding Data Visualizations via Question Answering. In *IEEE/CVF CVPR*. 5648–5656. https://doi.org/10.1109/CVPR.2018.00592

[40] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint* (2017). arXiv:1710.07300

[41] Alex Kale and Jessica Hullman. 2019. Adaptation and learning priors in visual inference. In *VisxVision workshop at IEEE VIS*.

[42] Matthew Kay. 2021. *tidybayes: Tidy Data and Geoms for Bayesian Models.* https://doi.org/10.5281/zenodo.1308151

[43] Matthew Kay and Jeffrey Heer. 2016. Beyond Weber's Law: A Second Look at Ranking Visualizations of Correlation. *IEEE TVCG* 22, 1 (2016), 469–478. https://doi.org/10.1109/TVCG.2015.2467671

[44] Matthew Kay, Gregory L. Nelson, and Eric B. Hekler. 2016. Researcher-Centered Design of Statistics: Why Bayesian Statistics Better Fit the Culture and Incentives of HCI. In *ACM CHI*. 4521–4532. https://doi.org/10.1145/2858036.2858465

[45] Been Kim, Emily Reif, Martin Wattenberg, and Samy Bengio. 2019. Do neural networks show gestalt phenomena? an exploration of the law of closure. *arXiv preprint* 2, 8 (2019). arXiv:1903.01069.

[46] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*. http://arxiv.org/abs/1412.6980

[47] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. *ICLR*. http://arxiv.org/abs/1312.6114

[48] Nikolaus Kriegeskorte. 2015. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual review of vision science* 1 (2015), 417–446. https://doi.org/10.1146/annurev-vision-082114-035447

[49] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60, 6 (May 2017), 84–90. https://doi.org/10.1145/3065386

[50] Jonas Kubilius, Stefania Bracci, and Hans P. Op de Beeck. 2016. Deep Neural Networks as a Computational Model for Human Shape Sensitivity. *PLOS Computational Biology* 12, 4 (04 2016), 1–26. https://doi.org/10.1371/journal.pcbi.1004896

[51] Thomas W. Lauer and Gerald V. Post. 1989. Density in scatterplots and the estimation of correlation. *Behaviour & Information Technology* 8, 3 (1989), 235–244. https://doi.org/10.1080/01449298908914554

[52] Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. https://doi.org/10.1109/5.726791

[53] Tali Leibovich, Naama Katzin, Maayan Harel, and Avishai Henik. 2017. From "sense of number" to "sense of magnitude": The role of continuous magnitudes in numerical cognition. *Behavioral and Brain Sciences* 40 (2017).

[54] Fritz Lekschas, Brant Peterson, Daniel Haehn, Eric Ma, Nils Gehlenborg, and Hanspeter Pfister. 2020. Peax: Interactive Visual Pattern Search in Sequential Data Using Unsupervised Deep Representation Learning. *CGF* 39, 3 (2020), 167–179. https://doi.org/10.1111/cgf.13971

[55] Jing Li, Jean-Bernard Martens, and Jarke J van Wijk. 2010. Judging Correlation from Scatterplots and Parallel Coordinate Plots. *Information Visualization* 9, 1 (2010), 13–30. https://doi.org/10.1057/ivs.2008.13

[56] Grace W. Lindsay. 2020. Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. *Journal of cognitive neuroscience* (February 2020), 1–15. https://doi.org/10.1162/jocn_a_01544

[57] Drew Linsley, Sven Eberhardt, Tarun Sharma, Gupta Gupta, and Thomas Serre. 2017. What are the Visual Features Underlying Human Versus Machine Vision?. In *IEEE ICCV Workshops*. 2706–2714. https://doi.org/10.1109/ICCVW.2017.331

[58] Tingting Liu, Xiaotong Li, Chen Bao, Michael Correll, Changehe Tu, Oliver Deussen, and Yunhai Wang. 2021. Data-Driven Mark Orientation for Trend Estimation in Scatterplots. In *ACM CHI*. Article 473, 16 pages. https://doi.org/10.1145/3411764.3445751

[59] Min Lu, Chufeng Wang, Joel Lanir, Nanxuan Zhao, Hanspeter Pfister, Daniel Cohen-Or, and Hui Huang. 2020. *Exploring Visual Information Flows in Infographics*. 1–12. https://doi.org/10.1145/3313831.3376263

[60] Thomas Viehmann Luca Pietro Giovanni Antiga, Eli Stevens. 2020. *Deep Learning with PyTorch*. Manning Publications Co.

[61] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. DeepEye: Towards Automatic Data Visualization. In *International Conference on Data Engineering*. 101–112. https://doi.org/10.1109/ICDE.2018.00019

[62] Yuxin Ma, Anthony K. H. Tung, Wei Wang, Xiang Gao, Zhigeng Pan, and Wei Chen. 2020. ScatterNet: A Deep Subjective Similarity Model for Visual Analysis of Scatterplots. *IEEE TVCG* 26, 3 (2020), 1562–1576. https://doi.org/10.1109/TVCG.2018.2875702

[63] Najib J. Majaj and Denis G. Pelli. 2018. Deep learning—Using machine learning to study biological vision. *Journal of Vision* 18, 13 (12 2018), 2–2. https://doi.org/10.1167/18.13.2 arXiv:https://arvojournals.org/arvo/content_public/journal/jov/937687/i1534-7362-18-13-2.pdf

[64] Justin Matejka, Fraser Anderson, and George Fitzmaurice. 2015. Dynamic Opacity Optimization for Scatter Plots. In *ACM CHI*. 2707–2710. https://doi.org/10.1145/2702123.2702585

[65] Masakazu Matsugu, Katsuhiko Mori, Yusuke Mitari, and Yuji Kaneda. 2003. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks* 16, 5 (2003), 555–559. https://doi.org/10.1016/S0893-6080(03)00115-1

[66] Angela Mayhua, Erick Gomez-Nieto, Jeffrey Heer, and Jorge Poco. 2018. Extracting Visual Encodings from Map Chart Images with Color-Encoded Scalar Values. In *SIBGRAPI*. 142–149. https://doi.org/10.1109/SIBGRAPI.2018.00025

[67] Richard McElreath. 2016. *Statistical rethinking: A Bayesian course with examples in R and Stan*. CRC press. https://doi.org/10.1201/9781315372495

[68] Luke Melas-Kyriazi. 2019. EfficientNet PyTorch. https://github.com/lukemelas/EfficientNet-PyTorch

[69] Joachim Meyer and David Shinar. 1991. Perceiving Correlations from Scatterplots. *Proceedings of the Human Factors Society Annual Meeting* 35, 20, 1537–1540. https://doi.org/10.1177/154193129103502025

[70] Luana Micallef, Gregorio Palmas, Antti Oulasvirta, and Tino Weinkauf. 2017. Towards Perceptual Optimization of the Visual Design of Scatterplots. *IEEE TVCG* 23, 6 (2017), 1588–1599. https://doi.org/10.1109/TVCG.2017.2674978

[71] Reiichiro Nakano. 2019. A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features': Adversarially Robust Neural Style Transfer. *Distill* (2019). https://doi.org/10.23915/distill.00019.4 https://distill.pub/2019/advex-bugs-discussion/response-4.

[72] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2016. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *Visualization for Deep Learning workshop at ICML* (2016). arXivpreprintarXiv:1602.03616

[73] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature Visualization. *Distill* (2017). https://doi.org/10.23915/distill.00007 https://distill.pub/2017/feature-visualization.

[74] Brian D. Ondov, Fumeng Yang, Matthew Kay, Niklas Elmqvist, and Steven Franconeri. 2020. Revealing Perceptual Proxies with Adversarial Examples. *IEEE TVCG* 25, 1 (2020). https://doi.org/10.1109/TVCG.2020.3030429

[75] Jorge Piazentin Ono, Ray Sungsoo Hong, Claudio T Silva, and Juliana Freire. 2018. Why should we teach machines to read charts made for humans? https://vgc.poly.edu/~jhenrique/files/chi2019_workshop_ML_Evaluate_Vis.pdf

[76] Alvitta Ottley, Roman Garnett, and Ran Wan. 2019. Follow The Clicks: Learning and Anticipating Mouse Interactions During Exploratory Data Analysis. *CGF* 38, 3 (2019), 41–52. https://doi.org/10.1111/cgf.13670

[77] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: Visual Reasoning with a General Conditioning Layer. *AAAI* 32, 1 (Apr. 2018). https://ojs.aaai.org/index.php/AAAI/article/view/11671

[78] Joshua C. Peterson, Joshua T. Abbott, and Thomas L. Griffiths. 2018. Evaluating (and Improving) the Correspondence Between Deep Neural Networks and Human Representations. *Cognitive Science* 42, 8 (2018), 2648–2669. https://doi.org/10.1111/cogs.12670 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.12670

[79] Daniel Pineo and Colin Ware. 2008. Neural Modeling of Flow Rendering Effectiveness. *ACM Transactions on Applied Perception* 7, 3, Article 20 (June 2008), 15 pages. https://doi.org/10.1145/1773965.1773971

[80] Daniel Pineo and Colin Ware. 2012. Data Visualization Optimization via Computational Modeling of Perception. *IEEE TVCG* 18, 2 (2012), 309–320. https://doi.org/10.1109/TVCG.2011.52

[81] Jose Pinheiro, Douglas Bates, Saikat DebRoy, Deepayan Sarkar, and R Core Team. 2021. *nlme: Linear and Nonlinear Mixed Effects Models*. https://CRAN.R-project.org/package=nlme R package version 3.1-152.

[82] Jorge Poco and Jeffrey Heer. 2017. Reverse-Engineering Visualizations: Recovering Visual Encodings from Chart Images. *CGF* 36, 3 (2017), 353–363. https://doi.org/10.1111/cgf.13193

[83] Revanth Reddy, Rahul Ramesh, Ameet Deshpande, and Mitesh M. Khapra. 2019. FigureNet : A Deep Learning model for Question-Answering on Scientific Plots. In *The International Joint Conference on Neural Networks*. 1–8. https://doi.org/10.1109/IJCNN.2019.8851830

[84] Revanth Reddy, Rahul Ramesh, Ameet Deshpande, and Mitesh M Khapra. 2019. FigureNet: A Deep Learning model for Question-Answering on Scientific Plots. In *IJCNN*. 1–8. http://arxiv.org/abs/1806.04655

[85] Daniel Reimann, Christine Blech, and Robert Gaschler. 2020. Visual model fit estimation in scatterplots and distribution of attention: Influence of slope and noise level. *Experimental Psychology* 67, 5 (2020), 292–302. https://doi.org/10.1027/1618-3169/a000499

[86] Daniel Reimann, Christine Blech, Nilam Ram, and Robert Gaschler. 2021. Visual Model Fit Estimation in Scatterplots: Influence of Amount and Decentering of Noise. *IEEE TVCG* 27, 9 (2021), 3834–3838. https://doi.org/10.1109/TVCG.2021.3051853

[87] Ronald A. Rensink. 2017. The nature of correlation perception in scatterplots. *Psychonomic Bulletin & Review* 24 (2017), 776–797. Issue 3. https://doi.org/10.3758/s13423-016-1174-7

[88] Ronald A Rensink. 2022. Visual features as carriers of abstract quantitative information. *Journal of Experimental Psychology: General* (2022).

[89] Ronald A. Rensink and Gideon Baldridge. 2010. The Perception of Correlation in Scatterplots. *CGF* 29, 3, 1203–1210. https://doi.org/10.1111/j.1467-8659.2009.01694.x

[90] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing between Capsules. In *NeurIPS*. Curran Associates Inc., 3859–3869.

[91] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. 2011. ReVision: Automated Classification, Analysis and Redesign of Chart Images. In *ACM UIST*. 393–402. https://doi.org/10.1145/2047196.2047247

[92] M. Sedlmair and M. Aupetit. 2015. Data-driven Evaluation of Visual Quality Measures. *CGF* 34, 3 (2015), 201–210. https://doi.org/10.1111/cgf.12632

[93] Michael Sedlmair, Andrada Tatu, Tamara Munzner, and Melanie Tory. 2012. A Taxonomy of Visual Cluster Separation Factors. *CGF* 31, 3pt4 (2012), 1335–1344. https://doi.org/10.1111/j.1467-8659.2012.03125.x

[94] Thomas Serre, Lior Wolf, and Tomaso Poggio. 2005. Object recognition with features inspired by visual cortex. In *IEEE CVPR*, Vol. 2. 994–1000 vol. 2. https://doi.org/10.1109/CVPR.2005.254

[95] Varshita Sher, Karen G. Bemis, Ilaria Liccardi, and Min Chen. 2017. An Empirical Study on the Reliability of Perceiving Correlation Indices using Scatterplots. *CGF* 36, 61–72. https://doi.org/10.1111/cgf.13168

[96] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. 2019. A comprehensive guide to bayesian convolutional neural network with variational inference. *arXiv preprint* (2019). arXiv:1901.02731

[97] Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. 2016. FigureSeer: Parsing Result-Figures in Research Papers. In *ECCV*. Springer International Publishing, Cham, 664–680.

[98] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. (2015). http://arxiv.org/abs/1409.1556

[99] Tom AB Snijders and Roel J Bosker. 2011. *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. sage.

[100] Stan Development Team. 2020. RStan: the R interface to Stan. http://mc-stan.org/ R package version 2.21.2.

[101] Lim Swee Kiat, Nolan Dey, Mehdi Cherti, and et al. 2021. lucent: Lucid library adapted for pytorch. https://github.com/greentfrapp/lucent

[102] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *ICML*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, 6105–6114. https://proceedings.mlr.press/v97/tan19a.html

[103] Alberto Testolin, Serena Dolfi, Mathijs Rochus, and Marco Zorzi. 2020. Visual sense of number vs. sense of magnitude in humans and machines. *Scientific reports* 10, 1 (2020), 1–13.

[104] Andy Thomas and Christian Kaltschmidt. 2014. Bio-inspired Neural Networks. In *Memristor Networks*. Springer International Publishing, 151–172. https://doi.org/10.1007/978-3-319-02630-5_9

[105] Melanie Tory, David Sprague, Fuqu Wu, Wing Yan So, and Tamara Munzner. 2007. Spatialization Design: Comparing Points and Landscapes. *IEEE TVCG* 13, 6 (2007), 1262–1269. https://doi.org/10.1109/TVCG.2007.70596

[106] Jiachen Wang, Xiwen Cai, Jiajie Su, Yu Liao, and Yingcai Wu. 2021. What makes a scatterplot hard to comprehend: data size and pattern salience matter. *Journal of Visualization* (2021), 1–17. https://doi.org/10.1007/s12650-021-00778-8

[107] Pei Wang, Yijun Li, and Nuno Vasconcelos. 2021. Rethinking and improving the robustness of image style transfer. In *IEEE/CVF CVPR*. 124–133. https://arxiv.org/abs/2104.05623

[108] Qianwen Wang, Zhutian Chen, Yong Wang, and Huamin Qu. 2021. A Survey on ML4VIS: Applying MachineLearning Advances to Data Visualization. *IEEE TVCG* (2021). https://doi.org/10.1109/TVCG.2021.3106142

[109] Yunhai Wang, Xin Chen, Tong Ge, Chen Bao, Michael Sedlmair, Chi-Wing Fu, Oliver Deussen, and Baoquan Chen. 2019. Optimizing Color Assignment for Perception of Class Separability in Multiclass Scatterplots. *IEEE TVCG* 25, 1 (2019), 820–829. https://doi.org/10.1109/TVCG.2018.2864912

[110] Zijie J. Wang, Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, and Duen Horng Polo Chau. 2021. CNN explainer: Learning convolutional neural networks with interactive visualization. *IEEE TVCG* 27, 2 (2021), 1396–1406. https://doi.org/10.1109/TVCG.2020.3030418

[111] P.D. Wasserman and T. Schwartz. 1988. Neural networks. II. What are they and why is everybody so interested in them now? *IEEE Expert* 3, 1 (1988), 10–15. https://doi.org/10.1109/64.2091

[112] G. N. Wilkinson and C. E. Rogers. 1973. Symbolic description of factorial models for analysis of variance. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 22, 3 (1973), 392–399. https://doi.org/10.2307/2346786

[113] Leland Wilkinson, Anushka Anand, and Robert Grossman. 2005. Graph-Theoretic Scagnostics. In *IEEE Information Visualization*. 21. https://doi.org/10.1109/INFOVIS.2005.14

[114] Leslie Wöhler, Yuxin Zou, Moritz Mühlhausen, Georgia Albuquerque, and Marcus Magnor. 2019. Learning a Perceptual Quality Metric for Correlation in Scatterplots. In *Vision, Modeling and Visualization*. https://doi.org/10.2312/vmv.20191318

[115] Cindy Xiong, Ali Sarvghad, Çağatay Demiralp, Jake M Hofman, and Daniel G Goldstein. 2022. Investigating Perceptual Biases in Icon Arrays. In *ACM CHI*.

[116] Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. 2014. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences* 111, 23 (2014), 8619–8624. https://doi.org/10.1073/pnas.1403112111 arXiv:https://www.pnas.org/content/111/23/8619.full.pdf

[117] Fumeng Yang, Lane Harrison, Ronald A. Rensink, Steven Franconeri, and Remco Chang. 2019. Correlation Judgment and Visualization Features: A Comparative Study. *IEEE TVCG* 25, 3 (2019), 1474–1488. https://doi.org/10.1109/TVCG.2018.2810918

[118] Linping Yuan, Ziqi Zhou, Jian Zhao, Yiqiu Guo, Fan Du, and Huamin Qu. 2021. InfoColorizer: Interactive Recommendation of Color Palettes for Infographics. *IEEE TVCG* (2021), 1–1. https://doi.org/10.1109/TVCG.2021.3085327

[119] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide Residual Networks. In *Proceedings of the British Machine Vision Conference*, Edwin R. Hancock Richard C. Wilson and William A. P. Smith (Eds.). Article 87, 12 pages. https://doi.org/10.5244/C.30.87

[120] Richard Zhang. 2019. Making Convolutional Networks Shift-Invariant Again. *CoRR* abs/1904.11486. arXiv:1904.11486 http://arxiv.org/abs/1904.11486

[121] Jian Zhao, Mingming Fan, and Mi Feng. 2020. ChartSeer: Interactive Steering Exploratory Visual Analysis with Machine Intelligence. *IEEE TVCG* (2020), 1–1. https://doi.org/10.1109/TVCG.2020.3018724