



NORTHWESTERN UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2026-33

June, 2026

Computational Models of Visualization Perception

Sheng Long

Abstract

When viewers read a data visualization, they perform perceptual computations, translating visual marks into quantitative judgments, that are systematically imperfect. Decades of graphical perception research have documented these errors, but most findings take the form of ordinal rankings or categorical taxonomies. While helpful, these descriptions do not make quantitative predictions: they cannot predict how large a viewer's error will be for a specific chart design, dataset, and task, nor can they generalize to combinations not yet tested experimentally.

This dissertation develops computational models that make quantitative predictions about how viewers interpret data visualizations. Three studies model different aspects of visualization perception with increasing generality. First, I develop a formal model of y-axis truncation in bar charts that defines task-dependent conditions under which truncation preserves or distorts data structure, replacing heuristical prescriptions with formally grounded, computable design guidance. Second, I test whether deep neural network features trained on natural images can serve as computational proxies for human similarity judgments of visualizations. Finally, I propose visual decoding operators --- composable perceptual primitives, each with estimable bias and variance --- and provide an existence proof that operators characterized on PDF and CDF charts compose to predict scatterplot mean-estimation performance with no parameters fit to the target data.

Together, these studies demonstrate that computational models of visualization perception are both feasible and productive: they predict quantities that ordinal rankings cannot, expose mechanisms that holistic accuracy measures obscure, and generalize across chart types and tasks.

Keywords

Information Visualization, Visual Decoding Operators, Computational Modeling

NORTHWESTERN UNIVERSITY

Computational Models of Visualization Perception

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Computer Science

By

Sheng Long

EVANSTON, ILLINOIS

September 2026

© Copyright by Sheng Long, 2026

All Rights Reserved

Abstract

When viewers read a data visualization, they perform perceptual computations, translating visual marks into quantitative judgments, that are systematically imperfect. Decades of graphical perception research have documented these errors, but most findings take the form of ordinal rankings or categorical taxonomies. While helpful, these descriptions do not make quantitative predictions: they cannot predict how large a viewer’s error will be for a specific chart design, dataset, and task, nor can they generalize to combinations not yet tested experimentally.

This dissertation develops computational models that make quantitative predictions about how viewers interpret data visualizations. Three studies model different aspects of visualization perception with increasing generality. First, I develop a formal model of y-axis truncation in bar charts that defines task-dependent conditions under which truncation preserves or distorts data structure, replacing heuristical prescriptions with formally grounded, computable design guidance. Second, I test whether deep neural network features trained on natural images can serve as computational proxies for human similarity judgments of visualizations. Finally, I propose visual decoding operators — composable perceptual primitives, each with estimable bias and variance — and provide an existence proof that operators characterized on PDF and CDF charts compose to predict scatterplot mean-estimation performance with no parameters fit to the target data.

Together, these studies demonstrate that computational models of visualization perception are both feasible and productive: they predict quantities that ordinal rankings cannot, expose mechanisms that holistic accuracy measures obscure, and generalize across chart types and tasks.

Acknowledgment

First and foremost, I want to thank my parents, Bin Long and Yanhua Xia. Without their continued love and support, I would not have been able to finish a doctoral program. There were times when they tried to talk me out of doing a Ph.D., which in actuality served as more fuel for my determination to finish what I started. My brother also played a role in encouraging me to keep exploring, and for that, I begrudgingly bestow him a spot in my acknowledgement. I also want to thank my grandparents and extended family for the love, guidance, and support that I received along the way.

No good Ph.D. program is complete without being advised and mentored by amazingly kind and smart people. I thank Jason Hartline for admitting me into the theory track at Northwestern University and for encouraging me to pursue research directions where I have more interest and competitive advantage in. I thank Matthew Kay for being the best advisor that one could possibly ask for. Matt inspired me to look at problems from an angle of “constructive dissatisfaction”. Thank you, Matt, for encouraging me to go on tangents and for believing in me and my way of asking and solving problems. I mostly likely won’t be able to finish this doctoral program without Matt’s continuous supply of thoughtful feedback and words of encouragement (interspersed with “not bad”s here and there). Finally I thank Evgenia Smirni from William & Mary, who gave me the advice to try out a PhD program (“if you don’t like it, you can always master out. It will be free education!”), and wrote stellar recommendation letters that helped me get into the doctoral program at Northwestern University.

A wise collaborator once told me that being an academic is like working a job where everyday is Saturday but you have to work on Saturdays. I think about this quote a lot. Without the friends I made along the way, everyday will be more like a working Monday. I thank my friends, Nora Pollard, Yifan Wu, Vaidehi Srinivas, and Jifan Zhang for the inspiring conversations, countless coffee breaks, and long walks filled with philosophical thoughts and gossip. I thank my roommate who became friends, Yiming Liu, Ru Ruby Yang, and He Jia, for the delicious meals we made together and for making it less painful to deal with household chores while managing stressful paper deadlines. I thank my lab mates from MU Collective for providing constructive criticism and fostering a friendly, open environment that made me a better

researcher: Mandi Cai, Charles Cui, Lily Ge, Ziyang Guo, Maryam Hedayati, Paula Kayongo, Hyeok Kim, Taewook Kim, Priyanka Nanayakkara, Fumeng Yang, and Dongping Zhang.

I thank my talented collaborators, Hedyeh Beyhaghi, Modibo K Camara, Aleck Johnsen, Chenhao Zhang, Michael Correll, Angelos Chatzimpampas, Emma Alexander, Jessica Hullman, Atsuya Kobayashi, Kei Tateno, Remco Chang, Eugene Wu, Alex Kale, Arran Zeyu Wang, and Danielle N. Szafr. I thank Alex Lex, Lane Harrison, and Andrew McNutt for inviting and hosting me for a retreat at the University of Utah and for the advice they gave me about career choices. I thank my thesis committee members — Matt, Emma, Eugene, Alex, and Michael — for their thoughtful feedback, guidance, and continued support.

Starting a PhD in 2020 during the middle of COVID was not on my bingo card, and I want to thank my great-aunt Melody Shaw and her husband Steve Lin for welcoming me into their home and for hosting me when I was transitioning from a fresh-out-of-college student into a fresh-into-phd student. Thank you for including me in your Mahjong sessions, your walks around the neighborhood, and for teaching me how to make scallion pancakes. I am forever grateful for this tremendous act of love and support during those trying times. I also want to thank John and Margaret Pollard, for hosting me multiple times and for sending me a Thanksgiving gift basket in 2020. It helped a hungry PhD student who forgot to order groceries online and was facing an empty fridge and contemplating surviving the day with only water.

Let me end by saying that I never pictured myself doing a Ph.D., yet here I am. It won't be possible without the kind gestures and support I received along the way. Thank you.

TABLE OF CONTENTS

1	Introduction	18
1.1	Organizing the Space: Data, Visual Form, and Task	21
1.2	Thesis Statement	22
1.3	Thesis Overview	22
2	General Background and Related Work	25
2.1	Graphical Perception & Comprehension	25
2.2	Structure Preservation and Representational Equivalence	26
2.3	Empirical Evaluation and Modeling	28
2.3.1	Evaluating Visualizations	28
2.3.2	Bayesian Hierarchical Modeling	28
3	A Systematic Exploration of Y-Axis Truncation	30
3.1	Introduction	30
3.2	Related Work	31
3.2.1	Graphical Perception of Bar Charts	31
3.2.2	Deceptive Visualization & Y-Axis Truncation	32
3.3	A Model for Y-Axis Truncation	35
3.3.1	Data Relations	38
3.3.2	Monotonicity of Truncation	39
3.4	Methodology	39
3.4.1	Judgment Tasks	40

3.4.2	Experiment stimuli and data generation	7
3.4.2.1	Data-generating distribution	41
3.4.2.2	Data relation	42
3.4.2.3	Type of truncation	42
3.4.3	Experiment procedure	43
3.4.4	Participants and Exclusion Criteria	44
3.4.5	Analysis Approach	44
3.5	Results	45
3.5.1	Data Preliminaries	45
3.5.2	Gap	46
3.5.2.1	Compare Gap	46
3.5.2.2	Estimate Gap	47
3.5.3	Ratio	48
3.5.3.1	Compare Ratio	48
3.5.3.2	Estimate Ratio	49
3.5.4	Summary	50
3.6	Discussion	51
3.6.1	Recommendations for Y-Axis Truncation	51
3.6.1.1	$D \leftrightarrow V$ correspondence	51
3.6.1.2	$V \leftrightarrow T$ correspondence	52
3.6.1.3	D or T	53
3.6.1.4	Our Approach and Recommendation	53
3.6.2	Limitations	55
4	Applying Deep-feature-based Similarity Metrics to Information Visualization	56
4.1	Introduction	56
4.2	Background	57
4.2.1	Perceptual Similarity Metrics	57
4.2.1.1	Collecting Human Similarity Judgments	59

4.2.2 Comparing Human and Machine Behavior	8
4.2.2.1 Functional Similarity	60
4.2.2.2 Representational Similarity	62
4.2.2.3 Texture vs Shape	62
4.3 Research Goals and Experiment Overview	63
4.4 Implementing Deep-Feature-Based Perceptual Similarity Metrics	64
4.4.1 Pre-processing	67
4.5 Replicating Veras and Collins [179]	67
4.5.1 Overview of Original Paper	67
4.5.1.1 Data, Visual Stimuli, Task	68
4.5.1.2 Similarity Measure	69
4.5.1.3 Performance Evaluation	70
4.5.2 Implementation	70
4.5.3 Our Results	71
4.5.4 Ablation Study	72
4.5.4.1 Robustness Check — Random Weights	73
4.5.4.2 Impact of Each Feature Extraction Layer	73
4.5.4.3 Impact of Training Objective and Training Dataset	75
4.5.5 Discussion of Veras and Collins	76
4.6 Replicating Demiralp et al. [34]	77
4.6.1 Overview of Original Paper	77
4.6.1.1 Data and Visual Stimuli	77
4.6.1.2 Tasks	78
4.6.2 Implementation	78
4.6.2.1 Data and Visual Stimuli	79
4.6.2.2 Performance Evaluation	79
4.6.3 Results	79
4.6.3.1 Color	80

4.6.3.2	Size	9
4.6.3.2	Size	81
4.6.3.3	Shape	81
4.6.3.4	Size-color	82
4.6.4	Robustness Check – Random Weights	83
4.6.4.1	Color	84
4.6.4.2	Size	84
4.6.4.3	Shape	85
4.6.4.4	Size-color	86
4.6.5	Discussion of Demiralp et al.	87
4.7	Discussion	89
4.7.1	Key Findings	89
4.7.2	Limitation and Future Work	90
4.7.3	Similarity, Constraints, and Inferred Representation	91
4.7.4	Potential Applications and Broader Implications	92
5	Towards A Compositional Theory of Visualization Perception	93
5.1	Introduction	93
5.2	Background and Related Work	96
5.2.1	Theories and Models of Visualization Comprehension	96
5.2.2	Error Attribution in Visualization Experiments	98
5.3	A Model For Visual Operators	99
5.4	Building Operators	102
5.4.1	Experiment Design	103
5.4.1.1	Tasks, Instructions, and Training	103
5.4.1.2	Stimuli and Apparatus	104
5.4.1.3	Experiment Procedure and Participants	105
5.4.1.4	Analysis Approach	106
5.4.2	Results	106
5.4.2.1	Projection Operators	106

5.4.2.2	Highest Point Operator	108
5.4.2.3	Max Slope Operator	109
5.4.2.4	Bisect Area Operator	110
5.5	Applying Learned Operators	113
5.5.1	Moritz et al. Experiment 2	114
5.5.2	Our Experiment Design	114
5.5.2.1	Tasks, Instructions, and Training	114
5.5.2.2	Stimuli and Apparatus	114
5.5.2.3	Experiment Procedure and Participants	115
5.5.3	Analysis Approach	116
5.5.3.1	Hypothesized Strategies	116
5.5.4	Results	117
5.5.4.1	Part 1: ProjToAxis_y Operator Generalizes Across Participant Groups	117
5.5.4.2	Part 2: Comparing Composition Strategies	117
5.5.4.3	Implications	118
5.6	Discussion	119
5.6.1	A Compositional Interface to Formalize Graphical Perception Knowledge	119
5.6.2	Toward Algorithmic-Level Theories of Visualization Interpretation	121
5.6.3	Perceptual Proxies and the Problem of Identification	123
5.6.4	Limitations and Future Work	124
6	General Discussions	126
6.1	How Should We Do Empirical Visualization Research?	126
6.2	What's Next?	129
6.2.1	Toward a Library of Operators	129
6.2.2	Attention, Salience, and the Tools to Capture Them	130
6.2.3	Is Reading A Graph Similar to Reading a Paragraph?	131
6.3	Limitations	131
6.3.1	Scope of chart types	131

6.3.2 Ecological validity	11
6.3.3 Tasks and Subjectivity Embedded Within	132
7 Conclusion	133
References	134
Appendix	154
A.1 Chapter Three Related	154
A.1.1 brms formulae	154
A.1.2 Math calculations	154
A.2 Chapter Four Related	155
A.2.1 Neural Network Architectures and Extraction Points	155
A.2.2 Neural Network Weights	157
A.2.3 Pre-processing with Resolution 224×224	157
A.2.4 Clustering of Scatterplots	158
A.3 Chapter Six Related	159
A.3.1 Full Hierarchical Model Specifications	159
A.3.2 Screenshot of Experiment Interface	160
A.3.3 Skewed Generalized T Distribution	160
A.3.4 Additional Posterior Check Related Plots	161
A.3.4.1 Checks for Experiment 1	161
A.3.4.1.1 Remaining Two Projection Operators	162
A.3.4.1.2 Highest Point Y Operator	162
A.3.4.1.3 Gaussian model for HighestPoint_x	162
A.3.4.1.4 Max Slope Operator	163
A.3.4.1.5 Bisect Area Operator	164
A.3.5 Checks for Experiment 2	164
A.3.5.1 Part 1 – Project to Axis Y	164
A.3.5.2 Part 2 – Project-Once	164
A.3.5.3 Part 2 – Project-Twice	165

LIST OF FIGURES AND TABLES

Figure 1	Screenshot of Nvidia stock price, taken 06/04/26.	18
Figure 2	Obtained from this blog post	18
Figure 3	Obtained from scikit-learn's documentation	18
Figure 4	UK Public Sector Capital Expenditure, 2014/15, obtained from this website . For illustration purposes I removed the numbers for “10.1 Sickness and disability” and “10.2 Old age”. . . .	19
Figure 5	Heatmap of monthly visitor counts to the Yosemite National Park from 1979 to 2023. Created using SveltePlot https://svelteplot.dev/ with data obtained from https://www.responsible-datasets-in-context.com/posts/np-data	21
Figure 6	Screenshot of the framework from Kindlemann and Scheidegger [96].	26
Figure 7	Screenshot of Figure 1 from Demiralp et al. [35].	27
Figure 8	Figure 4 from Cleveland and McGill [23], demonstrating the sample stimuli used in the position-length experiment.	31
Figure 9	What are monotonic truncations and what are non-monotonic truncations?	34
Figure 10	Left: example of untruncated bar chart. Right: example of truncated bar chart. The two charts share the same data set $\{A_1, A_2, B_1, B_2\}$	36
Figure 11	An example of the four kinds of data relations.	38
Figure 12	Sample question from the experiment. The participant is asked to estimate the ratio between the profit in 2020 and the profit in 2021 for April. Note that the y-axis is truncated to start at 55.	41
Figure 13	Overview of experiment procedure. Participants were first randomly assigned to one of four tasks, then to the same set of training stimuli in the same order, and then to the same set of 20 conditions but different testing stimuli. Observe that $2 \times 4 \times 3 \times 2 = 48 \neq 40$ trials — this is because it is impossible to truncate non-monotonically for the two discordant data relations, thus resulting in $2 \times 4 \times 3 \times 2 - 2 \times 2 \times 1 \times 2 = 48 - 8 = 40$ trials.	43

	13
Figure 14	<i>Left:</i> Posterior density, median, 66% and 95% quantile interval of the probability of correctly answering. <i>Right:</i> Expected difference in the means between different values of a variable. Pink ticks represent the percentage of correct response answers. 46
Figure 15	<i>Left:</i> Posterior density, median, 65% and 95% quantile interval of absolute error. <i>Right:</i> Expected difference in the means between different values of a variable. 47
Figure 16	<i>Left:</i> Posterior density, median, 66% and 95% quantile interval of the probability of correctly answering. Pink ticks represent the percentage of correct response answers. <i>Right:</i> Expected difference in the means between different values of a variable. Similar to our results for Compare Gap in Figure 14, the median of posterior predictions (represented by black dots) are closer together than the raw data (represented by pink ticks), which demonstrates <i>model shrinkage</i> . This will make estimates of differences more conservative and is thus suitable for the sample size we’ve collected. 48
Figure 17	<i>Left:</i> Posterior density, median, 66% and 95% quantile interval of the absolute error for task Estimate Ratio. <i>Right:</i> Expected difference in the means between different values of a variable. 49
Figure 18	A general framework for comparing two information-processing systems (e.g., humans vs. machines), inspired by the framework proposed by Sucholutsky et al. [165]. Framework uses Demiralp et al.’s experiment on the perception of shape similarity as one concrete example [34]. 60
Figure 19	Diagram showing how the deep-feature-based perceptual similarity metric calculates “perceptual distance”, using VGG16 as the DL network $\mathcal{F}$ 64
Table 1	Network Architectures and Weights 66
Figure 20	Screenshot obtained from the supplementary materials of Pandey et al. [129] showing Participant 1’s final screen after completing the naming phase, with group descriptions and their corresponding easiness and confidence scores (separated by pipe characters). 68
Figure 21	Black diamonds represent Veras and Collins’ [179] results using gradient-descent-tuned MS-SSIM. Crosses represent results using Mean Squared Error (MSE). Points are offset for clarity. 72
Figure 22	Black diamonds represent Veras and Collins’ [179] results using gradient-descent-tuned MS-SSIM. Crosses represent results using MSE. Dots show means of 10 trials. Lines indicate 95% confidence intervals (CIs) from non-parametric bootstrap. Points and CIs are slightly offset for clarity. 73
Figure 23	Dashed lines (- -) represent Veras and Collins’ results using gradient-descent-tuned MS-SSIM. (a) Clustering quality measures for ● AlexNet, ● AlexNet (LPIPS), ● EfficientNet

	B0, ● SqueezeNet, and ● SqueezeNet (LPIPS) calculated using perceptual distance at each layer $l \in \mathcal{L}$. (b) Clustering quality measures for ● AlexNet, ● AlexNet (LPIPS), ● EfficientNet B0, ● SqueezeNet, and ● SqueezeNet (LPIPS) calculated using all layers except layer $l \in \mathcal{L}$	74
Figure 24	● Resnet18 and ● ResNet50 performance. Black diamonds represent Veras and Collins' [179] results using gradient-descent-tuned MS-SSIM. Crosses represent results using MSE. Points are dodged for clarity.	75
Figure 25	Palettes that maximize perceptual discriminability for color, shape, and size generated by Demiralp et al. [34], using the perceptual kernels from the triplet matching task.	77
Figure 26	(a) Spearman's rank correlation between distance matrices and Demiralp et al.'s [34] perceptual kernels (dashed: MSE-based distance matrices), (b) AlexNet (LPIPS) distance matrix, (c) Demiralp et al.'s Td perceptual kernel.	80
Figure 27	(a) Spearman's rank correlation between distance matrices and Demiralp et al.'s [34] perceptual kernels (dashed: MSE-based distance matrices), (b) AlexNet (LPIPS) distance matrix, (c) Demiralp et al.'s Sa perceptual kernel.	81
Figure 28	(a) Spearman's rank correlation between distance matrices and Demiralp et al.'s [34] perceptual kernels (dashed: MSE-based distance matrices), (b) VGG16 (LPIPS) distance matrix, (c) Demiralp et al.'s Tm perceptual kernel.	82
Figure 29	(a) Spearman's rank correlation between distance matrices and Demiralp et al.'s [34] perceptual kernels (dashed: MSE-based distance matrices), (b) Demiralp et al.'s Tm perceptual kernel, (c) AlexNet distance matrix, (d) Demiralp et al.'s Sa perceptual kernel.	83
Figure 30	(a) Spearman's rank correlations between random-weight architectures' distance matrices and perceptual kernels (95% bootstrap CI; dashed: MSE-based distance matrices), (b) Random AlexNet's average distance matrix, (c) Demiralp et al.'s Tm perceptual kernel. . .	84
Figure 31	(a) Spearman's rank correlations between random-weight architectures' distance matrices and perceptual kernels (95% bootstrap CI; dashed: MSE-based distance matrices), (b) Random AlexNet's average distance matrix, (c) Demiralp et al.'s Sa perceptual kernel. . .	85
Figure 32	(a) Spearman's rank correlations between random-weight architectures' distance matrices and perceptual kernels (95% bootstrap CI; dashed: MSE-based distance matrices), (b) Random AlexNet's average distance matrix, (c) Demiralp et al.'s Td perceptual kernel. . .	86
Figure 33	(a) Spearman's rank correlations between random-weight architectures' distance matrices and perceptual kernels (95% bootstrap CI; dashed: MSE-based distance matrices), (b) Random AlexNet's average distance matrix, (c) Demiralp et al.'s Tm perceptual kernel. . .	87

	15
Figure 34	An illustrative example of different ways tasks can be set up to elicit a response. We use elicitation method 3 (details in Section 5.4.1). 99
Figure 35	Diagram explaining the task decomposition process using median identification on CDF as a concrete example. 100
Figure 36	A horseshoe diagram [177] of this paper’s approach to modeling. 102
Table 2	Tasks and their representative instructions used during experiment. 103
Figure 37	 104
Figure 38	Diagram of experiment procedure. The study used a within-subjects design. 105
Figure 39	Plots for (a) ProjToCurve , (b) ProjToAxis_x , and (c) ProjToAxis_y : error vs. distance between start and end points of projection, where points show observed errors with blue bands representing 50%, 80%, and 95% posterior predictive intervals, marginalizing over participants. 107
Figure 40	Posterior predictive checks for ProjToCurve . (a) KDE density overlay plot with 100 posterior draws. (b) PIT-ECDF Calibration plot. 108
Figure 41	 108
Figure 42	Posterior predictive error distributions for HighestPoint_y applied to an example stimulus. The top-left panel shows the stimulus curve with the true peak marked. The bottom-left panel shows the marginal predictive distribution of error along the x-axis, and the top-right panel shows the corresponding distribution along the y-axis (HighestPoint_y). . 109
Figure 43	 109
Figure 44	(a) Posterior retrodictive density of participants’ errors overlaid with density of observed errors, plotted on top of strip plot of observed errors for MaxSlope . (b) PIT-ECDF Calibration plot. 110
Figure 45	 110
Figure 46	Posterior density, 50%, 85% and 95% quantile interval of signed error of BAHP overlaid on observed signed error from participants. 112
Figure 47	Overview of the cross-task generalization approach. Left: the ProjToAxis_y operator is measured by asking participants to project a dot to the y-axis on a 600×450 px chart. Right: the same operator, without refitting, is composed under candidate averaging strategies to predict mean-estimation responses on Moritz et al.’s [120] 500 × 200 px scatterplot task. All composition and prediction occurs in the perceptually-uniform visual angle space (center). 113

Figure 48	Error vs. distance between start and end points of projection for ProjToAxis_y , measured in radian units, where points show observed errors with blue bands representing 50%, 80%, and 95% posterior predictive intervals, marginalizing over participants.	117
Figure 49	Cross-task posterior predictive density overlay plot, each with 100 draws, faceted by variability upper/lower vs. variability 0/0.4 experiment conditions. Point intervals indicate the mean and 68% quantile interval (approximately ± 1 SD) for observed responses and model predictions respectively.	118
Table 3	Network architecture, feature extraction points, ImageNet top-1 accuracy, and model size	156
Figure 50	Black diamonds represent Veras and Collins' [179] results using gradient-descent-tuned MS-SSIM. Points are slightly offset for clarity. Results when replicating Pandey et al's experiment using resolution $224 \times 224 \times 3$	158
Figure 51	Consensus clustering of scatterplots from Pandey et al. [129]	158
Figure 52	Clustering labels of scatterplots from Pandey et al. [129] using SqueezeNet (LPIPS) as the backbone network for the deep-feature-based similarity metric.	158
Table 4	Clustering quality measures obtained from 64×64 , highlighting the top three performing backbones for each measure.	159
Figure 53	An example of the experiment interface. The generalized skewed- t distribution is parameterized with $\mu = -1.1, \sigma = 0.7, \lambda = -0.1, p = 4.6, q = 31.7$	160
Figure 54	Plots for ProjToAxis_x and ProjToAxis_y : (a), (c) Posterior predictive checks with 100 draws for x -Axis and y -Axis respectively. (b), (d) PIT-ECDF calibration plot for x -Axis and y -Axis respectively.	162
Figure 55	Plots for HighestPoint_y : (a) Posterior predictive check with 100 draws, (b) PIT-ECDF calibration.	162
Figure 56	Plots for HighestPoint_x task for error on the x -Axis, except <i>here we are using a Normal distribution instead of a Laplace distribution</i> to model: (a) PIT-ECDF calibration chart, (b) Posterior predictive check with 100 draws	163
Figure 57	Plots for MaxSlope : (a) Posterior predictive check with 100 draws, (b) PIT-ECDF calibration plot.	163
Figure 58	Plots for BisectArea . (a) Posterior predictive checks with 100 draws, where the plot is faceted to have three parts, where the label indicates whether the sign of λ , the skewness parameter, is negative, positive, or zero, and (b) PIT-ECDF calibration chart, where the plot is faceted to have three parts, where the label indicates whether the sign of λ , the skewness parameter, is negative, positive, or zero. (a) and (b) demonstrate good fits	164

Figure 59	Plots for ProjToAxis_y for Experiment 2. (a) Posterior predictive checks with 100 draws for the y -Axis. (b) PIT-ECDF calibration plot for y -Axis.	164
Figure 60	Predictions composed under project-once strategy for (a) mean, (b) median, and (c) weighted.	165
Figure 61	Predictions composed under project-twice strategy for (a) mean, (b) median, and (c) weighted.	166

1 INTRODUCTION

In science, if you know what you are doing, you should not be doing it. In engineering, if you do not know what you are doing, you should not be doing it. Of course, you seldom, if ever, see either pure state.

— Richard Hamming, *The Art of Doing Science and Engineering*

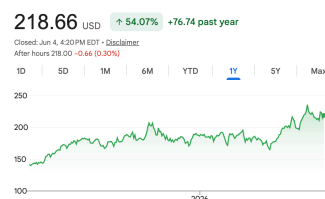


Figure 1: Screenshot of Nvidia stock price, taken 06/04/26.

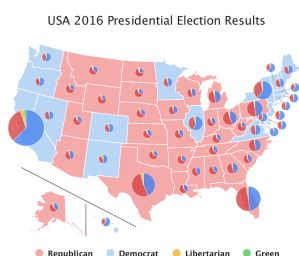


Figure 2: Obtained from [this blog post](#).

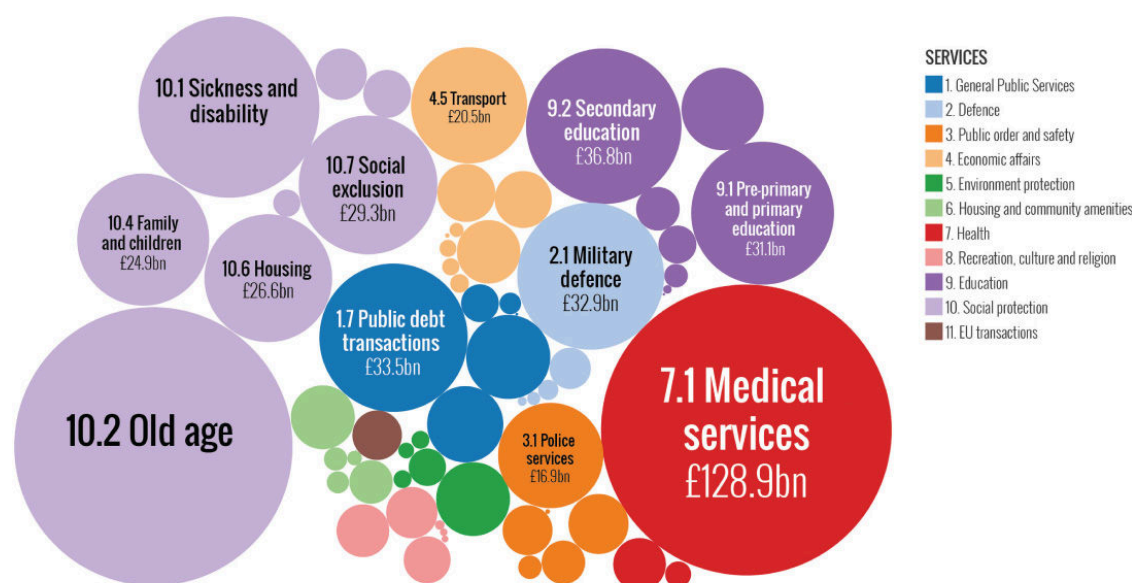


Figure 3: Obtained from [scikit-learn's documentation](#).

Visualizations are everywhere. We track a stock by its price line, [fig. 1](#) read real-time election results from maps combined with pie charts, [fig. 2](#) and judge machine learning classifier performance from hyperplanes drawn on scatterplots. [fig. 3](#) In none of these cases is the figure mere decoration: it is the interface through which the data reaches us.

Visualizations are, in this sense, the primary medium between human cognition and data. When a viewer reads a bar chart, estimates a trend from a line graph, or judges the median of a distribution plot, they are performing a “perceptual computation”: translating visual marks and spatial relationships into quantitative judgments. One interesting observation about this perceptual computation is that the same dataset, plotted differently, often yields different quantitative judgments, some more accurate than others. Consider the following bubble chart that displays the public sector capital expenditure (£ billion) on services by function of the UK Government during 2014/15 (Fig. 4). What share

of the old-age spending does the sickness-and-disability sharing represent? More than half, or less than half?



Analysis of Public sector capital expenditure (£ million) on services by function of the UK Government during 2014/15
Data from: HM Treasury, 'Public Expenditure Statistical Analyses 2015' <https://www.gov.uk/government/statistics/public-expenditure-statistical-analyses-2015>

It might surprise people to find out that the answer is less than half.¹ This is a predictable consequence of how humans decode information via *area*: since area is decoded sublinearly, the larger bubble is perceptually discounted. If we were to map the same dataset to a bar chart, or a line chart, the same comparison would have been more accurate.

Cleveland and McGill [23] define this “visual decoding of information encoded on graphs” as *graphical perception* in 1984. They ran psychophysical experiments to rank a set of *elementary perceptual codes* ² by the accuracy with which viewers decode quantities via those visual channels, yielding a design principle: encode data using channels higher on the accuracy ordering.³ In the decades since, hundreds of studies have followed this style of empirical research, documenting errors in the visual decoding process across chart types, tasks, and populations [144].

This empirical tradition has produced a rich body of knowledge,⁴ and the majority of such knowledge takes the following form: *ordinal* rank-

Figure 4: UK Public Sector Capital Expenditure, 2014/15, obtained from [this website](#). For illustration purposes I removed the numbers for “10.1 Sickness and disability” and “10.2 Old age”.

¹ The spending for sickness and disability is 50.0 bn while the spending for old age is 118.2 bn.

² i.e., position along a common scale, length, angle, area, and so on

³ As statisticians, Cleveland and McGill saw graphical perception as fundamental to the *construction of statistical graphics*. [24]

⁴ ... including but not limited to numerous guidelines for visualization design, such as [2,122]

ings and *categorical* taxonomies. We know that position encodings are more accurate than length encodings, which are more accurate than angle encodings [23]. We know that common visual analytic tasks can be classified at a low level as “retrieve value”, “find extremum”, “determine range”, and so on [2]. While rankings and taxonomies organize knowledge and shape design guidelines, recommendation systems, and teaching practices [121], they share a limitation: they do not make quantitative predictions that *generalize beyond the conditions in which they were measured*.⁵ The consequence is that every new chart type, data distribution, or task combination requires a new experiment — an approach that, as the design space of possible visualizations grows,⁶ does not scale.

⁵ In other words, knowing that position outperforms angle does not tell us *by how much*, or *under what data conditions*, or *for which specific tasks*.

⁶ For example, consider Fuchs et al. [48], who looked at experiment studies on data glyphs and produced a 9×9 design table to classify which glyph studies fall in which categories. Introducing a new dimension will scale this table by at least 81 more cells.

⁷ For example, Gillan and Lewis’s MA-P model [56] proposes that response time scales linearly with the number of cognitive components a task requires.

The idea of building models that generalize for visualization perception is not new. Process models of graph comprehension decompose chart-reading into sequences of elementary operations and predict task performance as a function of those operations.⁷ An alternative “model” is the Draco system [121], which formalizes design knowledge as weighted logical constraints and optimizes over them to recommend effective chart designs. The constraint weights themselves can be learned from graphical perception experiments [204], making Draco a computational model of sorts whose inputs are empirical perceptual experiment outcomes and outputs are ranked recommendations of visualization design.

While these models demonstrate the value of computational approaches to visualization, they predict different quantities than the ones this dissertation targets. Process models like MA-P [56] and ACT-R [134] predict the time course of graph reading. Recommendation systems like Draco optimize over ordinal effectiveness rankings. Neither predicts the

perceptual output — the estimate a viewer produces and the bias and variance of that estimate for a given chart and task. This is the gap the present work addresses. **My goal is to build models that generalize beyond the conditions used to fit them and predict the quantitative error characteristics of viewers’ perceptual judgments**, such that performance on new visualization $\times$ task combinations can be anticipated *without* running new experiments.

1.1 ORGANIZING THE SPACE: DATA, VISUAL FORM, AND TASK

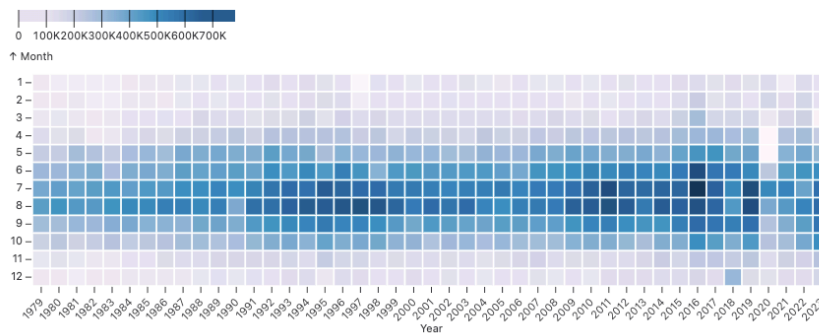


Figure 5: Heatmap of monthly visitor counts to the Yosemite National Park from 1979 to 2023. Created using SveltePlot <https://svelteplot.dev/> with data obtained from <https://www.responsible-datasets-in-context.com/posts/np-data>.

To illustrate the factors that go into a chart, consider the above heatmap (Fig. 5). The chart begins with *data*, number of visitors to the Yosemite National park, by month and year, from 1979 to 2023. A designer chooses how to map this data to *visual forms* — month to y-axis coordinate position, year to x-axis coordinate position, number of visitor per month to color on the Blues scale.⁸ Coordinate system, position, and color, together these visual channels form a *visualization*. A viewer reading the above heatmap can then arrive at some perceptual judgment.⁹

From this example, we see that the perceptual judgments a viewer makes from a visualization depend on the interaction of at least three components: the characteristics of the underlying *Data* (*D*), the chosen *Visual form* (*V*) used to represent the data, and the specific *Task* (*T*) the

⁸ Not to be confused with the musical blues scale, this is the ColorBrewer sequential color scale. See https://scales.r-lib.org/reference/pal_brewer.html.

⁹ For example, across the years, more people tend to visit Yosemite during the summer, or 2020 had almost no visitors during April and May.

¹⁰ Admittedly, viewers rarely see a visualization in the wild with a pre-supposed task. A single visualization usually can support different tasks with varying degrees. More on this in Chapter 6.

viewer is performing.¹⁰ This observation appears in various forms across the literature. For instance, in Hu et al. [76], who proposed viewing each realized visualization as a (D, V, T) tuple; in Kindlmann and Scheidegger [96], whose algebraic framework examines how changes in data map to changes in visual form; and in Demiralp et al. [35], who model visualizations as functions from data points to visual primitives.

Later, I will use the $D \leftrightarrow V \leftrightarrow T$ notation to serve as a framing device for illustrating what each chapter of this dissertation investigates.

1.2 THESIS STATEMENT

This dissertation argues that computational models can make quantitative predictions about how viewers read data visualizations that *generalize beyond the conditions in which they were measured*, and that building such models is both feasible and productive. I make this case through three studies that form a progression: each is prompted by a limitation the previous one exposes, moving from a model of a single perceptual effect toward a general account of how visualization reading composes.

1.3 THESIS OVERVIEW

The three studies in this dissertation form a progression but differ in what aspect of visualization perception they target, what level of generality they aim for, and what formal tools they bring to bear.

¹¹ Chapter 3 manipulates V (truncating the y-axis of a grouped bar chart) for systematically varied D (data values) and measures the effect on T (comparison tasks).

Chapter 3¹¹ develops a formal model of y-axis truncation in bar charts. Prior work established that truncation exaggerates the perceived magnitude of differences between bars [27]. This chapter formalizes truncation as a transformation of the visual encoding function and defines when that transformation is *structure-preserving*.¹² Experiments confirm that monotonic truncation does not impair performance while

¹² A truncation is *monotonic* if it preserves the data relation relevant to the viewer's task, and *non-monotonic* if it reverses it

non-monotonic truncation does, thus providing a formally grounded account of when and why truncation misleads.

Formalizing an effect this way buys generalization for a single perceptual effect, but deriving a model for every heuristic does not scale, raising the question of whether we can leverage existing computational models to serve as cheap proxies instead.

Chapter 4¹³ pursues exactly this: rather than building a perceptual model from domain-specific theory, it asks whether existing computational models of visual perception, that is, deep neural network features trained on natural scenes, transfer to information visualization. If the features that drive perceptual similarity for natural images also predict similarity judgments for data graphics, this would provide a computationally cheap proxy for a quantity that is expensive to measure behaviorally. The chapter compares multiple deep-feature similarity metrics against human judgments across visualization types and tasks, characterizing where these proxies succeed and where they break down. One of the findings is that the task used to elicit a judgment shapes the representation it recovers, so measuring and comparing at the level of tasks is too coarse.

Chapter 5¹⁴ responds by going beneath the level of tasks and proposes the central contribution of this dissertation. It asks whether the process of reading a visualization can be decomposed into elementary *visual decoding operators*,¹⁵ each with estimable bias and variance. Complex tasks are expressed as sequences of operators whose errors propagate traceably, enabling predictions for new chart×task combinations by specifying which operators they require rather than running new experiments. The chapter provides an existence proof: operators characterized on PDF and CDF charts compose to predict scatterplot mean-

¹³ Chapter 4 varies V across different visualization designs and tests whether computational similarity metrics predict human similarity judgments.

¹⁴ Chapter 5 decomposes T into sequences of elementary perceptual operators performed on V , characterizes each operation's error profile, and compose them to predict task performance on new $V \times T$ combination.

¹⁵ i.e., perceptual primitives such as projecting from a point to an axis, locating a curve's peak, or bisecting an area

estimation performance with no parameters fit to the scatterplot data, a prediction across chart type, dimensionality, and analytic goal.

The three studies thus pursue a single goal — formulating computational models that generalize — through progressively more general means, and the operator framework is its most complete realization. The dissertation concludes with a general discussion (**Chapter 6**) reflecting on what these studies, taken together, reveal about the path toward a broader library of visual decoding operators and more ambitiously, how people could and *should* be doing empirical research in information visualization.

2 GENERAL BACKGROUND AND RELATED WORK

2.1 GRAPHICAL PERCEPTION & COMPREHENSION

Graphical perception was introduced by Cleveland and McGill [23] as “the visual decoding of information encoded on graphs”. Building on Cleveland and McGill’s empirical methodology, later experiments in visualization sought to quantify how effectively viewers perceive different properties encoded using various visual features [166]. For example, Kim and Heer [94] considered the impact and interaction between data distribution and task type, Heer and Bostock [69] extended the experiment to a crowd-sourcing environment using Amazon Mechanical Turk. Other studies have looked into other aspects beyond the accuracy of decoding information, such as memorability [9] and recall [10].

Rensink [146] summarized how the typical procedures of an empirical visualization research is conducted: 1) Find or create a simple version of a commonly used visualization; 2) Select one of its functions or tasks; 3) Measure performance under controlled conditions; 4) Find regularities in this performance; and finally, 5) Find explanations for these regularities.¹⁶ Rensink argued that the “first-wave” of work on visualization typically had the first three of these, though I would think that Cleve-

This chapter introduces the foundational concepts and prior work that inform the computational models developed in this dissertation. Each subsequent chapter includes its own detailed related work section; the material here covers ideas that span multiple chapters.

¹⁶ Rensink [146] referred to this as the “five elements” and not necessarily as an ordered process, and argued that these elements “need not be considered in the exact order presented here, but all should ideally be present to some extent.”

land and McGill [23] also incorporates elements from the later two. See Quadri and Rosen [144] for a comprehensive review of perception-based visualization studies since 1980.

Building upon early empirical work on graphical perception by researchers like Cleveland and McGill [23], part of the field progressed towards developing models to *explain the underlying cognitive processes* involved in understanding graphs.¹⁷ Simkin and Hastie [160], for instance, extended this work by emphasizing the role of judgment tasks and proposing *elementary mental processes*, setting the stage for more detailed models such as Pinker [136], Lohse [111], Gillan and Lewis [56], Peebles and Chang [134], and Padilla et al [126]. See Fox [46] for a more detailed review of theories and models in graph comprehension.

2.2 STRUCTURE PRESERVATION AND REPRESENTATIONAL EQUIVALENCE

A recurring principle in visualization theory is that the visual form should preserve relevant structure from the data it represents. Several formalizations of this idea inform the models developed in this dissertation.

Kindlmann and Scheidegger [96] proposed Algebraic Visualization Design (AVD),^{fig. 6} a framework for reasoning about visualization design through intrinsic symmetries. AVD explores the effect of changes in data on resulting images and is composed of three principles, including the principle of visual-data correspondence, which advocates for matching mathematical structure in data with mathematical structure in the perception of the visualization. Demiralp et al. [34] formalized a related idea: they defined a visualization as a function mapping from a domain of data points to a range of visual primitives, and argued that a “good” visualization is one where the embedded visual elements preserve

¹⁷ Generally speaking, these models emphasize the complex interaction between bottom-up (perceptual) and top-down (knowledge-based) processes. Comprehension involves the visual system encoding display features, and users applying knowledge, such as graphical conventions and schemas, to interpret the information. Successful comprehension requires the integration of these processes, while incomplete knowledge or distractions can lead to errors.

$$v \circ r_2 \circ \alpha = \omega \circ v \circ r_1$$

$$\begin{array}{ccccc} D & \xrightarrow{r_1} & R & \xrightarrow{v} & V \\ \alpha \downarrow & & & & \downarrow \omega \\ D & \xrightarrow{r_2} & R & \xrightarrow{v} & V \end{array}$$

Figure 6: Screenshot of the framework from Kindlmann and Scheidegger [96].

structures present in the data domain.^{fig. 7} Though not mathematical, Zacks and Tversky [203] expressed a similar view, distinguishing *data correspondence*¹⁸ and *message correspondence*¹⁹. They showed that there was “a strong tendency to portray discrete comparison descriptions as bars and trend assessment descriptions as lines”.

The structure-preservation principle motivates the truncation model in Chapter 3, where y-axis truncation is analyzed as a visual form transformation and the central question is whether it preserves or distorts the data relations relevant to the viewer’s task.

However, preserving structure is necessary but not sufficient. Larkin and Simon [105] drew a foundational distinction between informational equivalence and computational equivalence. Two representations are *informationally equivalent* if all the information inferable from one is also inferable from the other. But informationally equivalent representations need not be computationally equivalent: the ease and speed of inference depend on the available information-processing operators, which can differ in their capabilities for pattern recognition, direct inference, and search control.

Peebles and Cheng [133,134] demonstrated this empirically with function graphs and parametric graphs — informationally equivalent representations that produce “consistent and substantial differences” in response time and error rate for common graph-reading tasks. Their ACT-R/PM models attribute these differences to distinct cognitive-perceptual processing sequences: different representations require different sequences of operations, even when the underlying information is identical.

This distinction is foundational for the present dissertation. The visual decoding operators of Chapter 5 are, in Larkin and Simon’s terms,

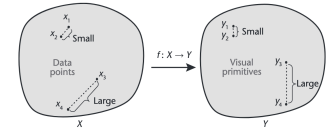


Figure 1. Visual embedding is a function that preserves structures in the data (domain) within the embedded perceptual space (range).

Figure 7: Screenshot of Figure 1 from Demiralp et al. [35].

¹⁸ ... between graph type and data type

¹⁹ ... between graph type and the intended message

the information-processing operators that determine computational equivalence. By characterizing these operators empirically — measuring their bias and variance — the framework provides a mechanism for predicting why informationally equivalent representations differ in perceptual performance, and by how much.

2.3 EMPIRICAL EVALUATION AND MODELING

2.3.1 EVALUATING VISUALIZATIONS

Mackinlay’s early influential work [115] has impacted not only how we think about creating graphical presentations (i.e., the *expressiveness* criteria), but also how we *evaluate* the created graphical presentations²⁰. Beyond these “local” effectiveness measures, other work, such as Lam et al. [104], have discussed broader evaluation scenarios such as visual data analysis and collaborative data analysis. See Elliott et al. [40] for a detailed account of the design space of methods and analysis approaches in visualization evaluation.

²⁰i.e, the *effectiveness* criteria, which determines “whether a graphical language exploits the capabilities of the output medium and the human visual system”

2.3.2 BAYESIAN HIERARCHICAL MODELING

All empirical chapters in this dissertation use Bayesian hierarchical models to analyze experimental data. This choice reflects several considerations: Bayesian methods provide full posterior distributions over parameters rather than point estimates, naturally accommodate hierarchical structure (e.g., observations nested within participants), and handle the kinds of complex response distributions (e.g., skewed, heavy-tailed) that arise in perceptual tasks.²¹ The modeling approach throughout is influenced by McElreath’s textbook [118] and follows the notation conventions of Gelman and Hill [54] for multilevel models. Specifically, I adopt the following notation throughout unless otherwise specified:

²¹See also Kay et al. [89] for a detailed argument for the advantages of Bayesian methods in HCI research.

- Units $i = 1, \dots, n$. By *units* we mean the smallest items of measurements, usually the row in a dataset
- Outcome measurements $y = (y_1, \dots, y_n)$. There are the unit-level data being modeled.
- Groups $j = 1, \dots, J$. This works for a single level of grouping.
- This is perhaps the one that is most unique, the one that I personally found most helpful but might be alien to those who are unfamiliar. Index variables $j[i]$ code group membership. For example, if $j[35] = 4$, then the 35th unit in the data $i = 35$ belongs to group 4.
- Coefficients are sometimes written as a vector β , sometimes as α, β .

3 A SYSTEMATIC EXPLORATION OF Y- AXIS TRUNCATION

This chapter is based on work published at *CHI 2025* with Matthew Kay [112].

Supplementary materials are available at <https://osf.io/k4hjd>.

²² My personal favorite is the decision tree included in this [blogpost](#).

3.1 INTRODUCTION

This chapter presents my first attempt at building a computational model of visualization perception. I was dissatisfied with existing work on y-axis truncation, which treated truncation as categorically deceptive without formalizing *when* or *why* it misleads. I was also intrigued by the many guidelines available online²² about whether, when and how to (un)truncate the y-axis. While some guidelines reason that it is never acceptable to truncate [79,173], others argue that it might be acceptable for some chart types but not others [6,17]. Additional guidelines such as Brinton [13] recommend no truncation when the reader’s focus is on the relative amount of change, while other guidelines like Skelton [162] recommend no truncation when the reader’s focus is on the absolute rate of increase/decrease. Only some of this guidance has been tested by empirical studies, which find that y-axis truncation results in quantifiable differences in how people subjectively interpret the size and significance of effects, even when behavioral and visual interventions (e.g. broken axes) are employed [27,130,193,198].

I wanted a way to understand precisely when it may be appropriate to truncate or not.²³ I did so by systematically analyzing how y-axis truncation affects visual properties and identify those that remain invariant after truncation. This work also introduced the $D \leftrightarrow V \leftrightarrow T$ framework, which becomes a recurring framing/thought device in later chapters.

3.2 RELATED WORK

3.2.1 GRAPHICAL PERCEPTION OF BAR CHARTS

Cleveland and McGill [23] tested the impact of bar chart designs (simple and stacked) on participants' ability to estimate the ratio between the lengths of two bars and found that the designs in which the bars are *aligned along a common baseline* scored better than the designs where the comparisons are unaligned. They hypothesized that this difference is due to the use of two different visual estimation strategies — for aligned bars, viewers make a visual comparison of *positions* while for unaligned bars, viewers make a much less accurate visual comparison of *lengths*. They suggest encoding information with *position* instead of *length* for better perceptual accuracy. Cleveland and McGill [23] also found that comparisons between adjacent bars are more accurate than between widely separated bars.

Talbot et al. [169] replicated Cleveland and McGill [23]'s finding that separating bars in space makes the comparison of their height more difficult. They also looked at the effects of distractor bars, but their results were not conclusive. Zhao et al. [208] investigated how neighboring bars affect the perception of order in bar charts and found that while neighborhood effects do exist in rank estimation tasks, the effect size of neighborhood effects is small compared to other data-inherent effects. From these prior results, we decided not to explicitly model distractor

²³... and also synthesize where existing guidelines align and diverge. With this many guidelines they have to share something in common.

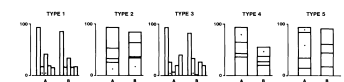


Figure 8: Figure 4 from Cleveland and McGill [23], demonstrating the sample stimuli used in the position-length experiment.

bars because we think the effect size is going to be small. We also did not include the distance between the two groups of bars in our analysis because it is randomized across all conditions.

In our experiment, we had two comparison tasks. The first comparison task asked the participants to compare the difference in heights between two groups of bars, which would require the participants to make an unaligned length comparison. The second comparison task asked the participants to compare the percentage change²⁴ between two groups of bars, which would either require the participants to use both unaligned and aligned length comparison or a similar perceptual judgment to that of a perceptual judgment of correlation.

²⁴i.e.,

$$\frac{\text{tall bar} - \text{short bar}}{\text{short bar}}$$

3.2.2 DECEPTIVE VISUALIZATION & Y-AXIS TRUNCATION

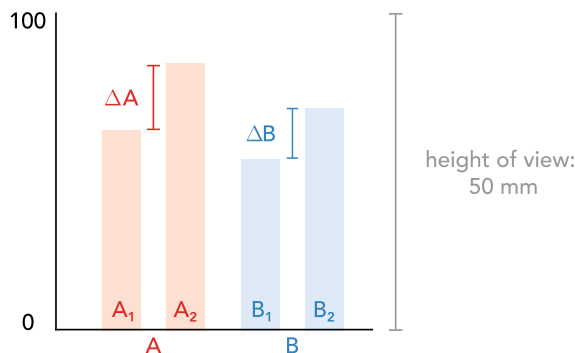
Pandey et al. [130], the first work to empirically examine deceptive techniques in data visualizations, defined *deceptive visualization* to be “a graphical depiction of information ... that may create a belief about the message and/or its components, which varies from the actual message”. Heer and Correll [70] also presented a definition that emphasizes the encouraged interpretation that deviates from the actual message in the data. Among the studied deceptive visualization practices, we focus on **y-axis truncation**, a practice of beginning the vertical axis at a value other than 0. Prior empirical research [27,130,193,198] has studied this topic by focusing on the *subjective* perception of change. These papers find that truncated bar charts persistently increase the perceived magnitude of trends, despite behavioral interventions (e.g., warning and teaching people about y-axis truncation before showing them the stimuli [198]) and visual interventions (e.g., adding broken axes to emphasize the truncated y-axis [28]). As a result, these papers suggest either completely avoiding y-axis truncation for bar charts [198]

or truncating it in a way that aligns with the task at hand or conveys meaningful effect size to the viewers [70,193]. In our paper, we quantify the degree of this increased perceived magnitude by asking participants to estimate the **absolute change**²⁵ or the **relative change**²⁶ between bars. We delay a more detailed discussion of prior recommendations and guidelines to Section 3.6.1 to better summarize motivations for these recommendations as well as how our motivations and recommendations fit in with existing ones. We note that truncated axes are not unique to bar charts. Nevertheless, arguments for other chart types tend to focus on how truncation changes the **aspect ratio** of these charts. Existing research has examined how to set the optimal aspect ratio for line charts [25,68,167,185] and for scatter plots [44]. Our work does *not* examine the effects of aspect ratio changes, as we do not manipulate the aspect ratios of the bars or the bar chart itself. We truncate the y-axis, which alters the length of bars within the bar chart. While one could argue that y-axis truncation alters the aspect ratio of the position encodings of bars, our experiment *primarily utilizes length judgments*, thus, the changed position encoding is not relevant to our work.

²⁵i.e., difference

²⁶i.e., percentage change

When we have a data set $\begin{cases} A_1: 63 \\ A_2: 84 \end{cases}$, $\begin{cases} B_1: 54 \\ B_2: 70 \end{cases}$ and plot it into a bar chart ...

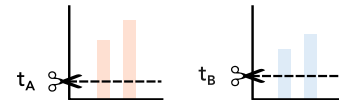


... the visual relation is consistent with the **concordant data relation**:

$$\Delta A > \Delta B$$

$$\Delta A / A_1 > \Delta B / B_1$$

If we treat **group A** and **group B** as separate bar plots, and truncate them differently with t_A and t_B ...



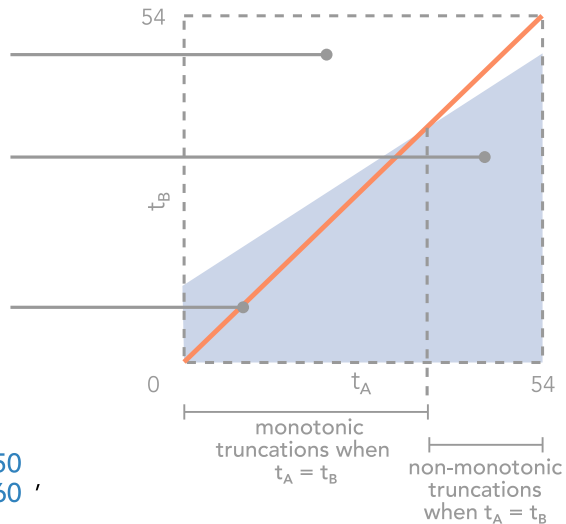
then ...

in the **non-shaded region**, all truncations render a reversed data relation

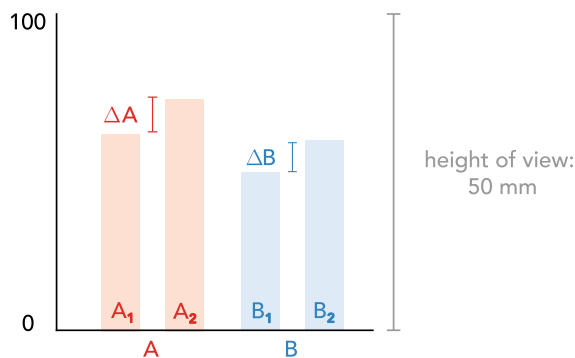
in the **shaded region**, all truncations retain the data relation

truncations on the 45 degree line imply that $t_A = t_B$.

This is the only case in the paper, since **group A** and **group B** are *always* in the same frame.



Likewise, if we have the data set $\begin{cases} A_1: 62 \\ A_2: 73 \end{cases}$, $\begin{cases} B_1: 50 \\ B_2: 60 \end{cases}$, the corresponding bar plot is ...



.... which is consistent with the **discordant data relation**:

$$\Delta A > \Delta B$$

$$\Delta A / A_1 < \Delta B / B_1$$

truncating when $t_A = t_B$ would **always lead to a monotonic truncation**. In other words, it is impossible to perform non-monotonic truncations on discordant data relations.

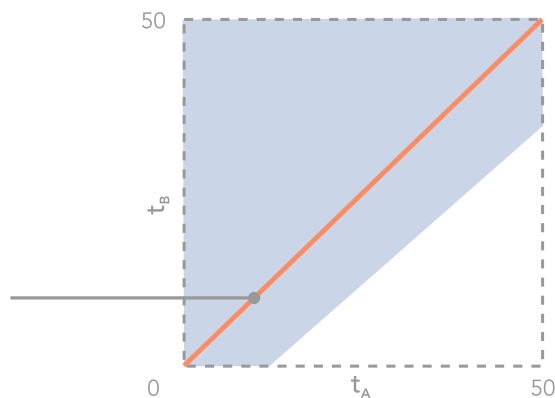


Figure 9: What are monotonic truncations and what are non-monotonic truncations?

3.3 A MODEL FOR Y-AXIS TRUNCATION

We synthesize the ideas from Demiralp et al. [34], Kindlmann and Scheidegger [96], and Hu et al. [76] and form the following model for reasoning about visualizations:

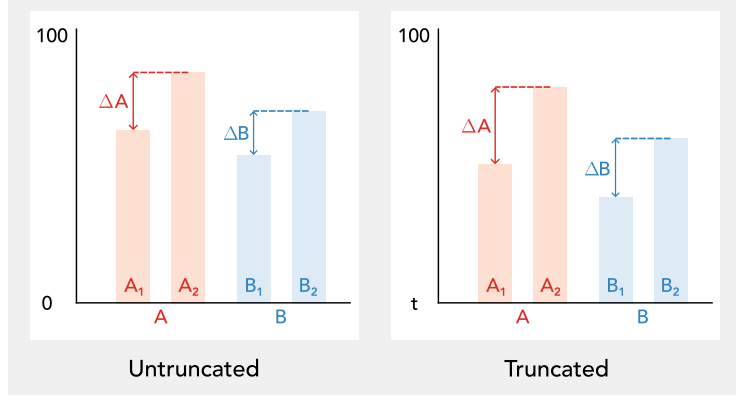
$$\text{Data } D \leftrightarrow \text{Visual forms } V \leftrightarrow \text{Tasks } T$$

Given a particular data set d in the space of all possible data sets D , d could be mapped to a set of different visual encodings $V_d \subset V$, some more structure-preserving (or having better “data-visual” correspondence) than others. Each feasible visual configuration $v \in V_d$ can be used to answer tasks, be it low-level or high-level. Some visual configurations provide more affordance for certain tasks and may support better perceptual accuracy. Others might not support perceptual accuracy but better decision-making [87] or require a lower level of cognitive effort.

This model allows us to go one step further beyond matching mathematical structure in data with that of visualization. Instead, we want to examine the nature of these correspondences between data, visual forms, and tasks and be able to characterize these correspondences. Similar to Kindlmann and Scheidegger [96], we start by introducing changes into our model — **if we alter the visual configurations in a way that preserves the inherent structures in data, how does it affect the performance of tasks?** Formally, if function $f : V \rightarrow V$ is a structure-preserving function applied to visual configurations, would $D \leftrightarrow f(V) \leftrightarrow T$ be much different than $D \leftrightarrow V \leftrightarrow T$? And what can we say about the properties of such a structure-preserving function $f(\cdot)$?

To make our question more concrete, consider the bar charts in Figure 10.

Figure 10: Left: example of untruncated bar chart. Right: example of truncated bar chart. The two charts share the same data set $\{A_1, A_2, B_1, B_2\}$



First, look at group A in the untruncated bar chart. Group A contains two elements, $\{A_1, A_2\}$, from which we can derive at least three quantities: mean $\frac{1}{2}(A_1 + A_2)$, gap $A_2 - A_1$, and ratio $\frac{A_1}{A_2}$. Let L denote the height of the viewpoint on which the bar chart materializes. Then, the gap ΔA in the data space maps to $\frac{\Delta A}{100} \times L$ in the visual space. The underlying relation between ΔA and ΔB in data space, i.e., $\Delta A > \Delta B$, still holds in the visual space:

$$\Delta A > \Delta B \Rightarrow \frac{\Delta A}{100} \times L > \frac{\Delta B}{100} \times L$$

which justifies the usage of length of bars to answer $\max\{\Delta A, \Delta B\}$.

Let's introduce a change by truncating the y-axis. Let t denote the amount truncated for the bar chart on the right in Figure 10. Then, in the truncated bar chart, the length of ΔA is now $\frac{\Delta A}{100-t} \times L$. If we simply compare the length of ΔA in the truncated bar chart to the length of ΔA in the untruncated bar chart, we would conclude that truncation leads to visual exaggeration, since $\frac{\Delta A}{100-t} \times L > \frac{\Delta A}{100} \times L$. However, using the rendered lengths of ΔA and ΔB in the truncated bar chart to answer the question of $\max\{\Delta A, \Delta B\}$ should nevertheless work since the relation that holds in data space still holds in the visual space:

$$\Delta A > \Delta B \Rightarrow \frac{\Delta A}{100-t} \times L > \frac{\Delta B}{100-t} \times L \quad (1)$$

Further, note that the above inequality in Equation 1 implies that regardless of the amount t we truncate, the relation between ΔA and ΔB in data space, $\Delta A > \Delta B$, would still hold in the visual space.

Consider another judgment task: What is the answer to $\max\left\{\frac{A_1}{A_2}, \frac{B_1}{B_2}\right\}$? In the example (Figure 10), the underlying relation in data is $\frac{A_1}{A_2} < \frac{B_1}{B_2}$. In the untruncated bar chart, we could still use the visual length of bar charts and arrive at the same conclusion:

$$\frac{A_1}{A_2} < \frac{B_1}{B_2} \Rightarrow \frac{\frac{A_1}{100} \times L}{\frac{A_2}{100} \times L} < \frac{\frac{B_1}{100} \times L}{\frac{B_2}{100} \times L}$$

However, in the truncated bar chart, if we rely on the visual length of A_1, A_2, B_1, B_2 to judge whether $\frac{A_1}{A_2} < \frac{B_1}{B_2}$, then for our judgment to be valid, we need

$$\begin{aligned} \frac{\frac{A_1-t}{100-t} \times L}{\frac{A_2-t}{100-t} \times L} &< \frac{\frac{B_1-t}{100-t} \times L}{\frac{B_2-t}{100-t} \times L} \\ \Rightarrow \frac{A_1-t}{A_2-t} &< \frac{B_1-t}{B_2-t} \\ \Leftrightarrow \left(\underbrace{A_2 - A_1}_{\Delta A} + \underbrace{B_1 - B_2}_{\Delta B} \right) \times t &< B_1 A_2 - A_1 B_2 \\ \Rightarrow t &< \frac{B_1 A_2 - A_1 B_2}{\Delta A - \Delta B} \end{aligned} \quad (2)$$

In other words, if the truncation t satisfies Equation 2, then using length in the truncated bar chart to judge $\max\left\{\frac{A_1}{A_2}, \frac{B_1}{B_2}\right\}$ should not differ from that of using length in the untruncated bar chart, since the relation between $\frac{A_1}{A_2}$ and $\frac{B_1}{B_2}$ in data is preserved by the relation in visualization.

3.3.1 DATA RELATIONS

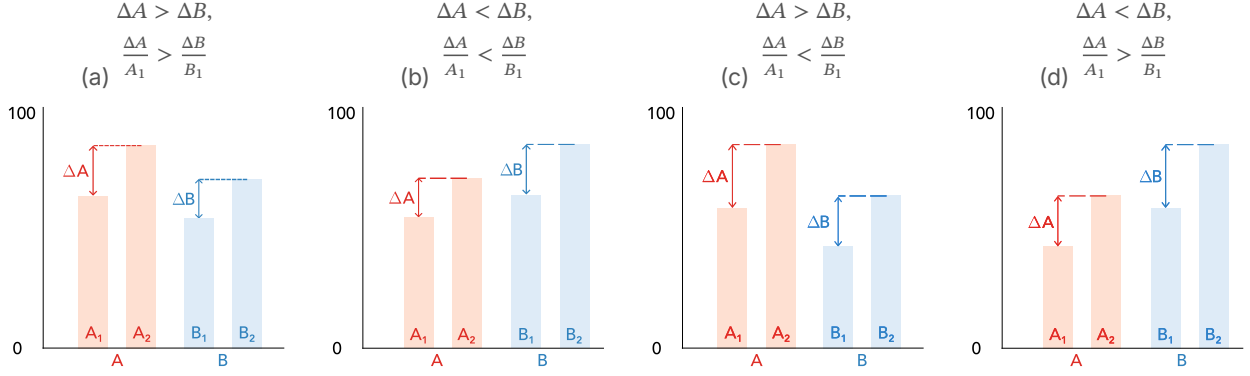


Figure 11: An example of the four kinds of data relations.

²⁷ Because all bars start at the same baseline, most of Allen’s relations do not apply. If we were examining another visual form, e.g., stacked bar charts, more of the relations in Allen’s interval algebra apply.

Among the different quantities of interest in a grouped bar chart, we choose to focus on two quantities that have been examined by prior work [23]: (1) **gap** $A_2 - A_1$, and (2) **percentage change** $\frac{A_2 - A_1}{A_1}$. To ensure completeness, we use Allen’s interval algebra [1], a calculus initially introduced for temporal reasoning but which has also found applications in visualization research [32]. Three out of the thirteen base relations of Allen’s interval algebra apply to the quantities we study: (1) $X > Y$, (2) $X < Y$, and (3) $X = Y$.²⁷ Ignoring equality, there are two relations in data space for **gaps** between groups A and B : (1) $\Delta A > \Delta B$ and (2) $\Delta A < \Delta B$. There are also two relations in data space for **percentage change** between groups A and B : (1) $\Delta \frac{A}{A_1} > \Delta \frac{B}{B_1}$ and (2) $\Delta \frac{A}{A_1} < \Delta \frac{B}{B_1}$. A combination of these relations in data space gives us a total of four kinds of **data relations** when looking at two groups of bar charts:

1. Figure 11a: $\Delta A > \Delta B, \frac{\Delta A}{A_1} > \frac{\Delta B}{B_1}$
2. Figure 11b: $\Delta A < \Delta B, \frac{\Delta A}{A_1} < \frac{\Delta B}{B_1}$
3. Figure 11c: $\Delta A > \Delta B, \frac{\Delta A}{A_1} < \frac{\Delta B}{B_1}$
4. Figure 11d: $\Delta A < \Delta B, \frac{\Delta A}{A_1} > \frac{\Delta B}{B_1}$

For ease of exposition, we group the first two data relations as **concordant data relations** (since the trend in comparing percentage changes is the same as the trend in comparing gaps) and the latter two relations as **discordant data relations**.

3.3.2 MONOTONICITY OF TRUNCATION

We define a **monotonic** y-axis truncation to be a truncation that does not result in a reversed data relation at hand (Fig. 9). Let t denote the amount of truncation performed. Then, for example, a **monotonic truncation** for a concordant data relation $\Delta A > \Delta B, \frac{\Delta A}{A_1} > \frac{\Delta B}{B_1}$ implies that

$$\begin{aligned} \frac{\Delta A}{A_1 - t} &> \frac{\Delta B}{B_1 - t} \\ \Leftrightarrow t &< \frac{\Delta B A_1 - \Delta A B_1}{\Delta B - \Delta A} \end{aligned}$$

while a **non-monotonic truncation** implies that

$$\begin{aligned} \frac{\Delta A}{A_1 - t} &< \frac{\Delta B}{B_1 - t} \\ \Leftrightarrow t &> \frac{\Delta B A_1 - \Delta A B_1}{\Delta B - \Delta A} \end{aligned}$$

Based on our definitions of data relations and monotonic y-axis truncation, it is impossible to perform a *non-monotonic* y-axis truncation for *discordant data relations* (refer to Appendix A.1.2). Put another way, for discordant data relations, it is impossible to alter the underlying data relations visually, regardless of how one truncates the y-axis.

Connecting this example back to the $D \leftrightarrow V \leftrightarrow T$ model, a *monotonic truncation* on the y-axis is simply a structure-preserving function $f(\cdot)$. We hypothesize that monotonic y-axis truncations will not affect user performance much (that is, $D \leftrightarrow \text{chart} \leftrightarrow T$ should behave similarly to $D \leftrightarrow f(\text{chart}) \leftrightarrow T$ when $f(\cdot)$ is monotonic), while non-monotonic truncations would have a more adverse impact on user performance.

3.4 METHODOLOGY

To test our hypothesis that (1) $D \leftrightarrow f(\text{chart}) \leftrightarrow T$ scores similarly to $D \leftrightarrow \text{chart} \leftrightarrow T$ when $f(\cdot)$ is monotonic, and (2) $D \leftrightarrow f(\text{chart}) \leftrightarrow T$ scores worse

compared to $D \leftrightarrow \text{Bar Chart} \leftrightarrow T$ when $f(\cdot)$ is non-monotonic, we designed our experiment with two different data-generating distributions and four different tasks. Our experiment followed a mixed factorial design. The **between-subjects** factor is the judgment task. The three **within-subjects** factors are (1) data-generating distribution, (2) data relation, and (3) type of truncation. We pre-registered our conditions, sample sizes, and analysis on osf.io. The remainder of this section details our choices for each experiment factor.

3.4.1 JUDGMENT TASKS

We used four judgment tasks with varying levels of cognitive and perceptual demand, following recent studies on how tasks affect user performance [\[153\]](#).

1. Compare Gaps — *Which month has a greater difference in profit between 2020 and 2021, [month A] or [month B]?*
2. Compare Ratios — *Which month shows a greater percentage change from 2020 to 2021, [month A] or [month B]?*
3. Estimate Gap — *What is the difference between the profit in 2020 and the profit in 2021 for [month]?*
4. Estimate Ratio — *What is the ratio between the profit in 2020 and the profit in 2021 for [month]?* (see example stimuli in Figure 12)

Please use the graph below to answer the following question:
What is the **ratio** between the profit in 2020 and the profit in 2021 for April?

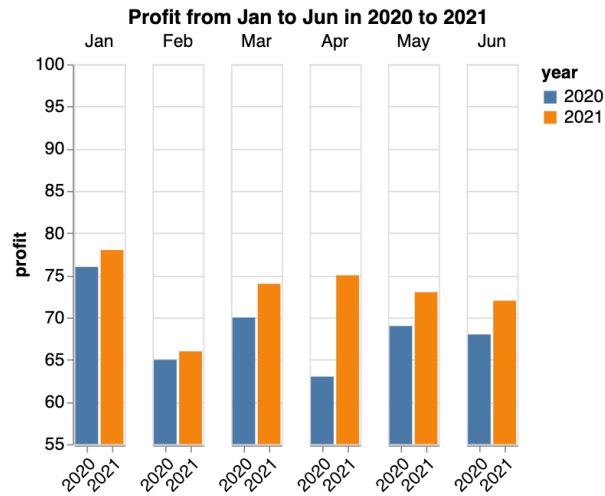


Figure 12: Sample question from the experiment. The participant is asked to estimate the ratio between the profit in 2020 and the profit in 2021 for April. Note that the y-axis is truncated to start at 55.

3.4.2 EXPERIMENT STIMULI AND DATA GENERATION

We generated visual stimuli using Vega-Altair²⁸ and hosted them on an online survey created via the Qualtrics platform. We framed our questions in the context of comparing profits between 2020 and 2021 for the first six months of the year to simulate a real-world setting where grouped bar charts are used.

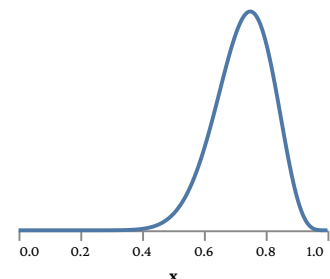
²⁸<https://altair-viz.github.io/index.html>

3.4.2.1 DATA-GENERATING DISTRIBUTION

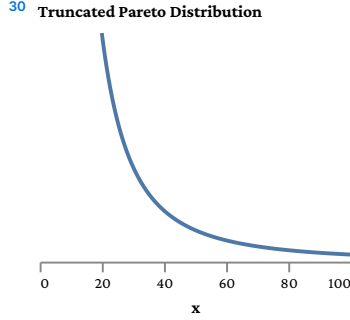
Existing research has shown that data impacts task effectiveness, yet prior work on y-axis truncation either did not specify how their data was generated or used only normal distributions. To test if different data-generating distributions have different impacts on y-axis truncations, we selected two distributions with bounded support between 0 and 100:

1. We used a **scaled Beta distribution** with parameters $\alpha = 16$, $\beta = 6$ as an approximation for data that is roughly normally distributed. We thought it necessary to include this data-generating distribution since (approximately) normal distributions are among the most commonly seen distributions in the wild.²⁹

²⁹ Scaled Beta Distribution



2. We used a **truncated Pareto distribution** with parameters $b = \log_4(5)$, $c = 5$, $\text{scale} = 20$ to model real-life scenarios where we have outliers in the data.³⁰



We sampled data using the Python packages `numpy` [63] and `scipy` [180].

3.4.2.2 DATA RELATION

As detailed in Section 3.3.1, we have four different kinds of data relations grouped into two categories: **concordant** and **discordant** data relations. For any data set, there are four potential judgment tasks that it could answer. The comparison tasks require two months to examine. These are selected by going through all $\binom{6}{2} = 15$ pairs of months, finding the pairs that satisfy the data relation we want to test, and selecting the pair that has the minimum absolute difference between their gaps, i.e., the pair that has minimum $|\Delta A - \Delta B|$. The estimation tasks require one month to examine, and we randomly selected either the month that has the biggest percentage change or the biggest difference between the two bars. We did not control for the distance between the two pairs of bars — they could be next to each other or separated by four other pairs of bars. We adjusted the ordering of bars such that for the two months for comparison, within each month, the left bar is always the shorter bar.

3.4.2.3 TYPE OF TRUNCATION

As detailed in Section 3.3.2, there are three types of allowed truncation: No truncation, monotonic truncation, and non-monotonic truncation. The actual amount of truncation was *randomly sampled* from the interval of monotonic truncations and the interval of non-monotonic truncations, respectively (§ 3.3). This differs from prior work where the authors either do not specify how they chose the truncation value [129] or start the y-axis from fixed values [27].

3.4.3 EXPERIMENT PROCEDURE

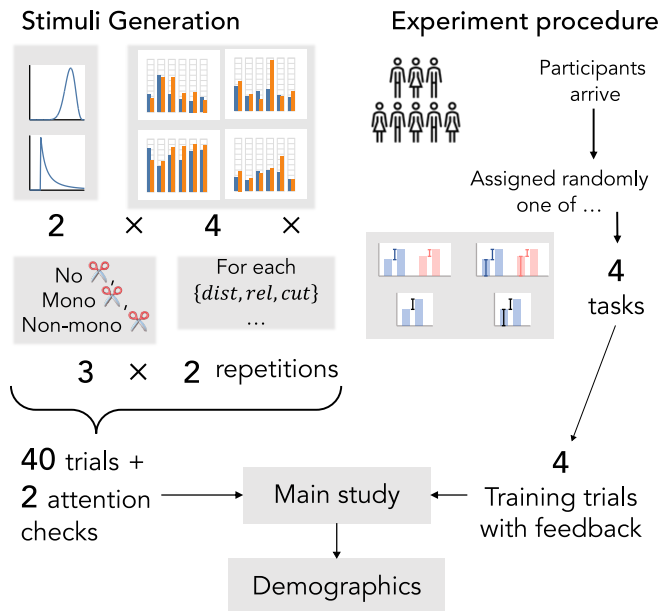


Figure 13: Overview of experiment procedure. Participants were first randomly assigned to one of four tasks, then to the same set of training stimuli in the same order, and then to the same set of 20 conditions but different testing stimuli. Observe that $2 \times 4 \times 3 \times 2 = 48 \neq 40$ trials — this is because it is impossible to truncate non-monotonically for the two discordant data relations, thus resulting in $2 \times 4 \times 3 \times 2 - 2 \times 2 \times 1 \times 2 = 48 - 8 = 40$ trials.

- **Introduction and Consent.** Participants began by reading about the study goals and compensation. Participants were instructed to not rely on any rulers and to refrain from taking long breaks while answering the questions.
- **Training.** Participants were given 4 practice trials. All participants were assigned the same set of training stimuli, displayed to them in the same order. After answering, participants will see their responses displayed alongside the actual answers for the trial. For example, a participant assigned the task Estimate Ratio will see “You have entered [...] as your answer. The actual ratio between the profit in 2020 and the profit in 2021 for [month] is [...]%.”
- **Testing.** Participants were given 42 testing trials, 2 of which were attention checks. The first attention check was inserted randomly into the first 20 trials, and the second attention check was inserted randomly into the remaining 20 trials. The attention check stimuli were the same for all participants.
- **Demographics and optional written feedback.** Participants were asked to provide demographic information on their age and gender. Participants were also asked to provide optional feedback on the strat-

egy they used for their assigned perceptual task, as well as whether they followed the instructions in the survey.

3.4.4 PARTICIPANTS AND EXCLUSION CRITERIA

We recruited participants in an IRB-approved study on [Prolific.co](https://prolific.co) who are between the ages of 18 to 65, have normal or corrected-to-normal vision, speak fluent English, and reside in the United States. The study was distributed to a balanced sample of participants.

We excluded participants who failed to correctly answer both of the attention check questions for comparison tasks and participants who provided estimates that were more than 5 units away from the actual answer for estimation tasks. We also excluded one participant who answered “no” to the question in the feedback about whether they followed the instructions or not. Due to a data collection issue, we had two pairs of participants who were assigned to the same task and the same stimuli. We excluded the participants who started the survey later from our analysis.

In total, we collected data from 136 participants, and after excluding 11 participants for failing the attention checks, 1 participant for not following the instructions, and 2 participants for data collection issues, we had a total of 122 participants for analysis. Successful participants were compensated with an average reward of \$17.02 / hour, exceeding the US minimum wage. The median completion time was 17 minutes.

3.4.5 ANALYSIS APPROACH

We used multi-level Bayesian regression to analyze our results. Specifically, we used **logistic regression** for the comparison tasks and **hurdle log-normal regression** for estimation tasks. The hurdle log-normal distribution is a modified log-normal distribution that allows zeros by modeling the probability of zero as a separate process. For computing

and presenting the results, we used the following R packages: `brms` [15], `tidybayes` [90], `rstan` [163], and `tidyverse` [190].

All of our models are random effects models. Due to the nature of the small sample size we collected for each task, we believe it is best to be conservative in our modeling approach, and thus adopted random effects to maximizing partial pooling of information with the amount of data we had [55].

Let $\text{PID}[i]$ denote the participant ID associated with the i -th observation. Let $\text{cut:dist:rel}[i]$ denote the combination of the type of truncation, data-generating distribution, and data relation for the i -th observation. For the two **comparison tasks**, let y denote the binary dependent variable and π denote the probability of success. Then, we have the following model:

$$y \sim \text{Bernoulli}(\pi)$$

$$\text{logit}(\pi) = \beta + \beta_{\text{PID}[i]} + \beta_{\text{cut:dist:rel}[i]}$$

For the two **estimation tasks**, let $y_i := |\text{response}_i - \text{answer}_i|$ denote the numerical response variable, i.e., y_i is the absolute error for the i -th observation. Let π denote the probability of zero. Then, we have the following model:

$$y \sim \text{Hurdle Log-Normal}(\mu, \sigma, \pi)$$

$$\mu = \bar{\beta} + \beta_{\text{PID}[i]} + \beta_{\text{cut:dist:rel}[i]}$$

$$\text{logit}(\pi) = \bar{\beta} + \beta_{\text{PID}[i]} + \beta_{\text{cut:dist:rel}[i]}$$

See Appendix A.1.1 for details on `brms` formulae and prior.

3.5 RESULTS

3.5.1 DATA PRELIMINARIES

Among the 122 participants, there were 58 female participants (47.5%), 59 male participants (48.4%), and 5 (4%) non-binary participants. The

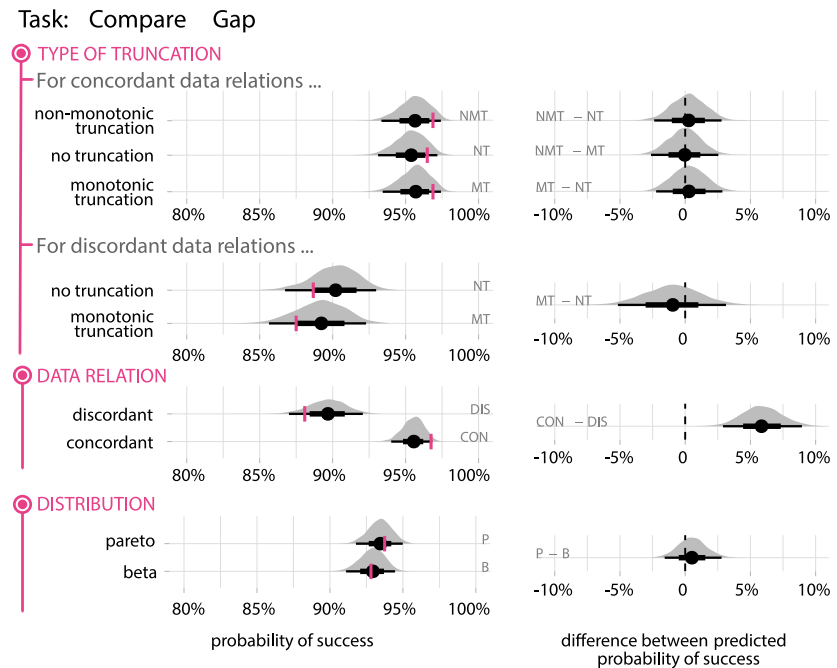
self-reported age of participants ranges from 20 to 60, with a median age of 32 and a mean age of 35.31. The average time for a participant to answer a question for each task, ordered by magnitude, is Compare Gaps (13.76) < Estimate Gap (13.86) < Compare Ratio (17.99) < Estimate Ratio (20.55).³¹

³¹ average time in seconds.

3.5.2 GAP

3.5.2.1 COMPARE GAP

Figure 14: *Left*: Posterior density, median, 66% and 95% quantile interval of the probability of correctly answering. *Right*: Expected difference in the means between different values of a variable. **Pink ticks** represent the percentage of correct response answers.



The 32 participants performed well for Compare Gap (Fig. 14).

We found no significant difference in performance between the different types of truncation: for concordant data relations, the difference between non-monotonic truncation and no truncation has a mean of 0.26% with 95% CI $[-1.89\%, 2.39\%]$, and the difference between monotonic truncation and no truncation has a mean of 0.29% with CI $[-1.83\%, 2.41\%]$. For discordant data relations, the difference between monotonic truncation and no truncation has a mean of -0.99% with CI $[-4.53\%, 2.52\%]$.

We also found no significant difference in performance between a Beta distribution and a truncated Pareto distribution, given that the difference in the means between the two is 0.53% (CI $[-1.25\%, 2.38\%]$).

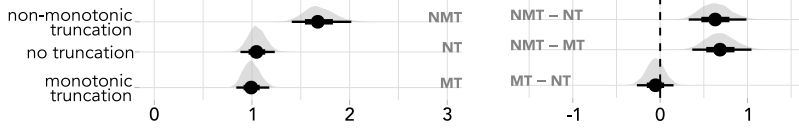
We did find a difference between performance for discordant data relations and concordant data relations: on average, concordant data relations lead to a higher probability of correctness compared to discordant relations, with a difference of mean 5.88% (CI $[8.46\%, 3.41\%]$).

3.5.2.2 ESTIMATE GAP

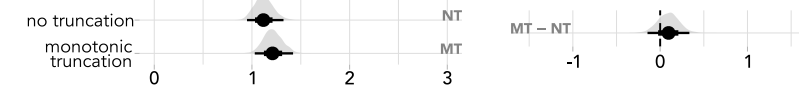
Task: Estimate Gap

TYPE OF TRUNCATION

For concordant data relations ...



For discordant data relations ...



DATA RELATION



DISTRIBUTION

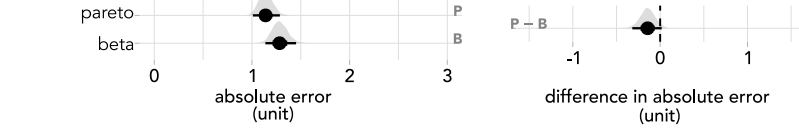


Figure 15: *Left*: Posterior density, median, 65% and 95% quantile interval of absolute error. *Right*: Expected difference in the means between different values of a variable.

The 31 participants performed fairly accurately for the task of Estimate Gap (Fig. 15): 52% of responses had an absolute error of 0, and the posterior distributions show a high concentration of absolute error around 1. For both concordant and discordant data relations, we found no significant difference in performance between monotonic truncation and no truncation: the difference in absolute error between no truncation and monotonic truncation has a mean of -0.05 (CI $[-0.23, 0.12]$) for concordant relations and a mean of 0.09 (CI $[-0.11, 0.29]$) for discordant relations.

Similarly, there is no significant difference between the two different kinds of data relations or the two different types of data-generating distributions. The difference between discordant and concordant data relations has a mean of -0.08 (CI $[-0.22, 0.06]$), and the difference between Pareto and Beta distributions has a mean of -0.14 (CI $[-0.29, -0.01]$).

However, where non-monotonic truncation is allowed (in concordant data relations), the difference in absolute error between non-monotonic truncation and no truncation has a mean of 0.635 (CI $[0.371, 0.926]$), suggesting that non-monotonic truncation causes a small exaggeration in people's estimation of gaps between groups of bars.

3.5.3 RATIO

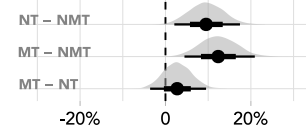
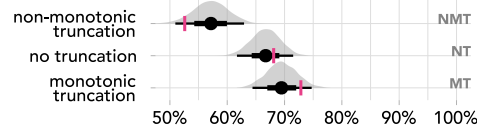
3.5.3.1 COMPARE RATIO

Figure 16: *Left*: Posterior density, median, 66% and 95% quantile interval of the probability of correctly answering. **Pink ticks** represent the percentage of correct response answers. *Right*: Expected difference in the means between different values of a variable. Similar to our results for Compare Gap in Figure 14, the median of posterior predictions (represented by black dots) are closer together than the raw data (represented by **pink ticks**), which demonstrates *model shrinkage*. This will make estimates of differences more conservative and is thus suitable for the sample size we've collected.

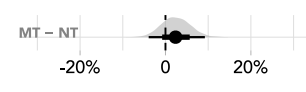
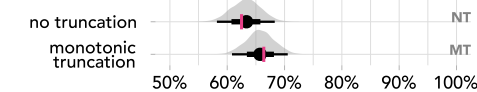
Task: Compare Ratio

TYPE OF TRUNCATION

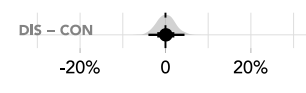
For concordant data relations ...



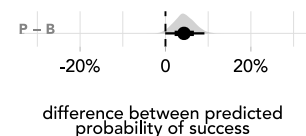
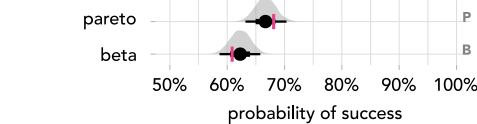
For discordant data relations ...



DATA RELATION



DISTRIBUTION



The 29 participants performed on average much worse in Compare Ratios than Compare Gaps (Fig. 16). For both concordant and discordant relations, we found that monotonic truncations slightly improve

the probability of success compared to no truncation – for concordant data relations, the difference in probability of success between monotonic truncation and no truncation for concordant data relations has a mean of 2.89% (CI $[-2.43\%, 8.38\%]$) and for discordant data relations has a mean of 2.44% (CI $[-3.18\%, 8.19\%]$).

We also found that non-monotonic truncation decreases the probability of success compared to no truncation, with the difference between the two having a mean of -12.5% (CI $[-19.5\%, -5.47\%]$).

We did not find a significant difference in performance between different kinds of data relations — between concordant and discordant relations, the difference in probability of success has a mean of 0.08% (CI $[-3.51\%, 3.65\%]$). We found a slightly higher probability of success in Pareto distributions compared to Beta distributions, with the difference between the two having a mean of 4.47% (CI $[0.83\%, 8.32\%]$).

3.5.3.2 ESTIMATE RATIO

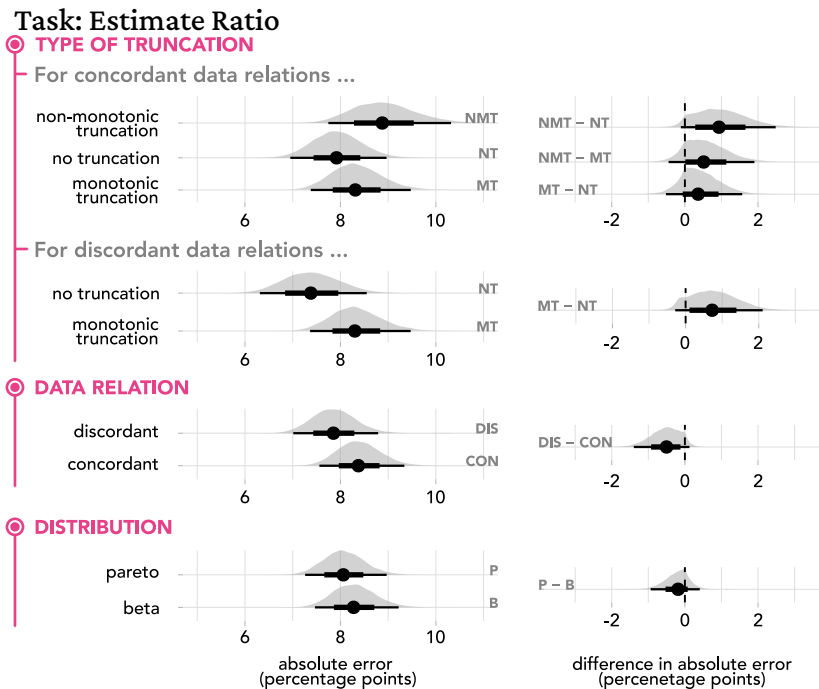


Figure 17: *Left:* Posterior density, median, 66% and 95% quantile interval of the absolute error for task Estimate Ratio. *Right:* Expected difference in the means between different values of a variable.

Previously, we saw a 13% decrease in the predicted probability of success between non-monotonic truncation and no truncation and a 3% increase between monotonic truncation and no truncation. Could this be due to how y-axis truncation alters the precision of the estimation of ratios (Fig. 17)?

We found that no truncation has a smaller absolute error compared to monotonic truncation for both concordant and discordant data relations – the difference in absolute error between the two is 0.42 (CI $[-0.37, 1.35]$) for concordant relations and 0.94 (CI $[-0.02, 2.04]$) for discordant relations. We also found that no truncation has a smaller absolute error compared to non-monotonic truncation — the difference in absolute error between the two is 0.99 (CI $[-0.01, 2.22]$). This implies that truncation, regardless of its monotonicity, is likely to bias the estimate of percentage change upwards, but non-monotonic truncations have larger absolute errors compared to monotonic truncations.

For different data relations, we found that discordant data relations have a smaller absolute error compared to concordant data relations — the difference between the two has a mean of -0.53 (CI $[-1.25, 0.05]$). We did not find much difference between the two kinds of data-generating distributions. The difference between Pareto and Beta in predicted absolute error has a mean of -0.22 (CI $[-0.81, 0.29]$).

3.5.4 SUMMARY

At the end of Section 3.3.2, we proposed two hypothesis:

1. Monotonic truncation would *not* impact $D \leftrightarrow f(\text{bar chart}) \leftrightarrow T$
2. Non-monotonic truncation would *adversely* impact $D \leftrightarrow f(\text{bar chart}) \leftrightarrow T$

We found that for tasks that examine a quantity whose data relation *unaffected by truncation* (i.e., gap), **truncation has little to no effect on user performance, regardless of whether it is monotonic or not.**

Specifically, for Compare Gaps, there is essentially no difference in user performance between no truncation and monotonic truncation, as well as no truncation and non-monotonic truncation. For Estimate Gap, non-monotonic truncation leads to a larger estimated magnitude (as also seen in prior work [27,130,193,198]), but the degree of increase is small.

On the other hand, for tasks that examine a quantity whose data relation is *affected by truncation* (i.e., ratio), **monotonic truncation has similar user performance as no truncation**, but both are much better than non-monotonic truncation. Specifically, for Compare Ratios, monotonic truncation slightly improves the probability of success compared to no truncation, while non-monotonic truncation decreases the probability of success. For Estimate Ratio, truncations (regardless of monotonicity) have larger absolute errors compared to no truncation, but non-monotonic truncation has a larger absolute error compared to monotonic truncation.

3.6 DISCUSSION

3.6.1 RECOMMENDATIONS FOR Y-AXIS TRUNCATION

In a blog post [29] accompanying their paper [27], Correll categorize y-axis truncation recommendations into four groups: the Anathemists, the Line Chart Exceptionists, the Signalers, and the Libertines. We take a different approach and employ the $D \leftrightarrow V \leftrightarrow T$ model to categorize recommendations into three broad groups: those emphasizing the $D \leftrightarrow V$ correspondence, those emphasizing the $V \leftrightarrow T$ correspondence, and those that focus on either D or T .

3.6.1.1 $D \leftrightarrow V$ CORRESPONDENCE

Arguments emphasizing the $D \leftrightarrow V$ correspondence are prominently featured in the bar vs line debate [6,17,43,97,162]. People have long

disagreed on whether the no truncation rule only applies to bar charts or includes line charts as well. Advocates of the “only bars” position argue that the correspondence between data and the **visual encoding** of bar charts is distorted by truncation. Since bar charts use either height/length [17,43,162,201] or size [6] to encode data, starting the y-axis from a non-zero value disrupts the correspondence between bar height/length/size and the original data value. On the other hand, since line charts and dot charts use position/angle to encode information [6,17,162,198], it is not necessary to hold them to the zero-baseline rule because these encodings do not get distorted [17].

3.6.1.2 $V \leftrightarrow T$ CORRESPONDENCE

While previous arguments focus on how truncation affects the $D \leftrightarrow V$ correspondence differently for different visual encodings, other arguments focus on how the **decoding** information in line charts and bar charts could be practically the same. For example, Skelton [162] and Kosara [97] argue that people decode line charts and bar charts both by the distance of the mark from the baseline, and thus a non-zero line chart poses a similar risk as a non-zero bar chart. This decoding argument is also empirically supported by Correll et al.’s finding that there is no significant difference between bars and lines when participants are decoding charts into a measure of subjective trend strength [27,29].

Other people whose arguments emphasize the $V \leftrightarrow T$ correspondence care about **visual impression** or **graphical perception**, which is operationalized by tasks T in $D \leftrightarrow V \leftrightarrow T$. For example, Huff [79] argues that in a truncated graph, “nothing has been falsified – except the impression that it gives”, and thus suggests that all charts should begin their y-axis from zero. Witt [193] reasons that “when the [perceived] visual size of the effect aligns with the actual size of the effect”, the reader exerts

less mental effort decoding information. Witt suggests setting the y-axis range to 1.5 SDs because she found “improved sensitivity in effect size” when doing so. However, Witt did not explicitly state whether her recommendation applies to bar charts because her experiment task “did not permit measuring the spontaneous impression given by the graphs”, and this impression could be either the *differences* or the *ratios* between bars.

3.6.1.3 *D* OR *T*

Other guidelines have proposed different exceptions for the no-truncation rule, focusing on either the nature of data *D* or the reader’s task/quantity of interest *T*. On the data side, a non-zero baseline is encouraged when it is meaningful or shows small but meaningful changes in the data [17,162]. On the task side, it depends on whether the visualization is communicating absolute change or relative change [13,162]. On one hand, Brinton [13] suggests that time-series line charts do not need to include the zero baselines when the reader is interested in “the absolute amount of change rather than ... the relative amount of change”. On the other hand, Skelton [162] argues that for comparisons between the relative rate of increase/decrease on the line chart, baseline zero is irrelevant, and for “communicating the actual rate of increase/decrease ..., baseline zero can be very important (and its absence potentially misleading).”

3.6.1.4 OUR APPROACH AND RECOMMENDATION

In our paper, we proposed the $D \leftrightarrow V \leftrightarrow T$ model, defined and examined properties of truncations that are structure-preserving transformations of the visual component of a $D \leftrightarrow V \leftrightarrow T$ correspondence, proposed hypotheses of the effects of such transitions, and designed experiments to test our hypotheses. As such, our recommendations stem

from a rigorous theoretical and empirical basis. Combined with findings from prior studies [27,193,198], we propose the following suggestions that we believe may extend well beyond grouped bar charts:

When the designer has control over user tasks or knows with high certainty the tasks a visualization will be used for, and when the task examines quantities that are unaffected by truncation (e.g., Compare Gaps and Estimate Gap), the designer can truncate the axis to elicit the desired subjective perception of effect size or make the visuals more task-aligned, as recommended by previous work [27,193]. However, if the tasks examine quantities that are affected by truncation (e.g., Compare Ratios and Estimate Ratio), then we suggest performing monotonic truncations or no truncations to avoid creating visual relations that are contrary to data relations.

In other scenarios **when the designer does not have control over user tasks or is uncertain about the tasks that visualizations will be used for**, then fixed truncation in a static visualization, such as tested in this paper, is likely inappropriate. However, other approaches that use interactivity to gain the potential benefits of truncation without introducing deception [147] may be effective.

Ultimately, it is the designer who faces constraints [17] and shoulders the responsibility to determine the effect size they want to show [27], but we provide a clear, empirically-verified rule on the conditions under which a designer can justify their choice for the amount they truncate. Our recommendations highlight the importance of tasks, data relations (as revealed by the examination of structure-preserving transformations in our work) and subjective perception (as revealed by previous work [70,147,193,198]) to making an informed decision about when and how to truncate the y-axis.

3.6.2 LIMITATIONS

Our research did not center on the cognitive aspect of graphical perception. We did not investigate the causes of deception, nor did we examine the relationship between inattention and deceptive visualizations. Additionally, we did not analyze how *framing*, *priming*, or *anchoring* impacts the graphical perception of bar charts.

There are many interesting directions as potential follow-ups to our work. A natural extension is to examine other tasks and see if we can offer structure-preserving transformations for those tasks. Additionally, we could investigate the feasibility of providing these transformations to tasks that involve the subjective perception of bar charts rather than numerical perception. A third extension is to impose a time limit on tasks and see if our results still hold while participants are more time-constrained. With more data, we could also characterize the Pareto-frontier of tradeoffs for different tasks. Finally, we could empirically investigate the just noticeable difference (JND) for the perception of percentage change and investigate its alignment with mathematical concepts of monotonicity.

4 APPLYING DEEP-FEATURE-BASED SIMILARITY METRICS TO INFORMATION VISUALIZATION

This chapter is based on work published at *CHI 2025*, with Angelos Chatzimpampas, Emma Alexander, Matthew Kay, and Jessica Hullman [113].

Supplemental materials can be found at <https://osf.io/dj2ms/>.

4.1 INTRODUCTION

The previous chapter demonstrated the value of modeling a specific visualization heuristic. Nevertheless, creating new models for every design rule is *unscalable*. The good news is that building computational models of visualization perception need not start from scratch; another approach is to test whether existing computational models from vision science transfer to the visualization domain. This chapter investigates deep-feature-based similarity metrics, originally developed for natural scenes, as proxies for human similarity judgments of information visualizations.

Similarity is a fundamental construct in human cognition, underlying numerous mental processes, including categorization, reasoning, and decision-making [119]. Human-perceived similarity has been studied extensively in psychology and cognitive science [99,149] and has been leveraged in applications such as image retrieval [61] and human-in-the-loop categorization [150,181] in recent years [148]. Recent studies in computer vision have shown that deep-feature-based similarity metrics

correlate well with perceptual judgments of image similarity [207] and serve as effective loss functions for tasks like image super-resolution [86] and style transfer [51]. The success of these applications in computer vision raises an intriguing question: *Can similar approaches be effectively applied to the domain of information visualization?*

Judging visualization similarity is crucial for various applications, ranging from visual search [194] and visualization recommendation [205] to automated design and sequencing [19,34]. Knowing whether a visualization is robustly discriminable across a range of datasets, i.e., whether it consistently maintains clear visual distinctions across a range of datasets, is crucial for effective communication and design. The idea that perceived visual structure should correspond with structure in the underlying data is recurring in multiple works, such as in the principle of visual-data correspondence by Kindlmann and Scheidegger [96], in the principle of congruence by Tversky [174], in similar principles proposed in multi-view visualization [142,143], and in the visual embedding model by Demiralp et al. [35]. However, such knowledge is challenging to obtain at design time and costly to evaluate empirically [179].

4.2 BACKGROUND

4.2.1 PERCEPTUAL SIMILARITY METRICS

Computer vision has long studied the problem of measuring the quality of an image for downstream tasks such as lossless compression. Due to the costly nature of running experiments to collect human data, significant research effort has been devoted to deriving an *objective* metric that can *approximate human perception*. For example, if two images appear *similar* to the human eye, then their “distance”, measured by this metric, should be close to 0.

Traditional metrics, such as Manhattan $\mathcal{L}_1$, Euclidean $\mathcal{L}_2$, Mean Squared Error (MSE), and Peak-Signal-to-Noise-Ratio (PSNR), rely on *point-wise* pixel differences, which offers computational ease but poorly matches perceived visual quality [42,57,186]. Subsequently developed patch-based metrics rely on the hypothesis that the human visual system is highly adapted for extracting *structural* information. One such metric is Similarity Index Measure (SSIM) [186], which compares “local patterns of pixel intensities that have been normalized for luminance and contrast”. One popular extension of SSIM is *Multi-Scale SSIM* (MS-SSIM) [187], which extends SSIM by evaluating image similarity at *multiple scales*, capturing both local and global structural information. Nevertheless, SSIM and MS-SSIM often do not capture the nuances of human vision when more structural ambiguity is present [154] and fall short for more complex downstream tasks, such as image generation and image synthesis.

More recently, researchers have leveraged deep neural networks (DNN) for tasks beyond classification to develop metrics such as LPIPS [207], PieAPP [139] and DISTS [37] that utilize *deep features* i.e., features obtained through deep-learning architectures trained on the ImageNet dataset [36]. One of the primary motivations for utilizing these deep-feature-based metrics is that they can transform pixel representations to a space that is more *perceptually uniform* [37]. They have also demonstrated excellent empirical performance for downstream tasks such as image super-resolution [86] and style transfer [51]. However, their application to the perceptual similarity of information visualization remains unexplored, perhaps due to the uncertainty about whether features learned from natural scenes can effectively transfer to the often

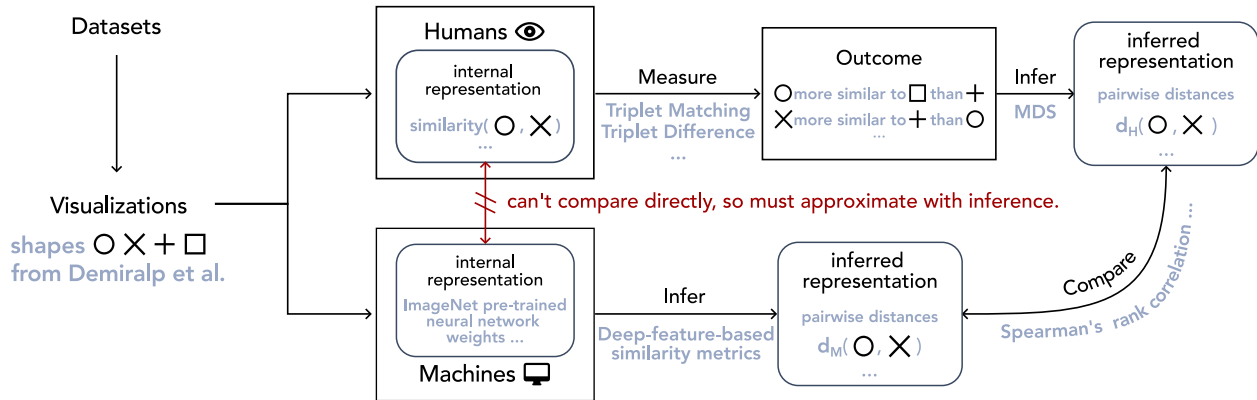
abstract and stylistic domain of information visualization — a gap this work addresses.

4.2.1.1 COLLECTING HUMAN SIMILARITY JUDGMENTS

Regardless of the underlying mechanics, the usefulness of similarity metrics is ultimately determined by how well they align with human judgment. Traditional Image Quality Assessment (IQA) databases, such as LIVE [158], CSIQ [106], and TID2013 [137], collect *Mean Opinion Scores* by asking participants to provide subjective ratings for pairwise images on a specified numerical scale (e.g., Likert scales). More recent IQA databases, such as BAPPS [207], use two-alternative forced choice (2AFC) experiments to collect similarity judgments on a dataset of images applied with low-level distortions. Beyond low-level perceptual similarity, the THINGS database collected behavioral odd-one-out similarity judgments [66,67] on everyday objects to generate interpretable object dimensions predictive of behavior and similarity, and the NIGHTS dataset [47] collected 2AFC judgments on diffusion-synthesized images perturbed along various dimensions.

Aside from the computer vision/machine learning databases mentioned above, information visualization studies have also collected data on similarity judgments (albeit on a much smaller scale). Demiralp et al. [34] collected similarity judgment via five different tasks on different visual channels, such as color, size, and shape. Pandey et al. [129] asked participants to group scatterplot thumbnails and compared their results to scatterplot diagnostics (scagnostics). Ma et al. [114] collected similarity judgment on scatterplots and trained a deep neural network, ScatterNet, to learn features that capture the human perception of scatterplot similarity. Beyond scatterplots, Gogolou et al. [58] collected similarity perception of time-series plots by asking participants to select

Figure 18: A general framework for comparing two information-processing systems (e.g., humans vs. machines), inspired by the framework proposed by Sucholutsky et al. [165]. Framework uses Demiralp et al.’s experiment on the perception of shape similarity as one concrete example [34].



Given the remarkable performance of DNNs at image classification, research across different fields — computer vision, cognitive science, neural science, to name a few — has sought to answer **whether DNNs are good models of the human visual system**. To answer this question requires *comparing* DNNs to the human visual system. Existing research has done this both at the *mechanistic* level (i.e., comparing internal representations) and *functional/behavioral* level (i.e., comparing outcomes). See Sucholutsky et al. [165] for an overview of comparing and aligning representations across different information processing systems.

4.2.2.1 FUNCTIONAL SIMILARITY

At the **outcome** level (Fig. 18), much work has focused on comparing and modeling classification performance [148]. Researchers have found that small perturbations that are imperceptible to humans can dramatically affect model classification decisions [59], and that texture and

among four options which time-series chart appears most similar to the reference chart. Others have collected judgments of visualization similarity in the form of transition costs when moving from one visualization to the next in a sequence [80,95]. In this paper, we conduct conceptual replications of the experiments by Demiralp et al. [34] and Veras and Collins [179] (that uses Pandey et al.’s data [129]).

4.2.2 COMPARING HUMAN AND MACHINE BEHAVIOR

local image features drive classifiers [5,53], whereas humans are more strongly influenced by Gestalt shape [4]. Beyond comparing classification accuracies, other work [52] has looked at additional outcome measures such as trial-by-trial error consistency.

In information visualization, Haehn et al. [62] trained four different convolutional neural network (CNN) architectures on five different visualization tasks to reproduce Cleveland and McGill’s graphical perception experiments [23]. They found that when compared to human performance baselines, CNNs are not “currently a good model for human graphical perception” — while CNNs performed better than humans in elementary perceptual tasks (e.g., estimating quantities from visual marks), humans significantly out-performed CNNs in visual relation tasks (e.g., comparing bar lengths). Yang et al. [200] trained DNNs on *human correlation judgments* in scatterplots across three studies and found that a subset of their trained DL architectures (e.g., VGG-19) has comparable accuracy to the best-performing regression analyses in prior research. The main difference between our work and Haehn et al. [62] is that we are not training network parameters on any domain-specific dataset but instead rely on pre-trained ImageNet and Stylized ImageNet weights. The main difference between our work and Yang et al. [200] is that we utilize DL architectures that are trained on natural images for classification and not on human judgment. Our metrics are therefore *information-visualization-domain-independent* and can be applied directly to any pair of visual stimuli,³² whereas the trained weights and architectures of Haehn et al. [62] and Yang et al. [200] may face challenges in generalizing to contexts beyond their specific training domains.

³²... so long as their spatial dimensions exceed 64×64 . Details in Section 4.3.

4.2.2.2 REPRESENTATIONAL SIMILARITY

At the **internal representation** level (Fig. 18), numerous techniques exist, and the most popular one is *Representational Similarity Analysis* (RSA), a multivariate technique introduced by Kriegeskorte et al. [99]. RSA compares different representations of the same set of stimuli, such as neural activity patterns, computational model outputs, and behavioral data, by computing similarity matrices for each representation and then comparing these matrices to assess how well different representations align with each other. Using RSA, Khaligh-Razavi and Kriegeskorte [91] compared the representational geometry of object recognition in the human brain with that of various computer vision models, and they found that a deep convolutional network performed best in both categorization and explaining inferior temporal cortex representations. For more on calculating similarity in psychological space, see Roads and Love [149] for an overview.

4.2.2.3 TEXTURE VS SHAPE

Most DNN models that perform well on Brain-Score [157] and other prediction metrics *do not* rely on global shape when classifying objects, contrary to the fundamental conclusion from psychological research that humans largely rely on shape when identifying objects [12]. DNNs largely classify objects based on *texture* and *local shape features* [188].

Prior work [53] showed that ML architectures can learn shape-based representations when trained on *Stylized ImageNet*, a version of ImageNet where images have their visual appearances transformed by applying artistic styles. ML architectures trained on the Stylized ImageNet have better object detection performance and robustness towards a wide range of image distortions. To the best of our knowledge, existing deep-feature-based similarity metrics have not examined the

³³... such as the CORnet-S model [102], one of the best models of human vision

performance of deep features extracted from networks trained on stylized images. We investigate whether the robust performance of ML architectures trained on Stylized ImageNet can transfer to the domain of information visualization, where humans tend to make judgments based on shape by extending the existing implementation of deep-feature-based similarity metrics, with details in .

4.3 RESEARCH GOALS AND EXPERIMENT OVERVIEW

Our ultimate goal is to assess the benefits and limitations of deep-feature-based perceptual similarity metrics for information visualization. For this goal, we conceptually replicated two experiments:

1. Veras and Collins [179] (using data from Pandey et al. [129])
2. Demiralp et al. [34]

Both studies provide high-quality, crowd-sourced human similarity judgments gathered on stimuli with varying visualization complexity. Together, these studies cover both holistic similarity judgments of data visualizations and fine-grained perceptual comparisons of visual encodings.

Our replication experiments follow a consistent procedure across both studies: assign pairwise distances between stimuli from the original studies using deep-feature-based similarity metrics, analyze the results using the same procedures as the original papers, and compare our results to the analyzed human judgment results in the original studies (Fig. 18). Due to the inability of deep-feature-based similarity metrics (specifically AlexNet) to process images with dimensions smaller than 63×63 , we generated new stimuli of size 224×224 when replicating Demiralp et al. [34] in Section 4.6.

4.4 IMPLEMENTING DEEP-FEATURE-BASED PERCEPTUAL SIMILARITY METRICS

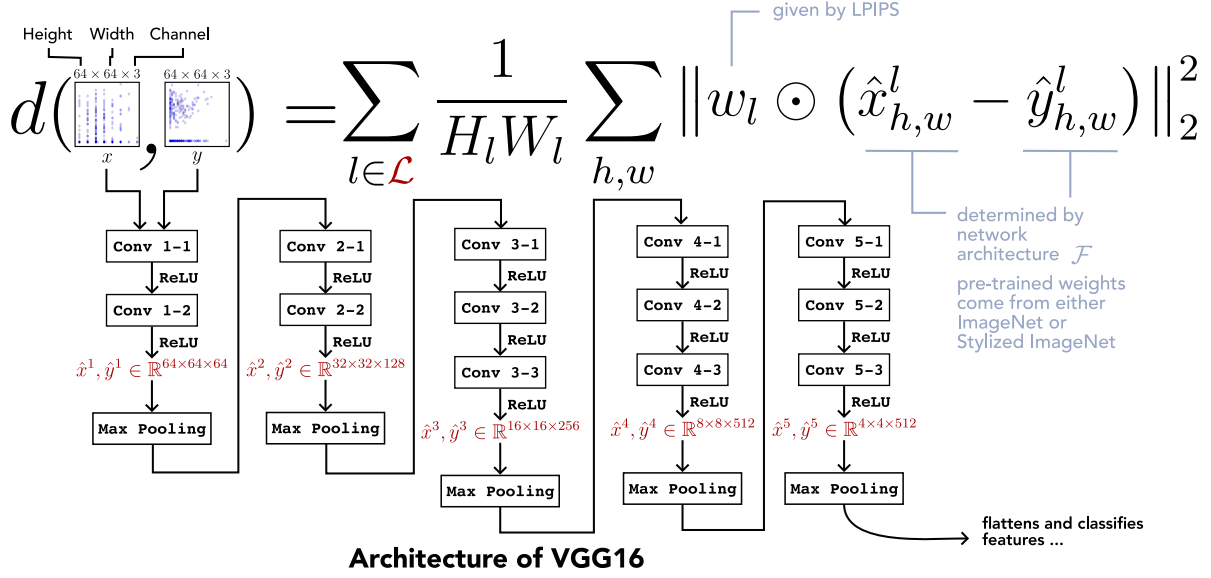


Figure 19: Diagram showing how the deep-feature-based perceptual similarity metric calculates “perceptual distance”, using VGG16 as the DL network $\mathcal{F}$.

We implement deep-feature-based perceptual similarity metrics that use ML architectures and three pre-trained weights. Our implementation primarily follows the implementations of Zhang et al. [207] and Kumar et al. [103]. We focus our attention on supervised models trained on the ImageNet-1k dataset [36], which contains 1,281,167 training images, 50,000 validation images, and 100,000 test images.

Given two images $x, y \in \mathbb{R}^{H \times W \times C}$, where H, W, C correspond to height width, and channel, and a network $\mathcal{F}$, the “perceptual” distance between x and y is the weighted sum of squared differences between feature activations of x and y across multiple layers and spatial positions. Formally, it is defined as

$$d(x, y) = \sum_{l \in \mathcal{L}} \frac{1}{H_l W_l} \sum_{h,w} \|w_l \odot (\hat{x}_{h,w}^l - \hat{y}_{h,w}^l)\|_2^2 \quad (3)$$

where $\mathcal{L}$ is the set of layers in network $\mathcal{F}$ from which features are extracted³⁴, $\hat{x}^l, \hat{y}^l \in \mathbb{R}^{H_l \times W_l \times C_l}$ are the unit-normalized³⁵ deep feature maps extracted by $\mathcal{F}$ at layer l , and vector $w_l \in \mathbb{R}^{C_l}$ is a channel-wise

³⁴the set of extraction layers $\mathcal{L}$ is architecture-specific; details of $\mathcal{L}$ in

³⁵... at the channel dimension

scaling vector for the difference between unit-normalized feature maps $\hat{x}^l$ and $\hat{y}^l$ at spatial location (h, w) .

Using out-of-the-box pre-trained weights from ImageNet or Stylized ImageNet corresponds to $w_l = \mathbf{1}_{C_l} \forall l \in \mathcal{L}$, where $\mathbf{1}_{C_l}$ is a vector of ones with length C_l , the number of channels at layer l . In other words, we use the extracted deep features *without any scaling or linear calibration*.

Using LPIPS (version 0.1)³⁶ corresponds to using ImageNet pre-trained weights *scaled by* parameters w_l learned from BAPPS [207], a dataset of 16K patches derived by applying exclusively low-level distortions to the MIT-Adobe 5k dataset [16] for training and the RAISE1k data [31] for validation. In other words, $w^l, l \in \mathcal{L}$ constitute a “perceptual calibration” of a few parameters in an existing feature space [207]. Zhang et al. [207] showed that LPIPS achieve better performance than traditional metrics (e.g., L_2 , SSIM [187]) and correlates well with human similarity judgments.

³⁶i.e., “lin” in Zhang et al. [207]

Our selection of DNN architectures is motivated by both theoretical considerations and empirical evidence from computer vision and prior work on training and predicting human similarity judgments [200]. We build upon Zhang et al.’s [207] finding that deep features from CNNs trained on natural images correlate strongly with human perceptual judgments. Our architectural choices span from SqueezeNet [81] and AlexNet [101] to more complex Residual Nets [65] and Efficient Nets [170], selected to systematically investigate how different levels of hierarchical feature extraction affect visualization similarity judgments. This range of architectures, covering ImageNet-1K classification accuracies from 56.522% (AlexNet) to 83.444% (EfficientNet B5), allows us to test whether improved natural image classification correlates with better visualization similarity assessment.

While newer architectures like Vision Transformers [39] exist, we prioritized CNNs for their architectural flexibility and relevance to our context: (1) CNNs allow processing of different input sizes, unlike Vision Transformers which require fixed input dimensions; (2) CNNs provide scale-invariant feature detection through their hierarchical structure; and (3) CNNs enable direct comparability with validated perceptual similarity metrics like LPIPS [207]. This flexibility is particularly important as our implementation builds on LPIPS, which has been rigorously validated against human perceptual judgments using the Berkeley-Adobe Perceptual Patch Similarity (BAPPS) dataset at 64×64 resolution. Importantly, our approach is architecture-agnostic and can be readily extended to any CNN architecture that provides hierarchical feature maps, enabling future comparisons to additional architectures.

Table 1: Network Architectures and Weights

Network Architecture	Weight
AlexNet [101]	ImageNet, Stylized ImageNet, LPIPS
VGG16 [161]	ImageNet, Stylized ImageNet, LPIPS
SqueezeNet [81]	ImageNet, LPIPS
ResNet18 [65]	ImageNet
ResNet50 [65]	ImageNet, Stylized ImageNet
EfficientNet B0 [170]	ImageNet
EfficientNet B5 [170]	ImageNet

³⁷<https://pytorch.org/vision/stable/models.html>

We use pytorch [132], torchvision weights,³⁷ and Zhang et al.’s implementation of LPIPS [207] at <https://github.com/richzhang/PerceptualSimilarity>. We use the following R packages for computing and presenting our results: reticulate [176], tidyverse [190], ggplot2 [191], aricode [21], MatrixCorrelation [82], ggtext [192], and ggpubr [88].

4.4.1 PRE-PROCESSING

We follow the same image pre-processing steps of Zhang et al. [207]³⁸. This approach differs from the standard $224 \times 224 \times 3$ resolution that ImageNet-1K is typically trained on. For completeness, we have also run the same experiments using the standard resolution with details provided in . While the overall trend remains similar, using the standard resolution leads to slightly worse performance than with a lower resolution of $64 \times 64 \times 3$. This finding suggests that the lower resolution may be more suitable for our specific task. By choosing a smaller size, we focus on the low-level aspects of perceptual similarity to mitigate the effect of “differing respects of similarity” [119] that may be influenced by high-level semantics [207]. Furthermore, the lower resolution yields significant improvements in computational efficiency, most notably reducing the computing time for VGG16 from approximately three hours to merely 30 minutes.³⁹

Prior research has shown that pre-processing images by transforming them into different color spaces yielded different image classification results [60]. We verify that all images in ImageNet-1K and BAPPS are in sRGB space, and we transform all input images into sRGB format before processing.

4.5 REPLICATING VERAS AND COLLINS [179]

4.5.1 OVERVIEW OF ORIGINAL PAPER

Veras and Collins [179] investigated the application of a well-established computer vision metric, the Multi-Scale Structural Similarity Index (MS-SSIM) [187], to assess the *discriminability* of a data visualization across a variety of datasets. In this context, discriminability refers to “the average perceptual distance between the corresponding visualizations” for a collection of datasets [179]. Their research proposed using

³⁸Namely, we convert the input stimuli into sRGB color space, resize them to $64 \times 64 \times 3$, and normalize them by the same mean and standard deviation coefficients used by Zhang et al. [207], which are the same mean and standard deviation for ImageNet-1K except adjusted for inputs ranging from $[-1, 1]$ instead of $[0, 1]$.

³⁹Experiments were conducted on a Dell XPS 15 (2019) with Intel Core i7-8750H @ 2.20GHz, 16GB RAM, NVIDIA GeForce GTX 1050 Ti with Max-Q Design, running Microsoft Windows 11 Pro.

perceptual similarity metrics to evaluate and rank competing visual encodings based on their discriminability, thereby informing visualization selection for specific data distributions.

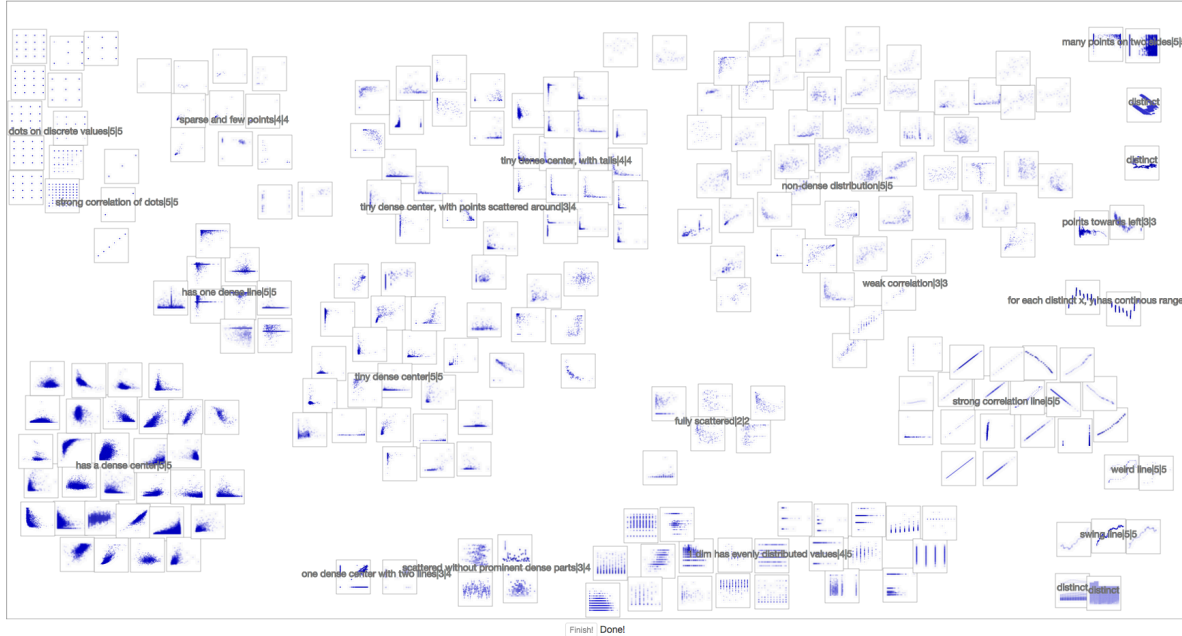


Figure 20: Screenshot obtained from the supplementary materials of Pandey et al. [129] showing Participant 1's final screen after completing the naming phase, with group descriptions and their corresponding easiness and confidence scores (separated by pipe characters).

Veras and Collins first used gradient descent to optimize MS-SSIM parameters using stimuli from a study of scatterplot similarity judgments by Pandey et al. [129]. With this tuned MS-SSIM, they calculated pairwise distances between scatterplots and derived grouping labels using hierarchical clustering. To validate their approach, they compared their group labels against the consensus group labels from human similarity judgments in Pandey et al.'s original study [129].

4.5.1.1 DATA, VISUAL STIMULI, TASK

Veras and Collins [179] based their research on data from Pandey et al.'s study [129], which investigated the subjective perception of similarity in scatterplots. Pandey et al. [129] conducted a study with 18 participants from scientific backgrounds, asking them to group similar scatterplot thumbnails and provide confidence and easiness ratings for their groupings (see Fig. 20). The participants were asked to group a total of 247

scatterplots. These human-generated groupings served as a benchmark for Veras and Collins [179] to validate the performance of their tuned MS-SSIM. Data from Pandey et al. is provided at <https://github.com/nyuvis/scatter-plot-similarity>.

4.5.1.2 SIMILARITY MEASURE

Given that different participants might group the same pair of scatterplots into two different groups, Pandey et al. [129] calculated the *consensus distance* between plots i and j as follows:

$$d_{i,j} = \frac{1}{N} \sum_{k=1}^N \left(1 - \frac{c_{i,j}}{\min(c_i, c_j)} \right)_k$$

where N is the number of participants, $c_{i,j}$ is the number of clusters that contain both plots i and j , and c_i and c_j are the number of clusters that contain the plots i and j respectively. The authors allowed participants to assign the same plot to multiple groups. These consensus distances formed a *consensus perceptual distance matrix* which was later clustered using hierarchical clustering to form groupings.

Veras and Collins [179] conceptually replicated Pandey et al.'s experiment [129] by using an alternative similarity measure to approximate the perceptual distance between scatterplots. They used the Multiscale-Structural Similarity Index (MS-SSIM) [187], an extension of the Structural Similarity Index (SSIM) [186] where the contrast and structural similarities are computed at K image scales. For image pair X, Y , MS-SSIM is defined as

$$\text{MS-SSIM}(X, Y) = l(x, y)^\alpha \prod_{i=1}^K c(x_i, y_i)^{\beta_i} s(x_i, y_i)^{\gamma_i}$$

where $l(\cdot)$ is the luminance similarity function, $c(\cdot)$ is the contrast similarity function, and $s(\cdot)$ is the structural similarity function. Veras and Collins set $\alpha = 1, \beta_i = \gamma_i = w_i$ for $i \in [K]$, where $K = 5$ is the number

of scales. The weights w_i can be interpreted as the relative importance of each image at scale i in determining similarity. Veras and Collins [179] employed an iterative process to adjust the scale weights $w_1, w_2, \dots, w_5$ in order to minimize the discrepancy between the similarity scores calculated by MS-SSIM and a set of empirical human judgments on scatterplots.

4.5.1.3 PERFORMANCE EVALUATION

Veras and Collins used pairwise distances calculated by fine-tuned MS-SSIM to construct clustering labels for the scatterplots and compared these clustering labels against empirical judgments using four established *cluster quality measures*⁴⁰ — adjusted mutual information (AMI), normalized mutual information (NMI), Rand index (RI), and adjusted Rand index (ARI). Based on information theory, NMI and AMI quantify the shared information between two clusterings, with NMI scaling mutual information to $[0, 1]$, while AMI correcting for chance. They are robust to differences in cluster numbers and sizes. RI measures the percentage of correct decisions made by the clustering algorithm, considering all possible pairs of points, while ARI adjusts RI for chance, providing a more reliable comparison between clusterings. All measures except RI range from 0 (random clustering) to 1 (perfect clustering relative to the ground truth).

⁴⁰Cluster quality measures are traditionally used to qualify the agreement between two independent label assignments on the same dataset.

4.5.2 IMPLEMENTATION

We extend Veras and Collins’ approach [179] by replacing MS-SSIM with deep-feature-based similarity metrics. One key difference between our approach and theirs is that while we exclusively rely on pre-trained ImageNet and Stylized ImageNet weights, Veras and Collins learned scaling weights for MS-SSIM using the scatterplots specific to their

study. As such, their learned metric might not transfer well to different visualization types or domains.

For each ML network (Table 1), we calculate $\binom{247}{2} = 30,381$ pairwise distances between all scatterplots and turn these pairwise distances into 247×247 distance matrices. For a baseline comparison, we also calculate pairwise distances using Mean Squared Error (MSE), a popular pixel-based metric. We then apply hierarchical clustering with Ward’s agglomeration strategy to compute the clusters for each distance matrix, using even-height tree cuts to yield 20 clusters — matching the number used by Pandey et al. [129] To evaluate performance, we calculate clustering quality measures between our labels and the labels obtained by Pandey et al. [129] We also compare our results to the results obtained by Veras and Collins [179]⁴¹ and the baseline MSE results.⁴²

⁴¹... represented by ♦ in Figure 21

⁴²... represented by × in Figure 21

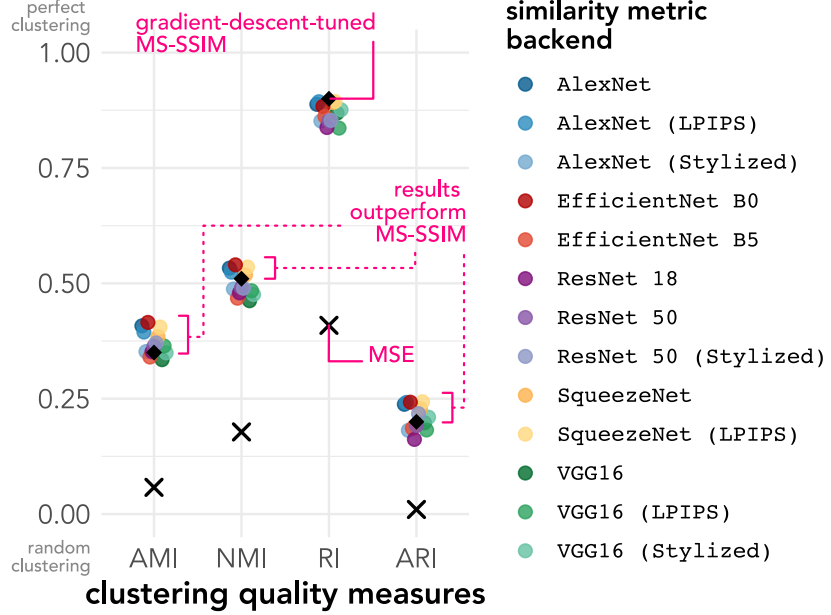
4.5.3 OUR RESULTS

We find that our top-performing architectures (● EfficientNet B0, ● ● AlexNet, and ● ● SqueezeNet) achieve comparable (if not superior) results to those achieved by gradient-descent-tuned MS-SSIM (♦), and far superior results to those achieved by MSE, the point-wise pixel difference metric. This outcome demonstrates robust performance across different CNN architectures and is particularly noteworthy as it relies purely on transfer learning with no training on the existing set of scatterplots.

Comparing between architectures, we find that ● SqueezeNet (LPIPS) generally outperforms all the other networks.⁴³ Except for RI, the best-performing ML networks, ● EfficientNet B0, ● AlexNet, and ● SqueezeNet (LPIPS) outperform the gradient-descent-tuned MS-SSIM results (♦) obtained in Veras and Collins [179].

⁴³ ● SqueezeNet (LPIPS) appear most frequently in the top-3 performing similarity metric backends for all clustering quality measures.

Figure 21: Black diamonds represent Veras and Collins’ [179] results using gradient-descent-tuned MS-SSIM. Crosses represent results using Mean Squared Error (MSE). Points are offset for clarity.



Put differently, the clustering labels using only ImageNet pre-trained weights match human labels better than Veras and Collins’ approach, which used parameters optimized for these scatterplots. Contrary to initial expectations, we find that incorporating stylized ImageNet weights does not consistently lead to performance improvements. For details, see the full set of clustered scatterplots, their labels (both from Pandey et al. [129] and from using ● SqueezeNet (LPIPS), and the table of all clustering performance measures (Table 4) in Appendix A.2.4.

4.5.4 ABLATION STUDY

To systematically analyze the factors contributing to the effectiveness of deep-feature-based similarity metrics, we conducted a series of ablation studies. The studies examine (1) the relative importance of network architecture versus learned weights, (2) the impact of different feature extraction layers on clustering quality, and (3) the role of supervised learning and dataset complexity in feature learning, comparing ImageNet-1K against simpler datasets.

4.5.4.1 ROBUSTNESS CHECK — RANDOM WEIGHTS

We initialize the same architectures with *random weights* to test whether our results stem from ImageNet pre-trained weights and assess if the resulting distance matrices and clustering labels still match empirical judgments. We repeat the process ten times and calculate the mean and 95% non-parametric bootstrap confidence interval (CI) for each architecture (Fig. 22).

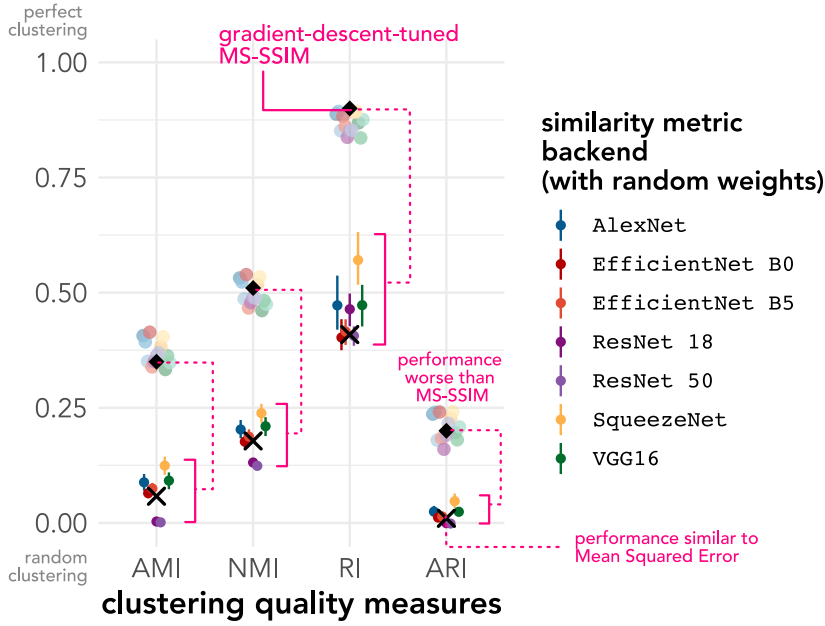


Figure 22: Black diamonds represent Veras and Collins’ [179] results using gradient-descent-tuned MS-SSIM. Crosses represent results using MSE. Dots show means of 10 trials. Lines indicate 95% confidence intervals (CIs) from non-parametric bootstrap. Points and CIs are slightly offset for clarity.

We find that when using randomly initialized weights, all network architectures consistently *under-perform* the results of the same networks that use ImageNet pre-trained weights (Fig. 21). In fact, the performance across network architectures is clustered around the performance of MSE, the pixel-based metric. These findings suggest that the clustering performance of deep-feature-based similarity metrics is largely attributable to weights trained on ImageNet.

4.5.4.2 IMPACT OF EACH FEATURE EXTRACTION LAYER

To identify which feature extraction layers contributed most to the clustering quality measures, we analyze the “perceptual distance” extracted

from *each individual layer* $l \in \mathcal{L}$, rather than summing across layers as in Equation 3. We also examine the effect of each layer by computing distances using *all layers except layer* $l \in \mathcal{L}$. We focus our analysis on the top-performing network architectures for each clustering quality measure.

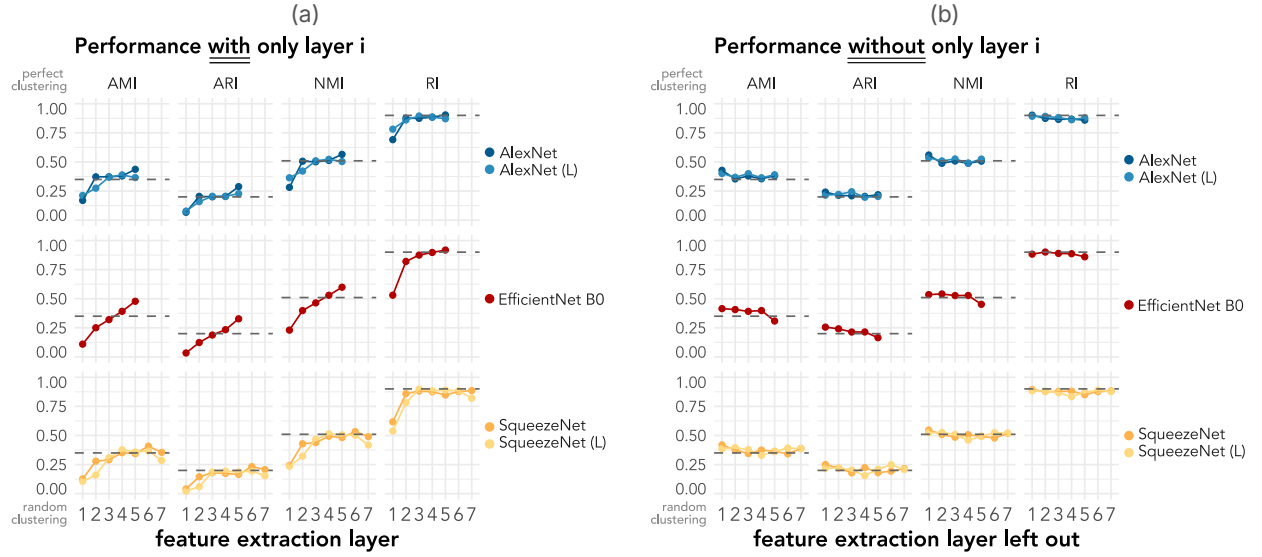


Figure 23: Dashed lines (- - -) represent Veras and Collins' results using gradient-descent-tuned MS-SSIM. (a) Clustering quality measures for AlexNet, AlexNet (LPIPS), EfficientNet B0, SqueezeNet, and SqueezeNet (LPIPS) calculated using perceptual distance at each layer $l \in \mathcal{L}$. (b) Clustering quality measures for AlexNet, AlexNet (LPIPS), EfficientNet B0, SqueezeNet, and SqueezeNet (LPIPS) calculated using all layers except layer $l \in \mathcal{L}$.

Our analysis shows that across all examined neural network backbones, performance around layers 3 or 4 reaches performance comparable to those reported by Veras and Collins (Fig. 23a). The best per-layer performance is usually the last or second-to-last layer.

From Figure 23b, the performance appears relatively stable, with no specific layer significantly decreasing or increasing performance. The LPIPS version of AlexNet and SqueezeNet (i.e., ImageNet pre-trained weights + linear calibration, AlexNet (L) and SqueezeNet (L)) in general does not improve performance on top of the existing performance of AlexNet and SqueezeNet. These findings suggest that: (1) the linear calibration added by LPIPS provides limited value for scatterplot similarity; (2) while no single layer appears to dominate performance, excluding the first layer slightly improves performance; and (3) using

only the last layer might be sufficient and could offer computational efficiency gains. These results align with prior observations [37,51] and show that later layers encode shape and structure information that better matches human similarity judgments.

4.5.4.3 IMPACT OF TRAINING OBJECTIVE AND TRAINING

DATASET

We compared models pre-trained using different frameworks: supervised learning (ImageNet-1K and CIFAR-10 [100]) and self-supervised learning (SimCLR on STL-10 [26]). Pre-trained weights were sourced from established repositories (see).

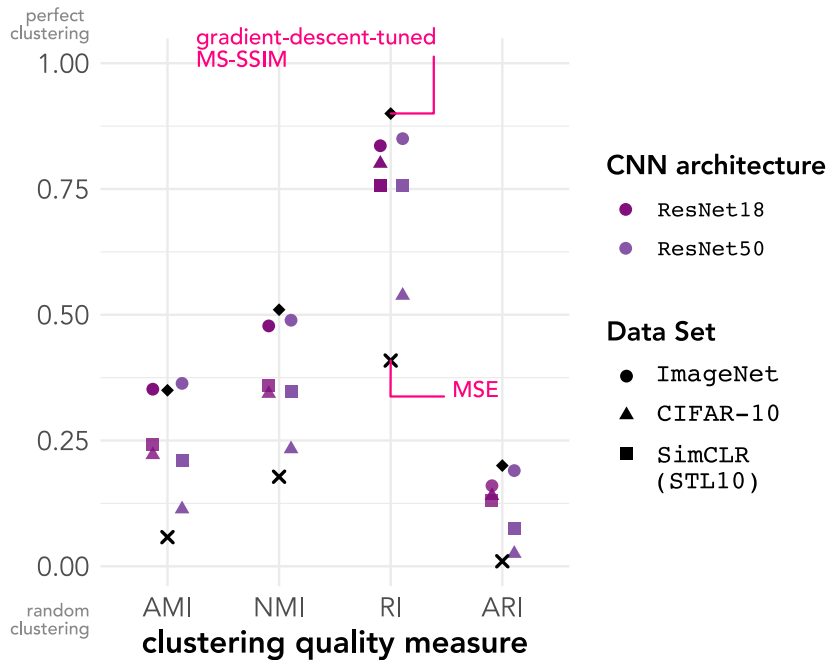


Figure 24: ● Resnet18 and ● ResNet50 performance. Black diamonds represent Veras and Collins' [179] results using gradient-descent-tuned MS-SSIM. Crosses represent results using MSE. Points are dodged for clarity.

Figure 24 reveals a clear performance hierarchy: ImageNet-trained models performed best across all clustering metrics (● ResNet50 slightly outperforming ● ResNet18), followed by ▲ CIFAR-10-trained models, while SimCLR achieved similar results to ▲ CIFAR-10 but underperformed compared to ImageNet. This gap could be due to differences in both the learning framework and dataset (STL-10 vs. ImageNet-1K).

We also observe an interesting pattern where ● ResNet50 trained on ▲ CIFAR-10 performs similarly to the MSE baseline. This poor performance likely stems from the mismatch between ● ResNet50's deep architecture — designed for complex, ImageNet-like datasets — and CIFAR-10's low-resolution (32×32) images. With limited input complexity propagating through 50 layers, the network might default to learning shallow, pixel-level features rather than developing rich semantic representations. This demonstrates how dataset complexity and the learning framework impact representation quality and, in turn, downstream performance.

4.5.5 DISCUSSION OF VERAS AND COLLINS

Our replication of Veras and Collins' study [179] yields compelling results: without any training or fine-tuning on the stimuli, the best-performing deep-feature-based similarity metrics outperform gradient-descent-tuned MS-SSIM, a traditional computer vision metric, on three out of four clustering quality measures by an average of 14.04% in aligning clustering labels to human labels of scatterplot. This is particularly noteworthy given that MS-SSIM's performance was specifically tuned on the scatterplots, while our deep learning models relied solely on pre-trained ImageNet weights. The ablation studies (§ 4.5.4) suggest that (1) perceptual losses extracted at later stages are generally better (§ 4.5.4.2) and (2) high-quality visual representations require both an appropriate learning framework and a sufficiently rich training dataset. The lower performance of CIFAR10-trained models (§ 4.5.4.3), despite using supervised learning, emphasizes that dataset characteristics (resolution, class diversity, sample complexity) may be as important as the choice of learning framework. These results provide evidence for effective transfer learning from large-scale natural image datasets to

data visualizations that utilize spatial encodings, suggesting that the features learned by neural networks on datasets such as ImageNet-1K may capture some fundamental aspects of visual perception that extend beyond the domain of natural images.

4.6 REPLICATING DEMIRALP ET AL. [34]

4.6.1 OVERVIEW OF ORIGINAL PAPER

Demiralp et al. [34] introduced the concept of *perceptual kernels*,^{fig. 25} which are distance matrices derived from aggregate perceptual judgments. Perceptual kernels encode perceptual differences in a reusable form that is directly applicable to visualization evaluation and automated design. For example, one can construct palettes that maximize perceptual discriminability, as shown in Figure 25.

The authors conducted two sets of crowd-sourced experiments, one to estimate the perceptual kernels *color*, *shape*, and *size*, and one to estimate perceptual kernels for the pairwise combinations (i.e., *color-shape*, *color-size*, *shape-size*). Each experiment used five different judgment types. They found that the triplet matching task, the task that asks participants to select between a pair of options that is most similar to the reference image, exhibits the least inter-subject variance, produces results that are less sensitive to subject count, and enables the most accurate prediction of bivariate kernels from univariate inputs [34].

4.6.1.1 DATA AND VISUAL STIMULI

In their first experiment, Demiralp et al. [34] used the default color and shape palettes from Tableau, each of which contained ten distinct values. For size, they used a palette consisting of ten circles with linearly increasing area. In the second experiment, Demiralp et al. [34] selected four values from each palette and performed a cross product, resulting in $4 \times 4 = 16$ distinct values for each pairwise combination of color,



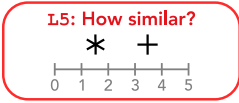
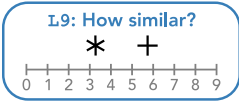
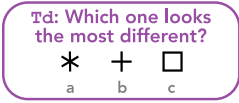
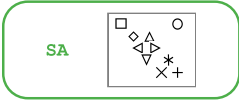
Figure 25: Palettes that maximize perceptual discriminability for color, shape, and size generated by Demiralp et al. [34], using the perceptual kernels from the triplet matching task.

shape, and size. Their source code, experiment interface, and data are at

<https://github.com/uwdata/perceptual-kernels>.

4.6.1.2 TASKS

Demiralp et al. elicited similarity judgments using five different tasks:

1.  **Pairwise rating on a 5-point scale (L5)** presents participants with a pair of stimuli and asks them to rate the similarity between these two stimuli on a 5-point Likert scale.⁴⁴
2.  **Pairwise rating on a 9-point scale (L9)** presents participants with a pair of stimuli and asks them to rate the similarity between these two stimuli on a 9-point Likert scale.⁴⁵
3.  **Triplet ranking with matching (Tm)** presents participants with three stimuli, one of which is the reference, and asks participants to select among the two remaining options which one is most similar to the reference.
4.  **Triplet ranking with discrimination (Td)** asks participants to select the *odd* stimuli out of a triplet of stimuli. This task is otherwise known as the *odd-one-out* task.
5.  **Spatial arrangement (Sa)** presents participants with all stimuli and asks them to arrange them on a 2D plane.

A total of 6 visual variables $\times$ 5 judgment types = 30 jobs were run on Amazon Mechanical Turk, with each job completed by 20 Turkers for 600 distinct subjects [34].

4.6.2 IMPLEMENTATION

We replicate parts of Demiralp et al.'s [34] experiments using deep-feature-based similarity metrics (Table 1) to generate 10×10 and 16×16 distance matrices. For a baseline comparison, we also calculate pairwise distances using Mean Squared Error (MSE), a popular pixel-based

⁴⁴These are actually 6-point Likert items (including 0), but we follow the naming scheme from the original work.

⁴⁵Likewise, these are 10-point Likert items (including 0).

metric. Because LAB color space is “perceptually uniform” [22], meaning numerical distances between colors match human perception of those differences, we compute MSE in LAB rather than sRGB space for visual channels that involve color. Following Demiralp et al.’s approach, we normalize each matrix to span the range $[0, 1]$.

4.6.2.1 DATA AND VISUAL STIMULI

Demiralp et al.’s experiments used stimuli of size 36×36 , which is too small for certain deep-feature-based similarity metrics to operate on. Since these stimuli consist of elementary visual channels (single colors, shapes, and sizes) rather than complex data visualizations, we do not expect human perceptual judgments to be affected by increasing the stimulus size. Therefore, we generated 10 (16 for *size-color*) PNG images in RGB color space of size $224 \times 224 \times 3$ for each visual channel.

4.6.2.2 PERFORMANCE EVALUATION

Demiralp et al. [34] compared the degree of compatibility between the five judgment tasks by calculating **Spearman’s rank correlation coefficient** on pairwise distances for the same visual channel. Spearman’s rank correlation coefficient, otherwise known as Spearman’s ρ , is defined as the Pearson correlation coefficient between the variables’ ranks and assesses how well the relationship between two variables can be described using a monotonic function. Spearman correlation ranges between -1 and 1 , with 1 indicating perfect rank correlation and -1 indicating perfectly opposite ranks. This metric has also been used to evaluate the performance of computer vision similarity metrics, such as DISTS [37], on Image Quality Assessment datasets.

4.6.3 RESULTS

For each of the four visual channels — *color*, *size*, *shape*, and *size-color* — we analyze the best-performing⁴⁶ DL network. For comparison, we

⁴⁶... which is the one that achieves the highest average Spearman rank correlation coefficient when compared to Demiralp et al.’s perceptual kernels across their five judgment tasks.

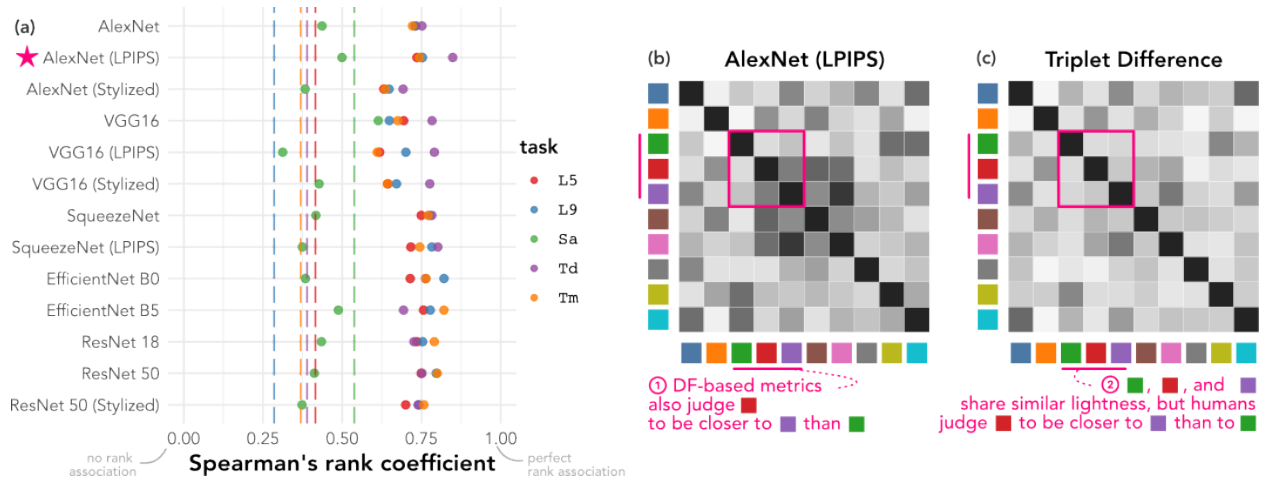
show the perceptual kernel most correlated with the distance matrix of the best-performing DL network. The complete set of distance matrices is available in the supplementary material.

4.6.3.1 COLOR

AlexNet (LPIPS) outperforms other DL networks. Distance matrices from deep-feature-based similarity metrics show the highest rank correlation with the perceptual kernel from **Td** and the lowest with **Sa** (Fig. 26 (a)).

AlexNet (LPIPS) demonstrates good overall performance in measuring relative color distances, but fails to replicate certain human perceptual patterns. For example, humans judge yellow to be more similar to green than teal, but AlexNet (LPIPS) “sees” yellow and teal as about equidistant from green.

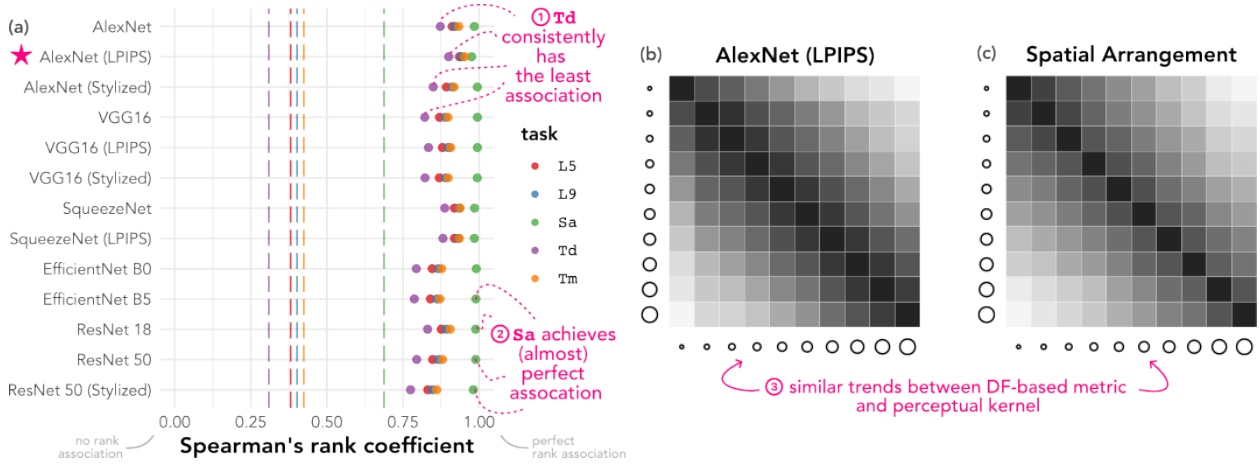
Figure 26: (a) Spearman's rank correlation between distance matrices and Demiralp et al.'s [34] perceptual kernels (dashed: MSE-based distance matrices), (b) AlexNet (LPIPS) distance matrix, (c) Demiralp et al.'s **Td** perceptual kernel.



The baseline MSE performances demonstrate the highest rank correlation with **Sa**, which makes sense given that MSE, a pixel-level metric, is directly measuring color differences in a perceptually uniform color space. This also suggests that **Sa** might be a better task to capture the non-linear aspects of human color perception. None of the pre-trained networks outperform the MSE baseline in **Sa**.

4.6.3.2 SIZE

AlexNet (LPIPS) outperforms other DL networks. Distance matrices from deep-feature-based similarity metrics and MSE show the highest rank correlation with the perceptual kernel from **Sa** and lowest with **Td** (Fig. 27 (a)).



Most DL networks achieve almost perfect rank correlation with **Sa** (Fig. 27 (a)) while MSE achieves around 0.7, suggesting these metrics capture more fundamental aspects of human size perception than pixel-level differences.

4.6.3.3 SHAPE

VGG16 (LPIPS) outperforms other DL networks. Distance matrices from deep-feature-based similarity metrics show the highest rank correlation with the perceptual kernel from **Tm** and lowest with **Sa** (Fig. 28 (a)). Stylized ImageNet weights weakly improve the correlation for VGG16 and ResNet50 but not AlexNet.

Figure 27: (a) Spearman's rank correlation between distance matrices and Demiralp et al.'s [34] perceptual kernels (dashed: MSE-based distance matrices), (b) AlexNet (LPIPS) distance matrix, (c) Demiralp et al.'s **Sa** perceptual kernel.

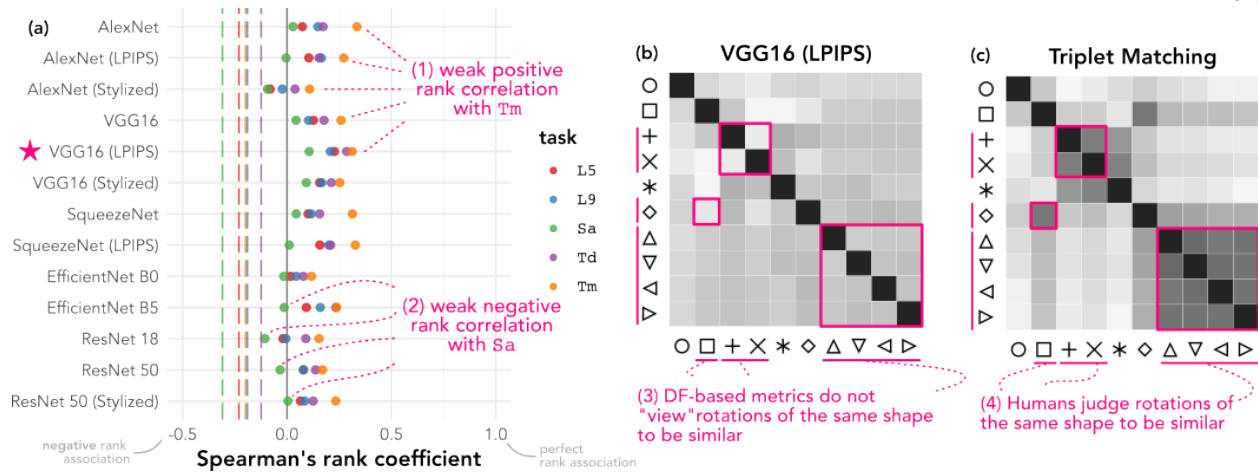
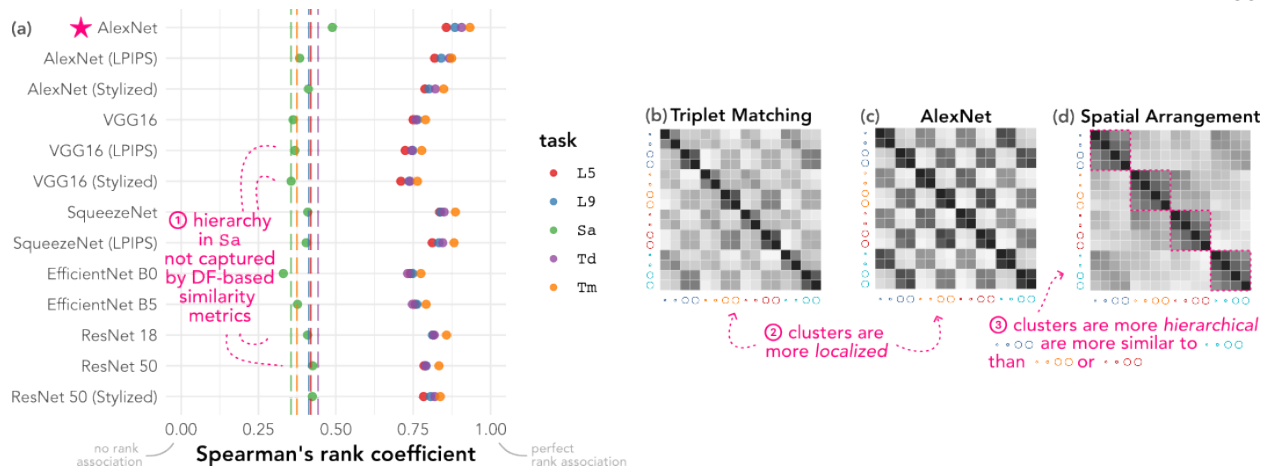


Figure 28: (a) Spearman's rank correlation between distance matrices and Demiralp et al.'s [34] perceptual kernels (dashed: MSE-based distance matrices), (b) VGG16 (LPIPS) distance matrix, (c) Demiralp et al.'s Tm perceptual kernel.

All MSE performances show a negative correlation, with Sa having the most negative correlation. Compared to MSE performances, the modest positive associations from CNNs show they are learning something, but there's still a large gap in capturing human shape perception. Specifically, all correlation coefficients are *weak* (i.e., $\in [-0.25, 0.5]$) compared to prior results for color (§ 4.6.3.1 Fig. 26) and size (§ 4.6.3.2 Fig. 27), where excluding Sa, rank correlations are around 0.75. Examining the distance matrix and perceptual kernel closely (Fig. 28 (b), (c)) reveals that, unlike humans, deep-feature-based perceptual similarity metrics do not view similarity as **rotation-invariant**. In other words, while humans may judge + and ×, □ and ◇ to be similar, as they are rotated variants of each other, VGG16 (LPIPS), the best-performing DL network, “sees” them as different. We speculate why this is the case in Section 4.6.5.

4.6.3.4 SIZE-COLOR

AlexNet outperforms other DL networks. Distance matrices from deep-feature-based similarity metrics show the highest rank correlation with the perceptual kernel from Tm and the lowest with Sa (Fig. 29 (a)).



Interestingly, perceptual distances obtained via *Sa* (Fig. 29 (d)) demonstrate a clear *hierarchy* — circles are most similar to those sharing the same color, followed by those with similar sizes, and then those with related colors (i.e., blue to teal, red to orange). In contrast, similarity judgments elicited through *Tm* (Fig. 29 (b)) reveal more *localized* similarity clusters, possibly due to the absence of *context* and its implied constraints when using *Tm* to elicit similarity judgments. This suggests a potential need to reconsider Demiralp et al.'s advice to favor triplet matching (*Tm*) judgments [34], especially when collecting visual variables that encode multiple visual channels simultaneously.

This hierarchy of “*color* before *size*” is not captured by either MSE or deep-feature-based similarity metrics (Fig. 29 (c)), and using ImageNet pre-trained weights in general does not improve correlation much against simple pixel-based MSE baseline.

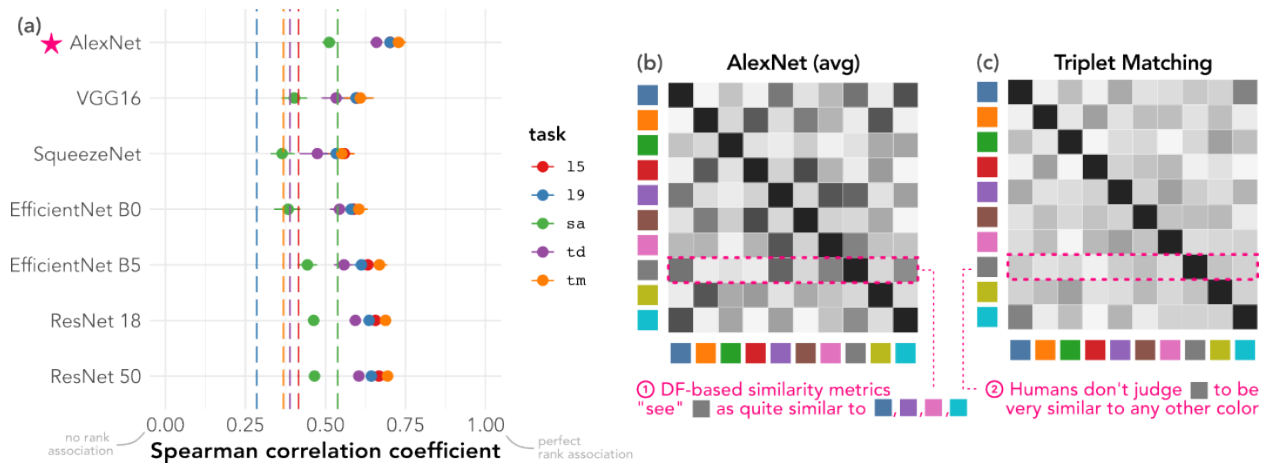
4.6.4 ROBUSTNESS CHECK – RANDOM WEIGHTS

Similar to Section 4.5.4.1, we initialize the same architectures with random weights to determine to what extent ImageNet pre-trained weights can explain our results. We repeat the process of calculating distance matrices and Spearman's rank correlation coefficients with perceptual kernels ten times and present the 95% non-parametric bootstrap CIs

Figure 29: (a) Spearman's rank correlation between distance matrices and Demiralp et al.'s [34] perceptual kernels (dashed: MSE-based distance matrices), (b) Demiralp et al.'s *Tm* perceptual kernel, (c) AlexNet distance matrix, (d) Demiralp et al.'s *Sa* perceptual kernel.

⁴⁷All distance matrices can be found in the supplementary material at <https://osf.io/dj2ms/>.

Figure 30: (a) Spearman's rank correlations between random-weight architectures' distance matrices and perceptual kernels (95% bootstrap CI; dashed: MSE-based distance matrices), (b) Random AlexNet's average distance matrix, (c) Demiralp et al.'s Tm perceptual kernel.



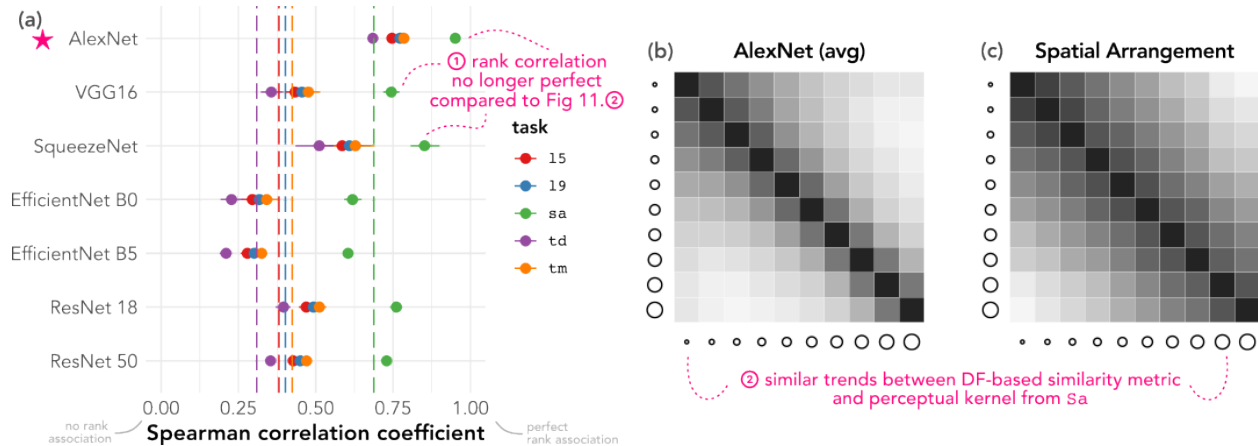
for Spearman's rank correlation coefficient. For each visual channel, we focus on the best-performing DL architecture and present the average distance matrix⁴⁷ that achieves the highest average Spearman correlation coefficient across all repetitions and all five judgment tasks. For direct comparison, we also present the perceptual kernel with which the average distance matrix has the highest Spearman correlation.

4.6.4.1 COLOR

Correlation results with random weights (Fig. 30 (a)) are only slightly worse than those with ImageNet pre-trained weights (Fig. 26 (a)) in Section 4.6.3.1, suggesting that existing deep-feature-based similarity metrics may not effectively capture the perception of color similarity.

4.6.4.2 SIZE

The correlation results for size (Fig. 31 (a)), except for AlexNet and SqueezeNet, are now around 75%, compared to perfect rank correlations for almost all networks (Fig. 27 (a)).



This suggests that deep-feature-based similarity metrics are learning something about size, but it is unclear to what extent this is attributable to ImageNet pre-trained weights or the functional form of deep-feature-based similarity metrics (Eq. 3) that relies on taking Euclidean differences between deep feature activations across spatial locations. For all tasks except **Sa**, using randomly initialized weights result in about similar rank correlation performance as using MSE (except for AlexNet and SqueezeNet).

4.6.4.3 SHAPE

Across all DL architectures, the performance is weakly negatively correlated to all perceptual kernels obtained via different tasks (Fig. 32 (a)) and is worse than the previous correlation results (Fig. 28 (a)) in Section 4.6.3.3.

Figure 31: (a) Spearman's rank correlations between random-weight architectures' distance matrices and perceptual kernels (95% bootstrap CI; dashed: MSE-based distance matrices), (b) Random AlexNet's average distance matrix, (c) Demiralp et al.'s **Sa** perceptual kernel.

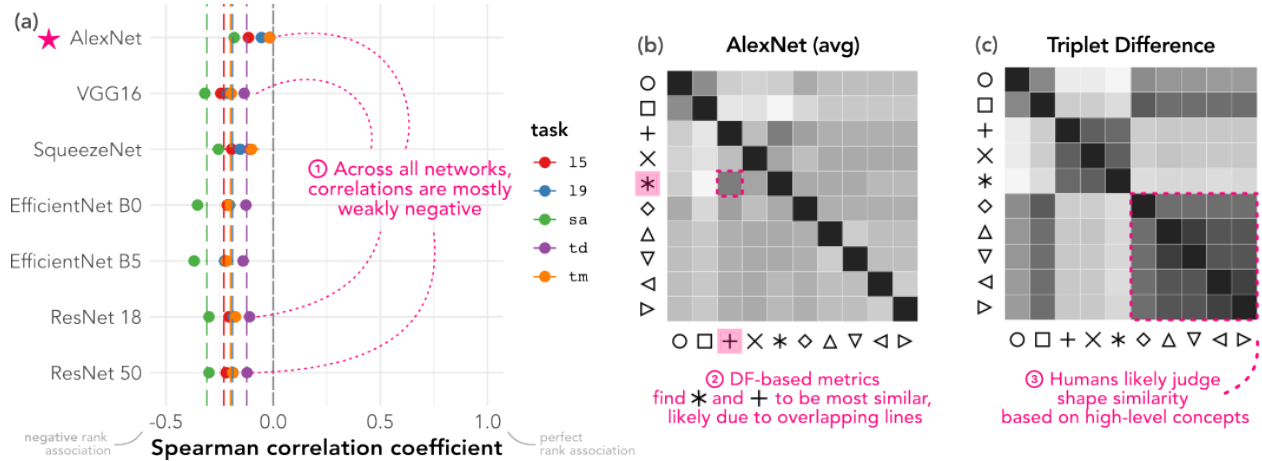
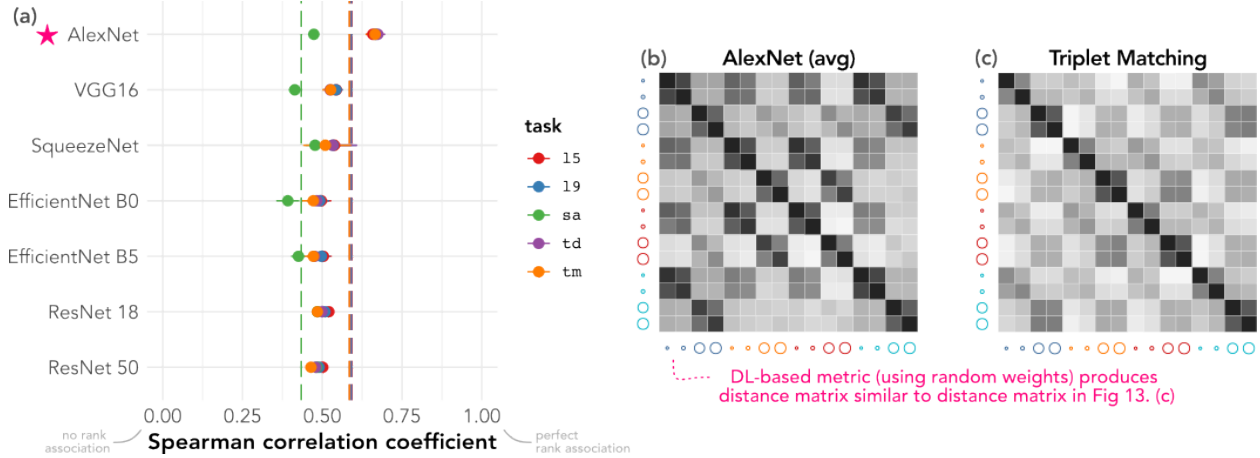


Figure 32: (a) Spearman's rank correlations between random-weight architectures' distance matrices and perceptual kernels (95% bootstrap CI; dashed: MSE-based distance matrices), (b) Random AlexNet's average distance matrix, (c) Demiralp et al.'s Td perceptual kernel.

The performance when using randomly initialized weights is very similar to that of using MSE. Given the existing functional form of deep-feature-based similarity metrics (Eq. 3) and its reliance on taking Euclidean differences between deep feature activations spatially, this result suggests that some transfer learning is happening in Figure 28, but not to the degree that it captures what humans perceive as shape similarity. We also suspect that glyph shape similarity judgments are not entirely perceptual but may also depend on context/concept — consider this triplet difference example: ($\times$, $\square$, $\circ$). One could choose $\circ$ as the odd one out, since the other shapes have angles and circles do not have angles; or choose $\times$ as the odd one out, since both $\square$ and $\circ$ can be drawn on a page without lifting the pen off the page.

4.6.4.4 SIZE-COLOR

Again, we observe similar trends — the rank correlation results (Fig. 33 (a)) are slightly worse compared to the rank correlation results (Fig. 29 (a)) in Section 4.6.3.4 when using ImageNet pre-trained weights. For all tasks, using randomly initialized weights result in rank correlations about the same as MSE.



Across the examined visual channels, the best-performing architecture, when using randomly initialized weight, is always AlexNet. We suspect that the simpler and shallower architecture and the larger filter size in the earlier layers (11×11 , 5×5 instead of 3×3) of AlexNet might “preserve” more of the raw visual information, which explains its close to 0.5 rank correlation with **Sa** for visual channels related to spatial information (*size* and *size-color*).

4.6.5 DISCUSSION OF DEMIRALP ET AL.

Although deep-feature-based similarity metrics achieve positive (Fig. 26, Fig. 29), if not perfect (Fig. 27), rank correlation for several visual variables, the comparison against baseline MSE performances and robustness check (§ 4.6.4) reveals that it is unclear whether such performance can be entirely attributed to pre-trained ImageNet or Stylized ImageNet weights.

We suspect the decent correlation in the robustness check stems from both the functional form of deep-feature-based similarity metrics (Eq. 3) and Demiralp et al.’s [34] choice of visual stimuli. Deep-feature-based similarity metrics, as we implemented, calculate *Euclidean differences between deep feature activations* and for simple visual stimuli without complex patterns or textures, the feature map difference pri-

Figure 33: (a) Spearman’s rank correlations between random-weight architectures’ distance matrices and perceptual kernels (95% bootstrap CI; dashed: MSE-based distance matrices), (b) Random AlexNet’s average distance matrix, (c) Demiralp et al.’s **Tm** perceptual kernel.

marily reflects how well the stimuli *spatially align/structurally correspond*. For example, with rotated shape stimuli (e.g., $\times$ and $+$), the feature map differences remain almost identical whether using pre-trained ImageNet or random weights, given the same DL architecture.

Across all five similarity judgment tasks (**L5**, **L9**, **Tm**, **Td**, and **Sa**), **Sa** shows the least correlation with deep-feature-based similarity metrics for all visual channels except *size*, and has the least average rank correlation with all other perceptual kernels of the same visual channel (Demiralp et al. [34]). We suspect this occurs because participants see the most visual stimuli simultaneously (all 10/16 of them), and *therefore face more global, hierarchical constraints when judging their similarities*. In contrast, pairwise rating tasks (**L5**, **L9**) and triplet judgment tasks (**Td**, **Tm**) only require focusing on the local relationships between 2–3 stimuli at a time. This provides one way to explain the **Sa** perceptual kernel for *size-color* (Fig. 29), where participants judge similarity first by color, then by size within each same-color group.

Our replication of Demiralp et al. [34]’s experiment demonstrates that deep-feature-based similarity metrics currently *do not approximate the perceptual distance between visual encodings* such as color and shape. Specifically, these metrics are *not rotation-invariant* (§ 4.6.3.3) and do not capture the *precedence effect* of certain visual channels over others in similarity judgments (§ 4.6.3.4). Finally, while we explored using Stylized ImageNet weights, we cannot conclude definitively whether this approach improves performance, suggesting the need for further research.

4.7 DISCUSSION

4.7.1 KEY FINDINGS

1. **Effectiveness for complex visualizations:** Our replication of Veras and Collins’ study [179] demonstrates that deep-feature-based similarity metrics can outperform traditional computer vision metrics (e.g., MS-SSIM) in aligning with human clustering judgments of scatterplot similarity. We highlight the fact that MS-SSIM parameters are optimized on the domain data (i.e., scatterplots) while deep-feature-based similarity metrics only rely on weights trained on *natural images* and are therefore completely domain-free. This implies that transfer learning from natural image domains to visualizations is feasible and that ImageNet pre-trained weights capture fundamental visual features that generalize across diverse visual domains. This could be good in practice by significantly reducing the effort required to develop and train models from scratch for each new set of domain-specific visualizations.
2. **Limitations for abstract visual encodings:** In replicating Demiralp et al.’s work [34], we find that deep-feature-based metrics struggle to capture human perceptual similarities for basic visual channels like color and glyph shape, but perform well when assessing size. We hypothesize that part of the reason for this poor performance is because judgments of color and glyph shape similarity are *not purely perceptual* that is, they also rely on high-level semantics and/or concepts. For example, different cultural associations with color hues [84] may lead people to judge the similarity of color between the same triplet differently. In terms of context/concept, see the example in Section 4.6.4.3.
3. **Architecture and weight sensitivity:** Our results show that the performance of deep-feature-based metrics does not vary significantly across different neural network architectures (Fig. 21) but does vary across different pre-trained weights of varying complexity (e.g., CIFAR-10 vs ImageNet-1K in Fig. 24). The impact of stylizing ImageNet is less clear.

In summary, our findings suggest a potential for deep-feature-based similarity metrics in tasks involving visualizations that exhibit texture

or pattern-like features (e.g., scatterplots) and highlight a need for caution when applying these metrics to primitive visualization glyphs or elements.

4.7.2 LIMITATION AND FUTURE WORK

While deep-feature-based similarity metrics offer a promising solution for approximating similarity perception in information visualization, future work should explore the **generalizability** of deep-feature-based similarity metrics, focusing on:

1. **Perceptual vs. conceptual similarity:** Our current approach focuses on low-level perceptual similarity and does not account for higher-level conceptual or semantic similarities [129] that may be important in visualization interpretation.
2. **Visualizations are multi-modal and relational:** Our study primarily focuses on static, visual elements. An interesting direction for future work is to investigate how these metrics can be extended to handle different types of visualizations and their encoded visual relations [45,62,92], diverse user groups, or multimodal visualizations that incorporate text, interactivity, or other non-visual elements.
3. **Benchmark development:** Deep-feature-based similarity metrics vary widely in their architectures [103,207], training/tuning datasets [37], feature extraction layers [135], and evaluation approaches [165]. Comprehensive benchmarks are needed to standardize the evaluation and comparison of these metrics across domains.
4. **Explore more DL architectures, learning frameworks, and training datasets:** While this work explores five architectures (trained for image classification), future work can investigate Vision Transformers (ViT) [178], alternative learning frameworks (e.g., self-supervised learning), and low-bit quantization. However, adapting ViT presents a fundamental challenge because the Transformer architecture relies on fixed-size sequences of patches and positional encodings. For example, the ViT-B/16 model processes 224×224 images into 196 patches (14×14), while our 64×64 resolution yields only 16 patches (4×4). This significant mismatch in sequence lengths makes it impractical to transfer pre-trained ViT weights to our deep-

feature-based similarity metric without developing novel positional encoding mappings — an algorithmic challenge beyond this paper’s scope.

5. **Replication scope and inherited design choices:** By replicating prior studies, our work inherits their insights and methodological constraints in comparing human and machine behavior. While we show deep-feature-based metrics can fit into existing frameworks (as shown in Fig. 18), we are limited by these studies’ choices in stimulus presentation, task design, and inference procedures. Future research should validate these metrics through new experimental designs that systematically vary how human similarity judgments are elicited, representations are inferred, and human-machine behaviors are compared.

4.7.3 SIMILARITY, CONSTRAINTS, AND INFERRED REPRESENTATION

Examining perceptual kernels from *size-color* (§ 4.6.3.4) reveals that different similarity judgment tasks likely impose *different constraints* on participants. These different constraints are then encoded in different outcome variables and translated into different inferred representations. Beyond traditional tasks such as triplet matching and triplet difference, there are also quadruplet matching [58], octet matching [150], and the lineup task [14]. Beyond the contextual constraints imposed by different similarity judgment tasks, human cognitive limitations also play a crucial role — just as Hosseinpour et al. [75] demonstrated that increasing the number of frames in a small multiples visualization leads to a linear decline in accuracy, the complexity of similarity judgments may also tax our limited cognitive processing capacity.

Moreover, different visualization tasks may require different similarity models. Existing research has predominantly modeled similarity via a *geometric model* [172]. Such models assume minimality, symmetry, and triangle inequality. However, violations of all these assumptions have

been empirically observed [174]. Further research is needed to understand how different judgment tasks affect behavioral outcomes, how task constraints influence representation inference, and how different similarity models perform across visualization contexts.

4.7.4 POTENTIAL APPLICATIONS AND BROADER IMPLICATIONS

While not yet a replacement for human testing, these metrics show promise for both pre- and post-experimental applications. For instance, they could help researchers narrow visualization design spaces before human studies by providing initial estimates of regions likely to be perceived as similar or dissimilar. They could also enable systematic comparison and evaluation of designs, such as complementing qualitative findings from human studies by quantifying how similar human-recreated charts are against the original charts they were shown [140].

This work contributes to the broader discussion about the role of computational models in understanding and automating aspects of visualization design and evaluation [35]. While our results show promise, they also highlight the complexity of human visual perception and the challenges in creating truly generalizable models. As the field progresses, interdisciplinary collaboration between visualization researchers, cognitive scientists, and machine learning experts will be crucial to developing more perceptually aligned computational models for visualization analysis and design.

5 TOWARDS A COMPOSITIONAL THEORY OF VISUALIZATION PERCEPTION

5.1 INTRODUCTION

The preceding chapters each model a specific aspect of visualization perception; this chapter asks whether there is a more general decomposition. It proposes *visual decoding operators* — composable perceptual primitives with estimable error profiles — and tests whether operators characterized on one set of charts and tasks can predict performance on structurally different ones.

Visualization research has a long tradition of *decomposition*: breaking the problem of understanding charts into smaller, nameable parts. Rankings order visual channels by perceptual effectiveness [23,115]. Task taxonomies classify what viewers accomplish with a chart [2,108]. Cognitive frameworks describe the stages of comprehension [98,136]. Each provides a useful organizational vocabulary, and together they have shaped design guidelines, recommendation systems, and teaching practices [121]. Nevertheless, the *units* that these decompositions produce share a common limitation: they do not capture enough internal structure to predict what will happen in situations not yet tested. This

This chapter is based on work submitted to IEEE VIS 2026, with Remco Chang, Eugene Wu, Alex Kale, and Matthew Kay.

All supplemental materials are available on OSF at <https://osf.io/prtfq/>.

implies that every new display, task, or viewing condition requires a new experiment, *an approach that does not scale*.

Why is generalization so difficult? We argue that the problem lies in the unit of analysis itself. Consider how existing decompositions carve the space. Rankings fix a task (e.g., judge a ratio) and vary the visual channel, producing an ordering over channels [23]. Task taxonomies classify what viewers accomplish with a chart (e.g., retrieve value, find extremum) but say nothing about which encoding supports each task or with what accuracy [2]. In both cases, the unit is bound to one side of the *visualization* $\times$ *task* pairing: channels are properties of the visualization; task categories are properties of the analysis goal. Findings obtained under one pairing do not transfer to the other dimension. Knowing that position outperforms angle for ratio judgments does not tell us how either channel performs for trend estimation. Knowing that a task is classified as “retrieve value” does not tell us how accurately a viewer retrieves it from a bar chart versus a heatmap.

What would a more useful unit of analysis look like? It would need to be *chart-agnostic*, appearing across different chart types. It would need to carry *quantitative error profiles in perceptual units*, so that predictions for untested conditions can be computed rather than read off ordinal comparisons that specify direction but not magnitude. And it would need to be *composable*: complex tasks expressed as sequences of operations whose error accumulates traceably, rather than evaluated as visual features where individual sources of error remain confounded.

In this chapter, I propose such a unit of analysis: the *visual decoding operator*. Operators are perceptual primitives, ranging from projecting from a point to an axis, to locating a curve’s peak, to judging the steepest slope. Each operator is modeled as a function with estimable bias and

variance, and a given operator⁴⁸ appears whether the chart is a Cumulative Distribution Function (CDF) or a scatterplot and whether the task is looking up a single value along a curve or estimating the mean of a cluster of points. Operators are parameterized in a perceptually uniform space (e.g. visual angle) so they generalize across display contexts. And because they compose, complex tasks can be built from simpler operations, each carrying a quantified error.

⁴⁸e.g., project from a known position to a curve

Three things must hold for this approach to deliver on the generalization problem. First, operators must be defined precisely enough that their parameters (bias and variance) can be estimated in isolation, without confounding from other operations in the same task. Second, the output of one operator must serve as valid input to the next, so that error propagation through a composed sequence is tractable rather than ad hoc. Third, the resulting predictions must actually account for empirical behavior on tasks the operators were not fit to.

Whether all three hold is an empirical question, and we provide an existence proof based on two stages that use structurally different visualization designs and tasks, each decomposable into separable operators. First, using Probability Density Function (PDF) and CDF charts, we design tasks that isolate individual operators and characterize their bias and variance through hierarchical modeling (§ 5.4). A surprising finding is that when a task admits multiple operators (e.g., median estimation on a PDF can use both peak detection and area bisection), a *sensor fusion* model that weights operators by their reliability outperforms a simple mixture model, suggesting that viewers integrate rather than select between available perceptual strategies. We then show generalizability by composing learned operators to predict performance on Moritz et al.'s [120] scatterplot mean-estimation task (§ 5.5), a task whose *chart*

type, dimensions, and analytic goal all differ from the conditions used to learn the operators. Operator composition captures observed response bias and variance *with no parameters fit to the observed response data*, effectively predicting out-of-sample performance on a new visualization×task pair based on the results of the first study. Five alternative prediction approaches fail in distinguishable ways.

The predictive and explanatory power of our visual decoding operators presents new opportunities for automated reasoning about visualization design. For example, using quantified operator-specific errors such as those we report in this study as a proxy for the likely cognitive effort required to decode a visualization, a new class of recommendation systems could optimize for designs that match a user’s tolerance for cognitive load. We discuss implications for reshaping design patterns in tools for data interaction as well as the new body of empirical work required to estimate a library of decoding operators.

5.2 BACKGROUND AND RELATED WORK

5.2.1 THEORIES AND MODELS OF VISUALIZATION COMPREHENSION

Rankings of visual channels [7,23,64,94] decompose the design space into an ordered list of encodings, providing easy-to-access heuristics for visualization recommendation. Task taxonomies such as Amar et al. [2] decompose analytic activity into categories⁴⁹ that describe *what* a viewer can accomplish with a chart. Cognitive frameworks decompose the comprehension process itself into named stages [71,126,136,160]. Each of these traditions contributes a useful organizational vocabulary for describing visualization comprehension.

However, the *units* these decompositions produce share a common property: they are *categorical or ordinal*. Rankings reduce experiment

⁴⁹e.g., Retrieve Value, Find Extremum, Correlate

outcomes to a single scalar, collapsing over variance in responses and masking differences across individuals and contexts [33]. Task taxonomies classify what a viewer does but say nothing about how accurately or with what bias they do it. Cognitive stage models name the processes involved but remain specified at a conceptual level, without operationalizing them into units whose parameters can be estimated from data.

A separate line of work has pursued more operational forms of decomposition. Gillan and Lewis's Mixed-Arithmetic Perceptual (MA-P) model [56] predicts that graph comprehension time scales linearly in the number of cognitive components used. Peebles and Cheng [133,134] implemented process models of graph reading in the ACT-R/PM cognitive architecture, simulating the sequence of perceptual, cognitive, and motor operations a viewer executes to predict both response latency and eye fixation patterns. These computational models produce testable quantitative predictions, but their target is response time, not the perceptual output itself.⁵⁰ Wu and Chang [195] decompose the visualization design process into functional transformations and visual encodings, showing that design tasks can be described as compositions of operations. Nevertheless, their decomposition applies to the encoding stage, not the subsequent interpretation by a human viewer. More recently, Huang and Nobre [77] use large language models to decompose visualization tasks into sequences of analytic subtasks drawn from established task taxonomies [2], validated against expert reasoning patterns. Their subtask units, such as "Retrieve Value" or "Filter", describe what the viewer needs to accomplish at each step, but do not characterize the perceptual operations that execute each step or the error signatures those operations produce.

⁵⁰e.g., the estimate a viewer produces and the errors that arise from the visual encoding

A few empirical studies have moved toward quantitative models of specific perceptual operations: Talbot et al. [168] developed an empirical model of slope ratio comparison that captures aspect-ratio-dependent bias, characterizing a specific visual judgment with estimable parameters rather than an ordinal ranking. What remains missing is a framework that treats such characterized operations as composable units whose error propagates traceably through multi-step tasks.

5.2.2 ERROR ATTRIBUTION IN VISUALIZATION EXPERIMENTS

Observed error in a visualization experiment is rarely attributable to a single source. A participant's response reflects not only their perceptual judgment but also the elicitation method used to collect it, the study design that framed the task, and individual differences in visualization literacy, domain knowledge, and task interpretation. Each of these stages can introduce error, and without a way to separate these sources, the perceptual signal of interest remains confounded with measurement artifacts.

The elicitation method is one well-documented source of error. Since Heer and Bostock [69] demonstrated that graphical perception experiments could be successfully conducted online, crowdsourced studies have become the norm, and researchers have developed a range of *interactive response formats*, from clicking directly on the perceived centroid of the stimuli [74] to dragging response probes [197]. The choice of elicitation format is not neutral: numeric entry produces rounding behaviors [23,73], and sketching may be confounded with drawing ability and task interpretation alongside perceptual judgments [123,141].

Other sources of error operate upstream of the response. Nobre et al. [123] found that participants who misread visualizations failed at distinct cognitive stages, such as misreading channels, confusing

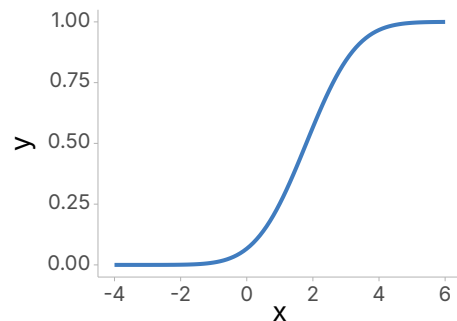
labels with values, or failing to translate questions into visual queries. Sarma et al. [155] argue complementarily that aspects of study design frequently treated as incidental, such as onboarding procedures, tutorials, and training phases, can interact with individual-level factors such as visualization literacy in ways that are difficult to separate from the perceptual signal itself.

To build componential models that predict the perceptual output of individual visual operations, we need an approach that both decomposes tasks into separable operators and explicitly controls for non-perceptual error sources. Section 5.3 introduces the decomposition framework, and Section 5.4.1 details how our experiment design addresses the sources of error identified above.

5.3 A MODEL FOR VISUAL OPERATORS

Original Question

Given the following cumulative distribution plot, what is the median?



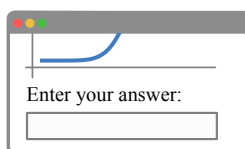
Domain knowledge needed to translate original question into visual task

Median is the value of x when $y = 0.5$.

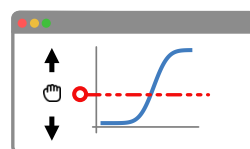
Translation of original question into a specific visual task

When $y = 0.5$ intersects the curve, what is the value of x ?

Elicitation Method 1



Elicitation Method 2



Elicitation Method 3



Figure 34: An illustrative example of different ways tasks can be set up to elicit a response. We use elicitation method 3 (details in Section 5.4.1).

Consider this scenario: you ask a participant to identify the median of a cumulative distribution function (CDF).⁵¹ A typical visualization study might elicit estimates of this value by showing participants different

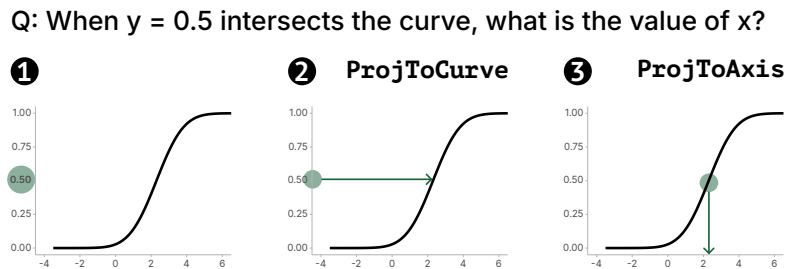
⁵¹ The mathematical definition is $y = \Pr(X \leq x)$.

CDFs and asking them to report the medians using different elicitation methods (Fig. 34). Researchers then calculate the error and variance in participants’ ability to identify medians from CDFs. As a starting point, assume the participant’s response is drawn from some distribution:

$$\text{response} \sim \mathcal{N}(\text{true value} + \text{bias}, \text{standard deviation}) \quad (4)$$

But this treats the task as *monolithic* — it does not distinguish where error enters. Instead, our task instructions (§ 5.4.1.1) decompose CDF median identification into three distinct steps that follow from the task definition itself (Fig. 35): ① locate $y = 0.5$ on the Y-Axis, ② perform a horizontal visual scan, projecting horizontally from $y = 0.5$ to where it intersects the CDF curve (**ProjToCurve**), then ③ project vertically from the curve intersection to the X-Axis (**ProjToAxis_x**). Given that the label “0.5” is aligned and next to its position on the y axis, ① is trivial. The core challenge then becomes measuring the error in the horizontal projection (②) and vertical projection (③).

Figure 35: Diagram explaining the task decomposition process using median identification on CDF as a concrete example.



Before we can specify these operators formally, we need to establish *what space we measure error in*. To create a model that generalizes across across viewing context, from desktop screens to museum installations, we need to measure error in a standardized, *viewing-context-independent* space. We choose to measure error in *visual angle space* which is the angle a distance subtends at the eye. Visual angles are typically used in vision science research [117] and allow us to approximate the perception of physical distances regardless of viewing conditions. Values are

transformed from data space to physical space, and then to visual angle space ⁵² through

⁵² measured in degrees

$$\text{visual angle} = 2 \times \tan^{-1} \left(\frac{\text{physical size}}{2 \times \text{distance to screen}} \right) \times \frac{180}{\pi} \quad (5)$$

To calculate the visual angle of on-screen stimuli, which requires knowing the viewing distance (Eq. 5), we implemented the “virtual chinrest” method proposed by Li et al. [109]. This browser-based method measures a participant’s viewing distance by two calibration tasks: 1) a *blind spot* task, where participants find their physiological blind spot to establish their distance from the screen, and 2) an *object scaling* task, where they match a digital rectangle to a physical object of known size (like a credit card) to find the monitor’s pixels-per-centimeter ratio. We reimplemented these tasks on the reVISit platform [30,38], following its open-source version.⁵³ In the paper we will denote visual angle transformations as:

⁵³ https://github.com/QishengLi/virtual_chinrest

$$\mathbf{va}(d; \text{context})$$

Where d is a value in data space and *context* includes all participant- and context-specific parameters needed to transform a value from data space to visual angle, including distance to screen, pixels-per-centimeter, and data units-per-pixel.⁵⁴

⁵⁴ For simplicity we typically omit *context* and write $\mathbf{va}_x(\cdot)$ or $\mathbf{va}_y(\cdot)$ to refer to the context-specific visual angle transformation along the x or y axes.

With visual angle space defined, we denote the true value in this space as θ and rewrite the two core operators (②, ③) in the median-finding decomposition as:

$$\begin{aligned} & \text{true value in visual angle space} \\ & \downarrow \\ \text{ProjToCurve}(\theta) & \sim \mathcal{N}(\theta + \beta_{\text{curve}}, \sigma_{\text{curve}}) \\ & \uparrow \quad \uparrow \\ & \text{bias for ProjToCurve} \quad \text{standard deviation for ProjToCurve} \\ \text{ProjToAxis}_x(\theta) & \sim \mathcal{N}(\theta + \beta_{\text{axis}}, \sigma_{\text{axis}}) \end{aligned}$$

Decomposing the task into visual operators become

$$\text{response} = \mathbf{va}_Y^{-1}[\text{ProjToAxis}_X(\text{ProjToCurve}(\mathbf{va}_Y(0.5), \mathbf{va}_X(d_X), \mathbf{va}_Y(d_Y)))]$$

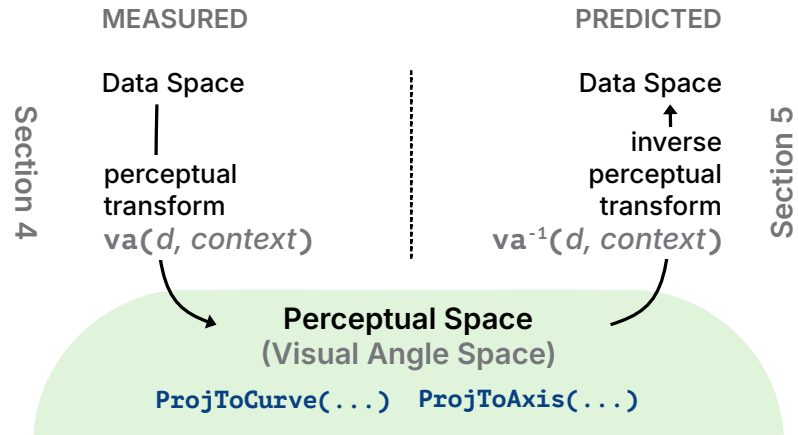
visual angle of distance to axis
102

visual angle of $y = 0.5$
visual angle of distance to curve

... where $\mathbf{va}_Y(\cdot)^{-1}$ is the inverse visual angle function that transforms a value in visual angle space back to data values along the y-axis, given relevant context-specific parameters.

Ultimately, we aim to demonstrate that under this modeling approach, we can decompose tasks into and generate estimates of individual decoding operators (§ 5.4), combine different operators (§ 5.4.2.4), and finally generate predictions for new experimental conditions and questions that the existing set of operators are not trained on (§ 5.5).

Figure 36: A horseshoe diagram [177] of this paper's approach to modeling.



5.4 BUILDING OPERATORS

In the following section, we detail our experimental approach for addressing two methodological challenges: 1) ensuring participants perform the intended visual operator(s) to minimize the effects of the identifiability problem and 2) obtaining measurements in online crowd-sourced studies, such as the physical distance to the screen and participants' responses as outcomes of visual decoding operators.

5.4.1 EXPERIMENT DESIGN

5.4.1.1 TASKS, INSTRUCTIONS, AND TRAINING

We use both probability density function (PDF) charts and cumulative distribution function (CDF) charts, focusing on **median** and **mode** identifications. We selected these statistics because the process of reading median and mode can be decomposed into separable visual operators, as demonstrated in Section 5.3.

A challenge in prior work is that tasks often require participants to already understand statistical concepts. Errors in such tasks may reflect misunderstanding of definitions rather than imprecision in perception. Thus, we avoided directly asking for statistical measures and instead decomposed abstract tasks into concrete graphical operations (Tab. 2). More concretely, rather than asking for “the median”, our instructions **specify the exact graphical components and actions required (effectively, the visual operators to use)**. For example, when eliciting median estimates from a CDF, we instruct participants to position a red dot where $y = 0.5$ intersects the CDF curve.

Task	Experiment Instruction
PDF Mode HighestPoint	<i>“Drag the slider to move the ball along the curve to the highest point (where y is greatest)”</i>
PDF Median BisectArea	<i>“Drag the slider to move the ball along the curve until a hypothetical vertical line through it divides the area into two equal halves.”</i>
CDF Mode MaxSlope	<i>“Drag the slider to move the ball along the curve until it rests where the curve crosses $y=0.5$”</i>
CDF Median ProjToCurve	<i>“Drag the slider to move the ball along the curve until it rests where the curve crosses $y = 0.5$”</i>
ProjToAxis_x , ProjToAxis_y	<i>“Drag the sliders such that the two balls on the x-axis and y-axis match the red dot.”</i>

Table 2: Tasks and their representative instructions used during experiment.

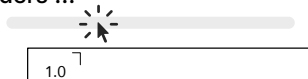
To further minimize confounding factors such as instruction misunderstanding, we provided screen recordings of intended behavior and training trials for each task. This approach allows us to attribute errors to the visual operators themselves rather than to task comprehension. Given our precise characterization of the error due to visual operators, future experiments should be able to precisely measure the additional error due to participants incorrectly translating the task to those operators (due to lack of prior knowledge or visualization literacy).

5.4.1.2 STIMULI AND APPARATUS

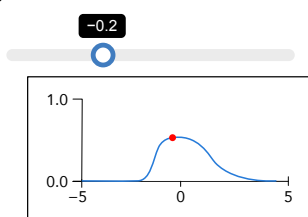
We used the skewed generalized t distribution [171] as our data-generating distribution for its ability to independently control skewness, kurtosis, and scale (details in Appendix A.3.3). Parameters of the SGT distribution were sampled such that the target statistics (i.e., median and mode) vary in the ranges of $x \in [-5, 5]$, $y \in [0, 1]$. Stimuli data were generated using `stidlib-js` [164] and rendered with D3 [11] at dimensions of $600 \text{ px} \times 450 \text{ px}$ (details in Appendix A.3.2).

To minimize potential confounds, we implemented a custom elicitation interface with several key design features. We used a slider with no default starting position to prevent anchoring effects. The slider was positioned away from the chart axes to prevent participants from directly aligning the slider position with axis values, encouraging them to focus on the red dot annotation that appears on the curve itself. Participants first clicked anywhere on the slider to initialize it (a), which displayed a red dot annotation on the curve at the corresponding position (b). They could then control the red dot's position by dragging the slider or using left and right arrow keys (c). We chose this slider-based approach over alternatives, such as direct clicking on the curve, because it eliminated ambiguity about the source of error while constraining participants to

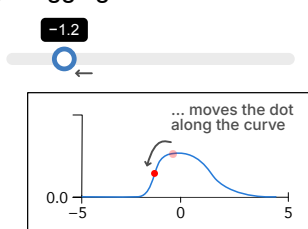
(a) users click on un-initialized sliders ...



(b) ... to initialize it.



(c) dragging the slider ...



perform the specific visual operators under investigation. We used the ReVISit platform [30,38] to elicit responses from participants⁵⁵.

⁵⁵ See example interface at <https://gleaming-dolphin-799fda.netlify.app/vis-decode-slider/>

5.4.1.3 EXPERIMENT PROCEDURE AND PARTICIPANTS

After consenting on the experiment website, participants first performed *calibration tasks* to measure screen-to-pixel ratio and eye-to-screen distance. The experiment consisted of five blocks, one for each task. For each task, participants first viewed instructional screens and a short video clip explaining the task, completed two training trials with fixed stimuli across participants, and then performed ten testing trials with randomized stimuli, resulting in a total of $5 \times (10 + 2) = 60$ trials per participant. Finally, participants provided demographic information and optional text feedback on their strategies and the clarity of instructions.

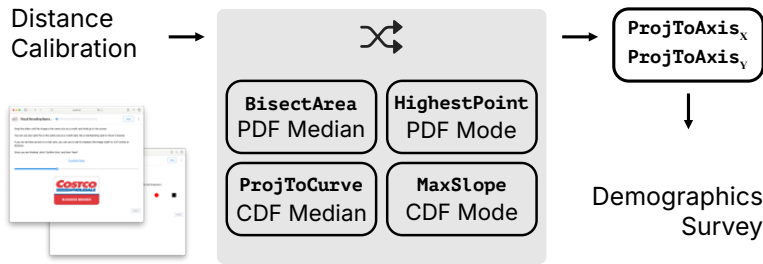


Figure 38: Diagram of experiment procedure. The study used a within-subjects design.

We recruited 16 participants in an IRB-approved study (STU00223203) on Prolific.co who are between the ages of 18 and 65, have normal or corrected-to-normal vision, speak fluent English, and reside in the United States. The study was distributed to a balanced sample of participants. We paid each participant \$4.86 for completing the experiment, with a potential bonus of up to \$1.50 depending on the participants' precision. The experiment took on average 18.6 minutes to finish, resulting in an hourly wage of \$20.51 / hr, exceeding the US federal minimum wage.

5.4.1.4 ANALYSIS APPROACH

We use multi-level Bayesian regression to analyze our results. Specifically, we fit models using values transformed from physical space (cm) into visual angle space, unless otherwise specified. We excluded participants with Pearson correlation < 0.5 to the actual answer or have eye-to-screen distance < 20 cm, as typical participants showed correlations > 0.95 and maintained the expected viewing distance of approximately 50 cm. We fit five Bayesian hierarchical mixed effects models, with all parameters achieving bulk effective sample size (`Bulk_ESS`) and tail effective sample size (`Tail_ESS`) values ranging from 1976 to 19940. We assess convergence using the Gelman-Rubin diagnostic ($\hat{R} = 1.00$ for all population-level parameters). We assess all model fits using posterior predictive checks following the recommendations of Säilynoja et al. [152].

5.4.2 RESULTS

We characterize five operators across four tasks: projection operators (`ProjToCurve` , `ProjToAxisx` , `ProjToAxisy`) from CDF median identification (§ 5.4.2.1), `HighestPoint` from PDF mode (§ 5.4.2.2), `MaxSlope` from CDF mode (§ 5.4.2.3), and `BisectArea` from PDF median (§ 5.4.2.4).

5.4.2.1 PROJECTION OPERATORS

`ProjToCurve` then `ProjToAxisx` , i.e., median identification on a CDF (Fig. 35), involves projecting from $y = 0.5$ to its intersection on the CDF curve and then projecting from that intersection to the x -axis. We model all projection-related operators as Normal distributions over values parameterized in visual angle on the respective axis. For all projection tasks, we hypothesize that the variance of the operator's output

increases with the visual angle distance between the projection start point and end point.

The following model is for **ProjToAxis_x** for a particular participant j :

$$\text{ProjToAxis}_x(\theta_X, \theta_Y) \sim \mathcal{N}(\theta_X + \beta_j, \sigma_j)$$

visual angle of true value
↓

ProjToAxis_x operator returns a Normal distribution centered at the true value plus a participant-specific bias β_j , capturing whether participant j systematically over- or under-estimates the projected position. The observation-specific standard deviation σ_j captures how much the participant's responses spread for a given trial:

$$\sigma_j = \alpha_j \cdot \theta_Y$$

The standard deviation scales multiplicatively with how far the point must be projected, i.e., the visual angle distance to x -axis, θ_Y , where α_j acts as a scaling parameter for participant j . We compared additive and multiplicative scaling relationships on pilot data; the multiplicative specification yielded better posterior predictive calibration and is used throughout. The models for **ProjToCurve** and **ProjToAxis_y** are defined similarly as the model above. Details of the full hierarchical specifications⁵⁶ are in Appendix A.3.

⁵⁶... which model population means and standard deviations of β_j and α_j

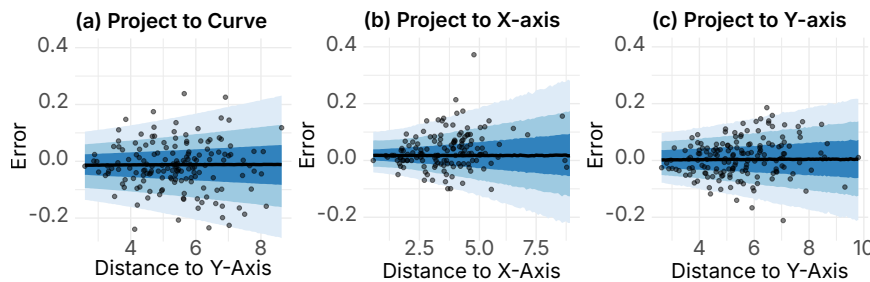
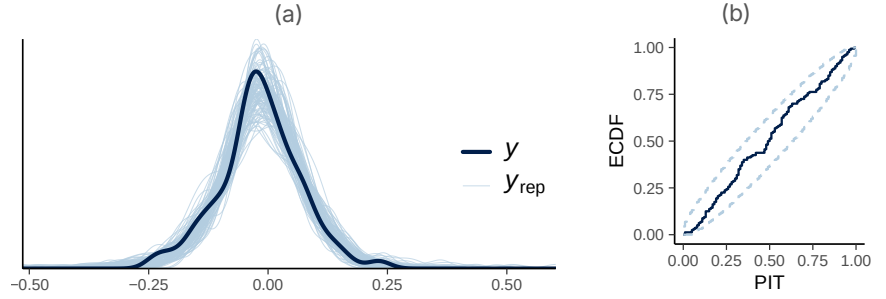


Figure 39 shows observed errors plotted against projection distance for all three projection operators, with posterior predictive intervals marginalizing over participants. In all three cases, the spread of errors increases with distance, consistent with the multiplicative scaling spec-

Figure 39: Plots for (a) **ProjToCurve**, (b) **ProjToAxis_x**, and (c) **ProjToAxis_y**: error vs. distance between start and end points of projection, where points show observed errors with blue bands representing 50%, 80%, and 95% posterior predictive intervals, marginalizing over participants.

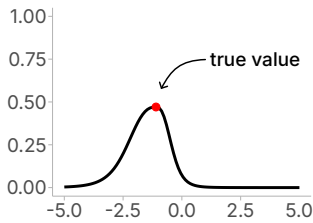
ification. The posterior predictive intervals cover the observed data well, with most points falling within the 95% interval.

Figure 40: Posterior predictive checks for **ProjToCurve**. (a) KDE density overlay plot with 100 posterior draws. (b) PIT-ECDF Calibration plot.



We assess model fit using two standard posterior predictive checks (Fig. 40). In the density overlay (Fig. 40 (a)), the dark line shows the observed distribution of responses and each lighter line is a density drawn from the model's posterior predictive distribution. If the model is adequate, the observed density should fall among the predicted densities in location, spread, and shape. The PIT-ECDF plot (Fig. 40 (b)) transforms each observation by its posterior predictive CDF and plots the resulting empirical CDF against a uniform reference. Adequate calibration means the empirical CDF falls within the simultaneous confidence envelope. For **ProjToCurve**, both diagnostics indicate adequate fit. Analogous checks for the remaining operators appear in Appendix A.3.4.

Q: Where is the highest point on the curve?



5.4.2.2 HIGHEST POINT OPERATOR

HighestPoint, i.e., mode identification on a PDF, asks participants to identify the highest point on the curve. While participants generally demonstrated high precision with this task, the distribution of errors exhibits a heavy tail (§ A.3.4.1.2). To capture this pattern effectively, we model the operator as returning the true value in visual angle θ minus a non-negative error term ε :

$$\text{HighestPoint}_Y(\theta) = \theta - \varepsilon$$

The error term follows a Weibull participant-specific distribution:

$$\varepsilon \sim \text{Weibull}(\lambda_j, k_j)$$

where $\lambda_j > 0$ and $k_j > 0$ are the scale and shape parameters for participant j respectively. The Weibull distribution generalizes the Exponential distribution through an additional shape parameter, allowing the rate of tail decay to vary. We selected it over other heavy-tailed candidates based on leave-one-out cross-validation (§ A.3).

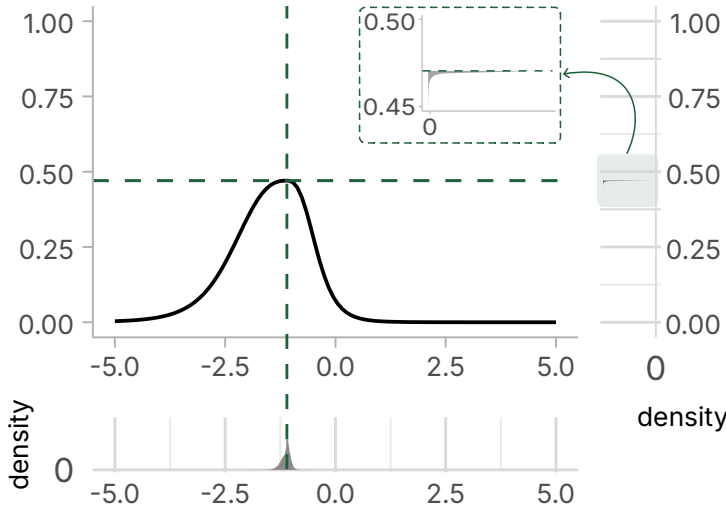


Figure 42: Posterior predictive error distributions for **HighestPoint_Y** applied to an example stimulus. The top-left panel shows the stimulus curve with the true peak marked. The bottom-left panel shows the marginal predictive distribution of error along the x -axis, and the top-right panel shows the corresponding distribution along the y -axis (**HighestPoint_Y**).

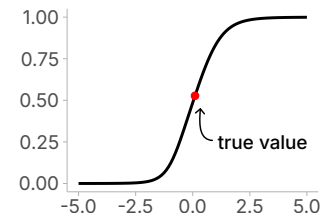
When the curve is relatively flat near the peak, even small errors along the y -axis can correspond to large displacements along the x -axis (Fig. 42). We derive a model⁵⁷ for x -axis error from the y -axis error through the geometry of the stimulus curve. In Section 5.4.2.4 we provide an alternative model for x -axis error for **HighestPoint** using a Normal distribution for tractability within the sensor fusion framework.

5.4.2.3 MAX SLOPE OPERATOR

MaxSlope, i.e., mode identification on a CDF, requires judging the point on the curve whose tangent has the maximum slope. We express slope in visual angle space by deriving the mapping between $\Delta x/\Delta y$ in data space and visual angle space. Similar to **HighestPoint_Y**, we model the

⁵⁷ ... more specifically, we derive the distribution of x axis errors by transforming the distribution of y axis errors through the inverse of the stimulus curve.

Q: Where is the slope the steepest on the curve?



MaxSlope operator as returning the true value minus a non-negative error term ε :

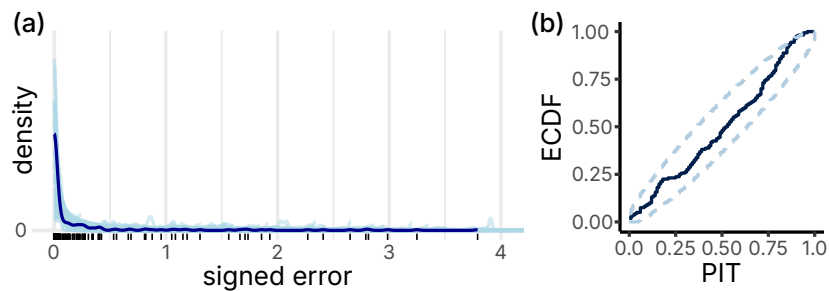
$$\mathbf{MaxSlope}(\theta) = \theta - \varepsilon$$

Because the true answer is the maximum slope on the curve, any other response necessarily has a smaller value. Participants thus tend to underestimate the maximum slope, and the distribution of errors is asymmetric with a heavy tail. We thus model this error ε using a Weibull distribution, selected over other heavy-tailed candidates via leave-one-out cross-validation (see Appendix A.3):

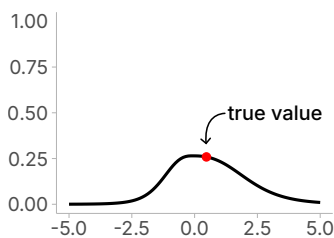
$$\varepsilon \sim \text{Weibull}(\lambda_j, k_j)$$

where $\lambda_j > 0$ and $k_j > 0$ are the scale and shape parameters for participant j respectively.

Figure 44: (a) Posterior retrodictive density of participants' errors overlaid with density of observed errors, plotted on top of strip plot of observed errors for **MaxSlope**. (b) PIT-ECDF Calibration plot.



Q: Where is the vertical line that splits area under the curve into two halves?



5.4.2.4 BISECT AREA OPERATOR

In pilots, we observed non-uniform errors, and posited that due to the difficulty of the task, participants may use two different operators to accomplish this task. For unimodal distributions like those in our experiment, when the distribution is *symmetric* the mode and median are equal and one can use the low-error **HighestPoint_x** operator. Even if skewness is low, the mode may still be a good approximation of the median considering the low perceptual error of **HighestPoint_x**. As described in Section 5.4.2.2, x-axis error is derived from y-axis error through the stimulus curve's geometry and approximated as Normal:

$$\mathbf{HighestPoint}_x(\theta) \sim \mathcal{N}(\theta + \beta_j^{\text{HP}}, \sigma_j^{\text{HP}})$$

When skewness is high, one may have to resort to a higher-error **BisectArea** operator, visually estimating the point that divides the region into two equal-area halves.

$$\mathbf{BisectArea}(\theta) \sim \mathcal{N}(\theta + \beta_j^{\text{BA}}, \sigma_j^{\text{BA}})$$

But how *precisely* do people select when to use which operator? One natural hypothesis is that participants probabilistically select between the two operators on each trial (a *mixture* model). However, posterior predictive checks of a mixture model do not adequately capture the observed response distribution (see Appendix A.3).

Alternatively, prior work [160] and our own observation suggests that an *anchoring-and-adjustment* process could be at work: perhaps participants first identify the mode of the PDF (anchoring), then adjust based on the skewness of the distribution. We model this by combining estimates from both operators using *inverse mean squared error weighting*, a technique commonly used in sensor fusion [41] for giving more weight to more reliable sensors [184]. Specifically, we assume that for the combined operator, which we dub **BAHP**, participants' selections follow a weighted mean of the other two operators:

$$\mathbf{BAHP}(\theta_{\text{median}}, \theta_{\text{mode}}) = \underset{\substack{\uparrow \\ \text{how much a person weights one operators versus another in their estimate}}}{w} \cdot \mathbf{BisectArea}(\theta_{\text{median}}) + (1 - \underset{\uparrow}{w}) \cdot \mathbf{HighestPoint}_x(\theta_{\text{mode}})$$

how much a person weights one operators versus another in their estimate

BisectArea targets the median directly, so its mean squared error (MSE) reflects only its own bias and variance:

$$\text{MSE}(\mathbf{BisectArea}) = (\beta_j^{\text{BA}})^2 + (\sigma_j^{\text{BA}})^2$$

HighestPoint_x targets the mode, so its MSE relative to the median includes the error caused by using the mode to approximate the median:

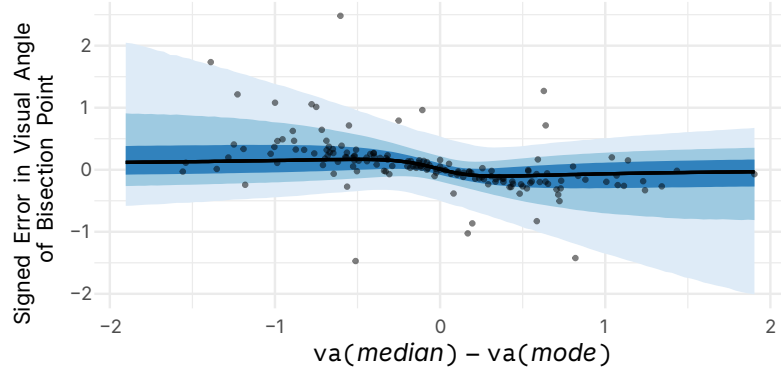
$$\text{MSE}(\mathbf{HighestPoint}_x) = (\theta_{\text{mode}} - \theta_{\text{median}} + \beta_j^{\text{HP}})^2 + (\sigma_j^{\text{HP}})^2$$

Note that β_j^{HP} and σ_j^{HP} are not re-estimated, as they are already learned from Section 5.4.2.2. The last step is to combine these two estimates using inverse-MSE weighting to define w :

$$w = \frac{\text{MSE}(\mathbf{BisectArea})^{-1}}{\text{MSE}(\mathbf{BisectArea})^{-1} + \text{MSE}(\mathbf{HighestPoint}_x)^{-1}}$$

Modeling $\mathbf{HighestPoint}_x$ as drawn from a Normal distribution allows for a closed-form likelihood function within the sensor fusion framework, simplifying modeling. Importantly, posterior predictive checks for a Gaussian fit to the $\mathbf{HighestPoint}_x$ data (Appendix A.3.4.1.3) still indicate a reasonable model fit, suggesting this simplification retains adequate descriptive power for its role in the combined model

Figure 46: Posterior density, 50%, 85% and 95% quantile interval of signed error of **BAHP** overlaid on observed signed error from participants.



This adaptive weighting between operators appears to better capture the observed pattern of responses compared to the simple mixture model, as confirmed by our posterior predictive checks (see Figure 58 in Appendix). While the mixture model failed to account for the systematic change in variance of participants' judgments across different skewness conditions, the weighted average model successfully captures this change, providing a superior fit to the empirical data. The sensor fusion model also provides a more cognitively plausible explanation

for participants' behavior — when the median and mode are close, participants tend to rely more heavily on their estimate of the mode (`HighestPointx` operator), which is more precise and has smaller variance (Fig. 46). However, as the distribution becomes more skewed and the distance between median and mode increases, participants can no longer effectively use the mode as a heuristic. In these cases, they increasingly incorporate the `BisectArea` operator, which requires area judgment and is thus less precise.

5.5 APPLYING LEARNED OPERATORS

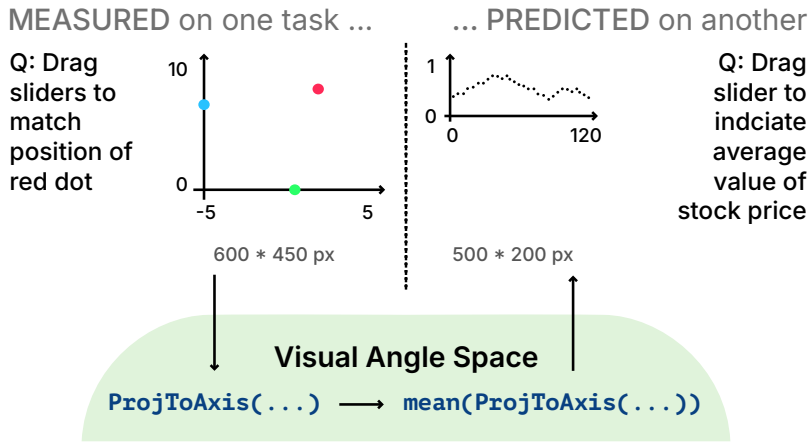


Figure 47: Overview of the cross-task generalization approach. Left: the `ProjToAxisy` operator is measured by asking participants to project a dot to the y-axis on a 600×450 px chart. Right: the same operator, without refitting, is composed under candidate averaging strategies to predict mean-estimation responses on Moritz et al.'s [120] 500 × 200 px scatterplot task. All composition and prediction occurs in the perceptually-uniform visual angle space (center).

To demonstrate the capabilities of our framework, we re-implemented experiment 2 from Moritz et al. [120], which asked participants to estimate the mean y-value of a scatterplot containing 60 data points under varying display conditions. This task requires participants to compose multiple perceptual operations — projecting individual points then aggregating across them — making it a natural test case for our operator framework. As an overview, we first re-administer `ProjToAxisy` task from our first study to a new set of participants, deriving decoding operators for each. We then use these per-participant operators to generate predictions under three pre-registered composition strategies for a mean estimation task, and evaluate which strategy best accounts for

empirical performance. Our analysis plan is pre-registered at <https://osf.io/wkvbe/>.

5.5.1 MORITZ ET AL. EXPERIMENT 2

Moritz et al. [120] investigated what they term *variability overweighting*: the hypothesis that when estimating the average of a time series, viewers' judgments are systematically pulled toward regions of higher variability. In their second experiment ($N = 420$), they manipulate the mark type between subjects, comparing line graphs to two point-based encodings: points equally spaced along the x -axis (point) and points equally spaced along the arc of the line (pointArc).

The experiment used a mixed design. Between subjects, participants were randomly assigned to one of three graph types. Within-subjects factors were 2 variability levels (0 vs 0.4) $\times$ 2 variability positions (upper vs lower half of the chart) $\times$ 12 random seeds (i.e. stimuli datasets) = 48 trials per participant. Stimuli were synthetic time series of 60 data points generated from geometric Brownian motion, with noise linearly interpolated as a function of the base value. Each participant estimated the average of 48 graphs by adjusting a draggable horizontal line.

5.5.2 OUR EXPERIMENT DESIGN

5.5.2.1 TASKS, INSTRUCTIONS, AND TRAINING

We first follow Section 5.4.2.1 to learn **ProjToAxis**, and then follow the mean estimation setup in Moritz et al. [120]. We added two training tasks for the Moritz experiment to familiarize participants with the different task interface and instructions.

5.5.2.2 STIMULI AND APPARATUS

We reused the experiment interface for **ProjToAxis** from the first experiment, and now use uniform random sampling from $[0, 200] \times [0, 500]$ to place the dot to be projected to the x and y -axis. For the second part of

the experiment, we followed Moritz et al.'s experiment 2 stimuli design and utilized the exact same stimuli obtained from their code repository⁵⁸. We include only the point and pointArc conditions from Moritz et al., omitting the line condition.⁵⁹ Our current operator vocabulary includes projection operators for dot-to-axis mappings but not for line-to-value extraction, which would require an additional decomposition step (e.g., line to point) that is outside the scope of this work.

⁵⁸<https://osf.io/aupbk/>

⁵⁹Example interface at <https://gleaming-dolphin-799fda.netlify.app/vis-decode-retrieve-value/>

5.5.2.3 EXPERIMENT PROCEDURE AND PARTICIPANTS

Similar to the first experiment (§ 5.4.1.3), participants first consent, then perform calibration tasks. The first part of the experiment is a short video clip explaining the **ProjToAxis** task, two training trials with fixed stimuli across participants, then ten testing trials with randomized stimuli. Then the participant is randomly assigned to one of two conditions, point or pointArc, and each condition has two fixed training trials and 48 testing trials, presented to them in random order. In total, each participant performed $10 + 48 = 58$ trials. Finally, participants provided demographic information and optional text feedback on their strategies and the clarity of instructions

Similar to the first experiment, we recruited 20 participants in an IRB-approved study⁶⁰ on Prolific.co who are between the ages of 18 and 65, have normal or corrected-to-normal vision, speak fluent English, and reside in the United States. The study was distributed to a gender-balanced sample of participants. We paid each participant \$3.10 for completing the experiment, with a potential bonus of up to \$2.90 depending on the participants' precision. The experiment took on average 20.5 minutes to finish, resulting in an hourly wage of \$17.56 / hr, exceeding the US federal minimum wage.

⁶⁰STU00223203

5.5.3 ANALYSIS APPROACH

Similar to Section 5.4.1.4, we used multi-level Bayesian regression to analyze our results. We fit Bayesian hierarchical mixed effects model on data from part 1 for **ProjToAxis_Y**, with all parameters achieving bulk effective sample size (**Bulk_ESS**) and tail effective sample size (**Tail_ESS**) values ranging from 2981 to 7981. We utilized the same set of R packages as Section 5.4.1.4, with the addition of `jsonlite` [125] and `purrr` [189].

5.5.3.1 HYPOTHESIZED STRATEGIES

Each trial presents 60 points (x_i, y_i) , $i \in [60]$ where the participant estimates the mean y -value, which we denote $\hat{\mu}_Y$. We posit strategies for mean estimation composed of projection operations and mental averaging. Our strategies vary along two dimensions: *projection path* (project-once vs. project-twice) and *aggregation rule* (mean, median, sensor-fusion), yielding six candidate strategies.⁶¹

⁶¹Our pre-registered analysis specified three strategies (project-once $\times$ mean, project-once $\times$ median, project-twice $\times$ sensor-fusion). During analysis, we recognized these three span a partial crossing of two orthogonal design factors, projection path and aggregation rule, and completed the full 2×3 factorial to avoid selectively pairing projection paths with aggregation strategies. No data-dependent decisions guided which cells to add; the posterior prediction pipeline is deterministic given the learned operator and the composition rule.

Projection paths. We consider two possible projection paths:

1. **Project-once.** The participant projects each point directly to the y -axis then takes an average:

$$\hat{\mu}_Y = \mathbf{va}_Y^{-1}[\text{Average}(\mathbf{ProjToAxis}_Y(\mathbf{va}_Y(y_i), \mathbf{va}_X(x_i)))]$$

where *Average* represents one of the *aggregation rules* (detailed below). As each projection traverses a horizontal distance of $\mathbf{va}_X(x_i)$, points farther from the y -axis incur more noise. This may not be a plausible account, motivating our second projection path.

2. **Project-twice.** Instead of projecting each point all the way to the y -axis, the participant first projects each point to an imagined vertical reference line at the midpoint (in this case, $x = 60$):

$$\hat{\theta}_i = \mathbf{ProjToAxis}_Y(\mathbf{va}_Y(y_i), \mathbf{va}_X(|x_i - 60|))$$

The participant then averages the projected points and projects that average from the midpoint to the y -axis:

$$\hat{\mu}_Y = \mathbf{va}_Y^{-1}[\mathbf{ProjToAxis}_Y(\text{Average}(\hat{\theta}_i), \mathbf{va}_X(60))]$$

This account may be more plausible if participants visually estimate the mean around the center of the data along both axes rather than at one edge.

Aggregation rules. Given internally-estimated visual angles $\hat{\theta}_i$ for each y -value $y_{i,i \in [60]}$ (from either projection path), we consider three possible definitions of the perceptual **Average** operation:

1. **Mean: Average** ($\hat{\theta}_i$) = $\text{mean}_{i \in [60]}(\hat{\theta}_i)$
2. **Median: Average** ($\hat{\theta}_i$) = $\text{median}_{i \in [60]}(\hat{\theta}_i)$
3. **Weighted mean: Average** ($\hat{\theta}_i$) = $\sum_{i \in [60]} w_i \hat{\theta}_i$, where $w_i = \frac{\text{MSE}(\hat{\theta}_i)^{-1}}{\sum_{k \in [60]} \text{MSE}(\hat{\theta}_k)^{-1}}$ weights each point by the inverse MSE of the **ProjToAxis_y** operation that produced it.

5.5.4 RESULTS

5.5.4.1 PART 1: ProjToAxis_y OPERATOR GENERALIZES ACROSS PARTICIPANT GROUPS

For the first part of the experiment, we fit the same model as in Section 5.4.2.1 and obtained a similar trend to Figure 39 (c): error increases with the visual angle distance between the point and the y -axis (Figure 48; posterior predictive checks in Appendix A.3.5.1). This provides preliminary evidence that our model is capturing a trend that generalizes across participant pools.

5.5.4.2 PART 2: COMPARING COMPOSITION STRATEGIES

The model fitted in Section 5.5.4.1 gives us posterior distributions for the parameters of each participant's **ProjToAxis_y** operator. Using these parameters, we can generate a predictive distribution for how each participant would respond if they had used each combination of *projection path* $\times$ *aggregation rule* on each mean-estimation trial in part 2 (Moritz et al. [120] experiment 2). We emphasize that these posterior predictions are generated entirely from operators learned on the **ProjToAxis_y** task

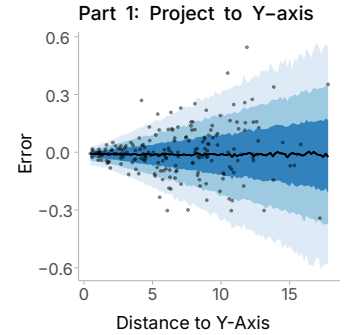


Figure 48: Error vs. distance between start and end points of projection for **ProjToAxis_y**, measured in radian units, where points show observed errors with blue bands representing 50%, 80%, and 95% posterior predictive intervals, marginalizing over participants.

on charts with a different size and aspect ratio, *then* re-composed under each prospective aggregation strategy to create predictions. *No parameters are fit to the mean-estimation data* (Fig. 49).

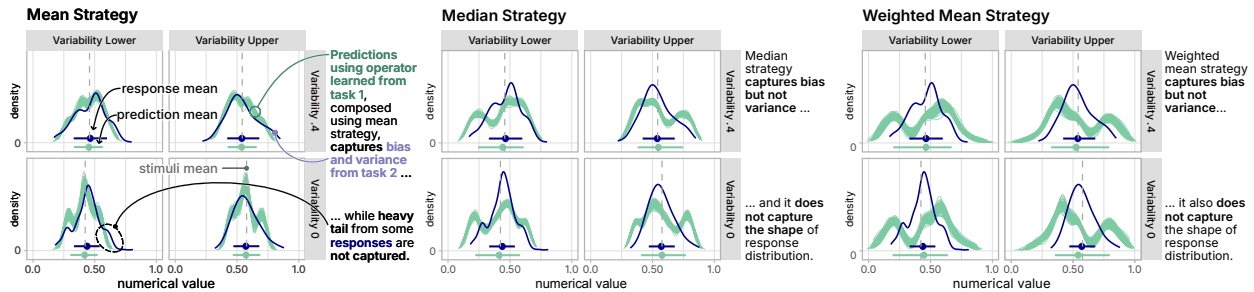


Figure 49: Cross-task posterior predictive density overlay plot, each with 100 draws, faceted by variability upper/lower vs. variability 0/0.4 experiment conditions. Point intervals indicate the mean and 68% quantile interval (approximately ± 1 SD) for observed responses and model predictions respectively.

Figure 49 shows posterior predictive density overlays for all three strategies for **project-twice**, faceted by variability conditions. The project-twice mean strategy (Fig. 49 left) captures both the central tendency and spread of observed responses across all four conditions. The median strategy (Fig. 49 middle) matches observed bias but produces prediction intervals that are too wide. The weighted-mean strategy (Fig. 49, right) shows a similar pattern. We defer posterior predictive density overlays for the **project-once** strategies to Appendix A.3.5.2, as these produced substantially narrower prediction intervals that failed to match observed responses.

5.5.4.3 IMPLICATIONS

That the project-twice mean strategy can capture bias and variance on a structurally different task, where five alternatives fail, has two implications. First, it provides evidence that individually-measured operators can generalize through composition to predict performance on unseen charts with a different size and aspect ratio, *without refitting*. Together with the operator-level results in Section 5.4.2, this constitutes an existence proof that visual decoding operators can be measured in isolation, composed under candidate strategies, and evaluated against held-out task performance. The strategy comparison also allows us to rule

out some prospective perceptual aggregation functions: e.g., it seems unlikely people are mentally calculating medians here.

5.6 DISCUSSION

Together, Section 5.4 and Section 5.5 establish that operators can be isolated, parameterized, and composed to predict task performance on unseen display conditions. Below, we consider how operators provide a compositional interface for formalizing graphical perception knowledge and enabling a new class of design tools (§ 5.6.1), what kind of theoretical contribution this represents within existing frameworks for model evaluation (§ 5.6.2), and what empirical work is needed to build out an operator library (§ 5.6.4).

5.6.1 A COMPOSITIONAL INTERFACE TO FORMALIZE GRAPHICAL PERCEPTION KNOWLEDGE

The visualization community has developed effective **domain-specific languages** to author visualizations (e.g., Vega-Lite [156]), and increasingly, to run experiments (e.g., ReVISit [30,38]). What we lack is a corresponding formal language for specifying how people *interpret* visualizations, one that can express a decoding procedure as a composition of measurable operations rather than as a verbal description or a ranked list. Visual decoding operators are a step toward filling this gap.

Consider what existing approaches provide. Ranking-based frameworks decompose visualizations into channels, and order them by effectiveness, determined by the tasks. While useful for pruning design spaces, an ordinal ranking is a coarse data structure. It tells a recommendation system that position is more effective than angle *under certain scenarios*, but not how much more effective, under what viewing conditions, or for which tasks. When Zeng et al. [206] incorporated results from previous graphical perceptual studies into Draco [121], they found

that individual papers could dominate or cancel each other’s influence in unpredictable ways. This is not because the prior studies were flawed, but because a constraint system built on ordinal preferences has no principled mechanism for resolving contradictions across studies. This instability is predicted by known results in social choice theory: no procedure for aggregating ordinal preferences can guarantee consistent outcomes across all inputs [3]. Rankings are *structurally lossy* — they discard the magnitude, context-dependence, and uncertainty of the underlying perceptual measurements, and systems built on this interface, however sophisticated their constraint solvers, inherit that loss.

One might hope that large vision-language models (VLMs) could bypass this problem by learning to interpret visualizations directly from data or learn visual design preferences [182]. Recent evidence suggests otherwise on two fronts. First, VLMs lack fundamental visual processing capabilities: they fail to solve tasks that three-year-old children could solve effortlessly [18], such as counting line intersections, detecting overlapping circles, and tracking colored paths [145]. Second, VLMs appear to be heavily influenced by textual priors: VLMs report visual illusions in images that contain none [175], and can correctly answer visualization questions without the visualization being shown at all [110]. These limitations suggest that VLMs do not yet offer a shortcut to the prediction problem identified above.

Operators provide a different kind of interface. Because operators are parameterized by *perceptual units*⁶² rather than screen pixels, they generalize across display context. And because operators are composable, complex tasks can be expressed as sequences of operations, each carrying a quantified cost. This composability opens several practical applications in visualization design. For instance, operators can serve as

⁶²here visual angle, but we hope by other perceptual units in the future, such as values in perceptually-uniform color spaces.

cost functions in a *query planner*: given a dataset and an analytic goal, the planner enumerates candidate visualization designs, decomposes each into its operator sequence, and predicts a user’s likely decoding path by balancing the cognitive cost of each operation against its expected informational gain. This differs from current ranking-based recommendation systems, in that the predicted error distribution tells the designer not just which option is better but *by how much* and *with what uncertainty*. The error measurements in this paper provide initial cost estimates; a fuller library (§ 5.6.4) would expand the space of tasks and chart types over which such a system can reason.

5.6.2 TOWARD ALGORITHMIC-LEVEL THEORIES OF VISUALIZATION INTERPRETATION

Visual decoding operators specify *how* a person extracts a piece of information from a chart: which visual features serve as input, what transformation is applied, and what error distribution results. This is what distinguishes the framework from a behavioral summary. A behavioral summary records that viewers underestimate means in scatterplots with high variability; an operator account specifies the projection and averaging operations that produce the underestimate, predicts its magnitude, and identifies which component to revise when the prediction fails.

This distinction maps onto a well-known framework for evaluating models of information-processing systems. Marr [116] proposed three levels of analysis: the **computational level**,⁶³ the **algorithmic level**,⁶⁴ and the **implementational level**.⁶⁵ Most modeling work in visualization operates at the computational level — we have extensive catalogs of tasks and methods for measuring their aggregate error [40], and recent models simulate observable behavior such as eye scan paths [159]

⁶³i.e., *what problem is the system solving?*

⁶⁴i.e., *what procedures does it use?*

⁶⁵i.e., *how is it physically realized?*

or leverage LLMs to predict task outcomes [183]. These approaches describe what the system does, or predict what it will do, but not *how*. Visual decoding operators engage at the *algorithmic level*: they specify the sequence of transformations a viewer executes and the error each transformation introduces.

The value of this level of description extends beyond taxonomy. Wichmann and Geirhos [188] argued that model evaluation is multidimensional: a good model should both predict and explain. Our cross-task results in Section 5.5 demonstrate prediction, in that operators learned on one task generate calibrated posterior predictive distributions for a structurally different task. The sensor fusion finding in Section 5.4.2.4 demonstrates explanation, in that the weighted combination of **High-estPoint_x** and **BisectArea** not only fits the data better than a mixture model but provides a mechanistic account of why variance changes with skewness, an account that a purely descriptive model would not generate.

Van Rooij et al. [151] have argued that psychology and its adjacent fields remain dominated by what they term *naïve empiricism*: accumulating experimental results without theoretical infrastructure to organize them into explanatory models. The visualization community faces a version of the same challenge. Verbal theories in graphical perception often provide post-hoc accounts of empirical patterns that appear explanatory but lack the computational structure to generate quantitative predictions for new conditions. The operator framework is one response to this challenge: it provides units whose predictions are falsifiable at the component level, so that when a composed prediction fails, the failure points to which operator needs revision rather than leaving an undifferentiated residual.

5.6.3 PERCEPTUAL PROXIES AND THE PROBLEM OF IDENTIFICATION

This chapter is not the only research attempt to move beyond static rankings. A related but different line of work [85,124,199,202] explores “*perceptual proxies*”, heuristic visual shortcuts such as shape outlines or centers of mass [124], that approximate how the visual system might extract quantitative information. The shared methodology between these works typically proceeds by 1) examining a data fact/signal of interest for the visual system to decode,⁶⁶ 2) coming up with a list ^{66 ... such as correlation} of pre-defined decoding heuristics, inspired by works in vision science and psychology, and 3) checking the compatibility of those proxies with observed behavior when study participants are asked to judge the data fact/signal of interest.

While promising, this approach faces a fundamental problem of *identification*: we cannot always know from behavioral outcomes alone which feature a participant actually used, because different features/heuristics often predict the same patterns of behavior in a given experiment. ⁶⁷ In other words, the mapping between a participant’s response (e.g., correlation estimate) and a visual feature (e.g., center of mass, shape of bounding box) is rarely one-to-one. This ambiguity makes it difficult to build generative models that can generalize across individuals, contexts, and tasks.

⁶⁷ To truly get at that question we need data of a more granular modality, e.g., eye-tracking data or neural imaging.

We do not claim to fully resolve the problem of identification. Rather than inferring possible visual operators, we constrain the space of possible visual operators by design. Our instructions prescribe the visual operations to perform, ample training, and an elicitation interface (§ 5.4.1.1). The cost of this approach is that we do not have a way to

capture how viewers spontaneously decompose tasks, which we return to in Chapter 6.

5.6.4 LIMITATIONS AND FUTURE WORK

Our experiments use explicit task instructions to isolate individual operators, leaving open whether people spontaneously decompose visualization tasks the same way when uninstructed. Future work could collect open-ended responses and test whether composed operator predictions produce posterior predictive distributions consistent with uninstructed behavior. If no composition fits, the operator set is incomplete or the decomposition is wrong; both outcomes are informative. Eye-tracking could provide complementary evidence: if uninstructed participants exhibit the predicted fixation sequence, this would support the decomposition itself, though not the error parameters of individual operators.

We characterized five operators across four tasks and demonstrated cross-task generalization on one additional task, but these five need not be definitive. If alternative operators better predict performance for the same tasks, or capture judgments we did not test, they should replace or supplement ours — the constraint is that each must be independently parameterizable and composable, not that it must match our derivation. Existing empirical work already suggests extensions: for example, Talbot et al.'s findings on aspect-ratio-dependent bias could refine the slope operator or motivate a new one [\[168\]](#).

Once operator-level error is established as a perceptual baseline, remaining error can be more readily attributed to non-perceptual sources such as task mistranslation, training effects, or elicitation artifacts. Building the operator library that the design applications in Section 5.6.1 require is itself a substantial empirical program, one that demands new

experiments designed to isolate operations we have not yet characterized, across chart $\times$ task types we have not yet tested. The experiments required to populate such a library constitute a research program comparable in scope to the channel-ranking studies that followed Cleveland and McGill [23], reframed around reusable primitives whose compositions generate predictions beyond the conditions they were measured in.

6 GENERAL DISCUSSIONS

6.1 HOW SHOULD WE DO EMPIRICAL VISUALIZATION RESEARCH?

One consistent lesson across the three chapters is that the choice of **unit of analysis** determines what one can see. Chapter 3 shows that formalizing y-axis truncation in terms of structure-preserving transformations reveals task-dependent effects that previous heuristics don't explicitly capture. Chapter 4 shows that the task used to elicit similarity judgments shapes the inferred similarity representation itself.⁶⁸ Chapter 5 shows that decomposing tasks into visual operators is feasible and reveals that a precision-weighted sensor fusion approach better models how users integrate different operators than a probabilistic selection approach.

⁶⁸that is, the spatial arrangement task produces a global hierarchy (i.e., color before size) that pairwise and triplet tasks do not

In Chapter 5, I mentioned that the idea of “decomposition” is not alien to the field. In fact, while revisiting Cleveland and McGill's lesser known works, I found this paragraph in their 1987 work [24]:

One person's elementary particle can be another's complex system. For most of the field of chemistry, the molecule is the elementary particle, but the particle physicist sees the molecule as made up of a large number of more basic constituents.

The elementary codes [...] are one collection of elementary particles of graphical perception. Their virtue is that *they can be manipulated by the maker of a graph*.⁶⁹ However, from the viewpoint of vision theory they are not so elementary. The textons of Julesz (1981) are a more fundamental set of elementary particles of vision theory. The important point is that, like a chemist manipulating molecules to good purpose, the graph maker can manipulate the elementary codes to make better graphs.

⁶⁹ This is the original emphasis.

The purpose of this dissertation is not to discredit the ranking of visual channels approach. In fact, just like we (sometimes) lie to children⁷⁰ to help slowly introduce more complex concepts to them, we need ranking as it serves as a good first pass approximation for pedagogy and practical design guidance. But based on the amount of follow-up works that extend this approach,⁷¹ I find it only natural to question whether this approach is capturing enough internal structure of how people perceive visualizations that allows for generalization.

⁷⁰ Credits to Michael for pointing me to this concept; see more at <https://en.wikipedia.org/wiki/Lie-to-children>.

While I see the attractiveness of the ranking approach for practical purposes, I find it limiting that the field has adopted it as the dominant structure for reasoning about visualization perception. This categorical mode of reasoning produces a research program that functions like a giant design table whose cells enumerate every combination of data, encoding (channel or glyph), and task, and whose implicit aim is simply to fill them in — one study at a time, with no principled account of why one should fill a particular cell except for the sake of completing the table. But consider a thought experiment. Is the purpose of academic linguistics to teach people how to write better English prose? If not, why should the purpose of visualization research be about graph construction?

⁷¹ ... including extending it to on-line crowd-sourced studies [69], to correlation visualizations that don't nicely fit in individual visual channels [64], to specific populations such as older and younger adults [107,127], and to different tasks and data distributions [94] than the original ones used by Cleveland and McGill [23].

I believe the way that we've always done experimental evaluation is committing at least one class of category error by measuring the wrong things in the wrong ways, and thus resulting in a bunch of seemingly contradictory and non-interoperable results. This doctoral thesis serves at the very least a feasibility proof that we can do better.

As a field we should re-consider what a good unit of analysis is, and *what properties* do we want in such a unit of analysis. In Chapter 5, I have proposed three potential properties and demonstrated how visual decoding operators satisfy them: chart-agnostic, carrying quantitative profiles in perceptual units, and composable. I do not claim these are the only desirable properties; if we consider the ranking approach, elementary visual codes or visual channels are somewhat chart-agnostic, but they are not composable and *not measured in perceptual units*. In fact, the measurements obtained via particular tasks (such as the compare ratio task in Cleveland and McGill [23]) are always going to be conditional on that task, and as such tasks are not as generalizable as we think they are.⁷² This can partially be seen in the results of Chapter 4.

⁷²The field does not even agree on what we mean by “task”; see Pandey et al. [128]

At the very least, there needs to be more discussion about the properties that a good unit of analysis should possess, and whether these properties form trade-offs. Once we are clearer about the properties we want, the work is to search for a unit of analysis that has them, rather than defaulting to filling in the cells of a design tensor.⁷³

⁷³Another side rant: at the beginning of my doctoral program, someone once told me that their approach to research is that they have very strong skills in math, and so they wield this skill like a hammer and try to find nails for this hammer to work on. But what if there aren't nails? What if there are only screws? Filling the design table for the sake of filling the design table feels like hitting screws with hammers to me.s

On measurement: we should, at a minimum, perform checks that we are modeling what we intend to, and be transparent about the steps we took to ensure it. Related points have been better argued by Sarma et al. [155], so interested readers should go read that paper instead. The problem of identification is a hard one, but that does not mean we should sweep it under the rug and pretend it is not there.

6.2 WHAT'S NEXT?

6.2.1 TOWARD A LIBRARY OF OPERATORS

The visual decoding operators introduced in Chapter 5 are demonstrated on a handful of operators⁷⁴ across three chart types.⁷⁵ While this constitutes an existence proof, ultimately the practical value of this approach depends on whether it can scale to a comprehensive library of operators covering a broader range of chart types, tasks, and perceptual operations. Several open questions bear on this scaling challenge.

⁷⁴ that is, projection to axis, projection to curve, peak detection, area bisection

⁷⁵ PDF, CDF, and scatterplot — all three are cartesian graphs

How many operators are there? The current set was identified by decomposing specific visualization tasks into constituent perceptual steps. But the space of possible tasks and chart types is large, and it is unclear whether the operator vocabulary is finite and enumerable or open-ended. Identifying the right *granularity*, finding operators that are fine-grained enough to be reusable across contexts but coarse-grained enough to be independently characterizable, is an important next-step empirical question.

How should new operators be discovered? The current approach relies on top-down task decomposition by the researcher. An alternative is bottom-up discovery from behavioral data: if viewers' response patterns on a new task are not well predicted by existing operator compositions, the residual structure may reveal a missing operator. Eye-tracking data could also inform operator discovery, since distinct operators should correspond to distinct gaze patterns.

How can this be combined with VLMs?⁷⁶ While frontier VLMs do well on aggregate Chart-QA leaderboards, they perform badly on a small, well-defined set of conditional pathologies.⁷⁷ With a library of visual decoding operators that gives good coverage, multimodal LLMs can break questions into combinations of perceptual primitives and reason-

⁷⁶ i.e., vision language models

⁷⁷ See [this blog post](#).

ing operators that correspond to operators, thus allowing predictions that compose, which can then be part of a chain-of-thought reasoning process.

6.2.2 ATTENTION, SALIENCE, AND THE TOOLS TO CAPTURE

THEM

Across the visualization literature, **attention** and **salience** are frequently invoked and measured, yet discussions about the constructs themselves are rare.⁷⁸ The two terms are routinely conflated: *salience* typically denotes bottom-up, stimulus-driven conspicuity,⁷⁹ whereas *attention* to a chart is typically treated as top-down, shaped by the task, by learned conventions for reading axes and legends, and by expectations about where information lives. Simply borrowing a bottom-up saliency model to explain where a reader looks therefore risks poor construct validity: it predicts what would pop out of an unfamiliar image, not what a person who knows they are reading a scatterplot actually samples.

The problem compounds because salience in practice is often *defined* by what attracts gaze and then used to *explain* gaze — a circularity that no quantity of additional data resolves. And gaze itself is only a proxy: eye tracking captures overt attention through fixation, but fixation and attention dissociate in both directions, since one can fixate without attending and attend covertly without fixating [138]. A program that wants to treat attention or salience as part of a perceptual operator in the sense of Chapter 5 cannot do so until it has principled sense of what each is.

This makes the choice of measurement instrument more consequential than it first appears. The available tools are usually arranged along a single frontier of cost against fidelity — from think-aloud and mouse-con-

⁷⁸ Given the amount of debate going on in psychology, ranging from papers that claim we know what attention is [196] and papers that claim that we don't know what attention is [72], I think the questions of “what is attention” and “how is it different than salience” are hardly settled.

⁷⁹ e.g., the feature contrasts in color, luminance, orientation, and size that classical saliency models predict [83]

tingent proxies such as BubbleView [93], through in-browser webcam estimation [131], to remote consumer trackers, and finally to research-grade systems at the precise and expensive end. It would be informative and interesting to characterize this tradeoff between equipment cost and fidelity of results.

6.2.3 IS READING A GRAPH SIMILAR TO READING A PARAGRAPH?

The title of Boger and Franconeri’s recent paper, “Reading a graph is like reading a paragraph” [8], is one I find myself returning to. I think this is partly because this title crystallizes a mantra that has circulated in the graph comprehension literature since Carpenter and Shah, and partly because it echoes the field’s earliest formal model: Pinker’s theory of graph comprehension [136], which proposed that viewers translate a graph’s marks into a sentence-like visual description interpreted through a learned graph schema.⁸⁰

⁸⁰Pinker’s work draws heavily on propositional theories of visual representation and on schema theory.

Is reading a graph similar to reading a paragraph though? The results in Boger and Franconeri [8] make sense but what then would ensemble perception be? And is there something in between? When does serial processing turn into parallel processing and vice versa?

6.3 LIMITATIONS

Several limitations apply across the dissertation.

6.3.1 SCOPE OF CHART TYPES

The models in this dissertation cover bar charts, scatterplots, PDF and CDF plots, lineup displays, and a set of basic visual encodings (color, shape, size). Many common and potentially more visually interesting chart types, such as heatmaps, node-link diagrams, maps, and polar coordinates, are unfortunately not addressed. Whether the operator framework extends to these chart types is plausible but remains to be

demonstrated. Some chart types may require operators not yet in the library (e.g., tracing a path through a node-link diagram, judging color intensity in a heatmap), while others may use the same operators in different configurations.

6.3.2 ECOLOGICAL VALIDITY

All three chapters use controlled tasks with relatively simple stimuli. Real-world visualization use often involves richer contexts, including but not limited to multiple coordinated views, interactive exploration, domain expertise, time pressure, and annotation. Whether and how the models developed in this thesis extend to more naturalistic settings is an interesting open question.

6.3.3 TASKS AND SUBJECTIVITY EMBEDDED WITHIN

This thesis rests on human-subject experiments, and measuring performance at all requires operationalizing it through tasks, which are unavoidably somewhat arbitrary. Despite everything I argued above, I have come to see tasks as something of a necessary evil: empirical research cannot proceed without them. That said, tasks are subject to the same threat to construct validity I raised previously on attention and salience — we are rarely certain that the task we administer actually measures the construct we intend.⁸¹ Related, it is impossible to know whether the way we decomposed the visualization comprehension process is the *right* one.⁸² Every hypothesis carries its own liabilities; the subjectivity in my choice of decomposition is one I am willing to accept.

⁸¹ Not to mention that the act of measuring might change what is being measured. Similar idea appears across disciplines, e.g., the Observer Effect, the Hawthorne Effect, Goodhart’s Law, to name a few.

⁸² This recalls the Platonic question of whether there is such a thing as a true chair — an ideal Form the instances only approximate. If we cannot know whether “true” operators exist in that sense, then perhaps the right question is not whether a decomposition is *true* but whether it is *useful*, and acknowledge the subjectivity embedded within.

7 CONCLUSION

This dissertation developed computational models of visualization perception across three studies. Chapter 3 introduced a formal model of y-axis truncation that identifies task-dependent conditions under which truncation preserves or distorts data structure. Chapter 4 tested the transfer of deep-feature similarity metrics from natural images to information visualization, finding that these computational proxies succeed for complex visualizations but fail for basic visual encodings. Chapter 5 proposed visual decoding operators, composable perceptual primitives with quantifiable error profiles, and provided an existence proof that operators characterized on one set of charts and tasks compose to predict performance on structurally different ones.

Together, these studies demonstrate that computational models of visualization perception that *generalize beyond the conditions they were fit in* are both feasible and productive. They predict quantities that ordinal rankings and categorical taxonomies cannot, they expose mechanisms that simple accuracy measures obscure, and, in the case of visual decoding operators, they generalize to contexts not used to fit them.

REFERENCES

1. James F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26, 11: 832–843. <https://doi.org/10.1145/182.358434>
2. Robert Amar, James Eagan, and John Stasko. 2005. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, 111–117. <https://doi.org/https://doi.org/10.1109/INFOVIS.2005.24>
3. Kenneth J Arrow. 2012. *Social choice and individual values*. Yale university press.
4. Ioanna Maria Attarian, Brett D Roads, and Michael Curtis Mozer. 2020. Transforming neural network visual representations to predict human judgments of similarity. In *NeurIPS 2020 Workshop SVRHM*. <https://doi.org/10.48550/arXiv.2010.06512>
5. Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. 2018. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology* 14, 12: e1006613. <https://doi.org/10.1371/journal.pcbi.1006613>
6. Carl Bergstrom and Jevin West. 2017. Calling bullshit: Proportional ink. Retrieved from https://callingbullshit.org/tools/tools_proportional_ink.html
7. Jacques Bertin. 1983. *Semiology of graphics*. University of Wisconsin press.
8. Tal Boger and Steven Franconeri. 2024. Reading a Graph Is like Reading a Paragraph. *Journal of Experimental Psychology: General*. <https://doi.org/10.1037/xge0001604>
9. Michelle A Borkin, Azalea A Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What Makes a Visualization Memorable?. *IEEE transactions on visualization and computer graphics* 19, 12: 2306–2315. <https://doi.org/10.1109/TVCG.2013.234>

10. Michelle A. Borkin, Zoya Bylinskii, Nam Wook Kim, Constance May Bainbridge, Chelsea S. Yeh, Daniel Borkin, Hanspeter Pfister, and Aude Oliva. 2016. Beyond Memorability: Visualization Recognition and Recall. *IEEE Transactions on Visualization and Computer Graphics* 22, 1: 519–528. <https://doi.org/10.1109/TVCG.2015.2467732>
11. Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D³: Data-Driven Documents. *IEEE transactions on visualization and computer graphics* 17, 12: 2301–2309. <https://doi.org/10.1109/TVCG.2011.185>
12. Jeffrey S Bowers, Gaurav Malhotra, Marin Dujmović, Milton Llera Montero, Christian Tsvetkov, Valerio Biscione, Guillermo Puebla, Federico Adolphi, John E Hummel, Rachel F Heaton, and others. 2023. Deep problems with neural network models of human vision. *Behavioral and Brain Sciences* 46: e385. <https://doi.org/10.1017/s0140525x22002813>
13. Willard Cope Brinton. 1939. *Graphic presentation*.
14. Andreas Buja, Dianne Cook, Heike Hofmann, Michael Lawrence, Eun-Kyung Lee, Deborah F Swayne, and Hadley Wickham. 2009. Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 367, 1906: 4361–4383. <https://doi.org/10.1098/rsta.2009.0120>
15. Paul-Christian Bürkner. 2017. brms: An R Package for Bayesian Multilevel Models Using Stan. *Journal of Statistical Software* 80, 1: 1–28. <https://doi.org/10.18637/jss.v080.i01>
16. Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. 2011. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011*, 97–104. <https://doi.org/10.1109/CVPR.2011.5995413>
17. Alberto Cairo. 2019. *How charts lie: Getting smarter about visual information*. WW Norton & Company.
18. Liang Chen, Weichu Xie, Yiyang Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Yiping Bao, and others. 2026. BabyVision: Visual Reasoning Beyond Language. *arXiv preprint arXiv:2601.06521*. <https://doi.org/https://doi.org/10.48550/arXiv.2601.06521>
19. Qing Chen, Shixiong Cao, Jiazhe Wang, and Nan Cao. 2024. How Does Automation Shape the Process of Narrative Visualization: A Survey of Tools. *IEEE Transactions on Visualization and Computer Graphics* 30, 8: 4429–4448. <https://doi.org/10.1109/TVCG.2023.3261320>

20. Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*.
21. Julien Chiquet, Guillem Rigau, Martina Sundqvist, Valentin Dervieux, and Florent Bersani. 2020. Package ‘aricode’. *R package version*. <https://doi.org/10.32614/CRAN.package.aricode>
22. CIE. 1978. Recommendations on uniform color spaces, color difference equations, psychometric color terms. *CIE Publication 15*: 9–12. <https://doi.org/10.1002/j.1520-6378.1977.tb00102.x>
23. William S Cleveland and Robert McGill. 1984. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association* 79, 387: 531–554. <https://doi.org/10.2307/2288400>
24. William S. Cleveland and Robert McGill. 1987. Graphical Perception: The Visual Decoding of Quantitative Information on Graphical Displays of Data. *Journal of the Royal Statistical Society: Series A (General)* 150, 3: 192–210. <https://doi.org/10.2307/2981473>
25. William S. Cleveland, Marylyn E. McGill, and Robert McGill. 1988. The Shape Parameter of a Two-Variable Graph. *Journal of the American Statistical Association* 83, 402: 289–300. <https://doi.org/10.1080/01621459.1988.10478598>
26. Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 215–223.
27. Michael Correll, Enrico Bertini, and Steven Franconeri. 2020. Truncating the Y-Axis: Threat or Menace?. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3313831.3376222>
28. Michael Correll, Mingwei Li, Gordon Kindlmann, and Carlos Scheidegger. 2019. Looks Good To Me: Visualizations As Sanity Checks. *IEEE Transactions on Visualization and Computer Graphics* 25, 1: 830–839. <https://doi.org/10.1109/TVCG.2018.2864907>
29. Michael Correll. 2020. Truncating the Y-Axis: Threat or Menace?. Retrieved December 5, 2023 from <https://engineering.tableau.com/truncating-the-y-axis-threat-or-menace-d0bce66d4d08>
30. Zach Cutler, Jack Wilburn, Hilson Shrestha, Yiren Ding, Brian Bollen, Khandaker Abrar Nadib, Tingying He, Andrew McNutt, Lane Harrison, and Alexander Lex. 2026. ReVISit 2: A Full Experiment Life Cycle User Study Framework. *IEEE Transactions on Visualization and Computer Graphics*. <https://doi.org/10.1109/TVCG.2025.3633896>

31. Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. 2015. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM multimedia systems conference*, 219–224. <https://doi.org/10.1145/2713168.2713194>
32. Aritra Dasgupta and Robert Kosara. 2010. Pargnostics: Screen-Space Metrics for Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics* 16, 6: 1017–1026. <https://doi.org/10.1109/TVCG.2010.184>
33. Russell Davis, Xiaoying Pu, Yiren Ding, Brian D Hall, Karen Bonilla, Mi Feng, Matthew Kay, and Lane Harrison. 2022. The Risks of Ranking: Revisiting Graphical Perception to Model Individual Differences in Visualization Performance. *IEEE transactions on visualization and computer graphics*: 1–16. <https://doi.org/10.1109/TVCG.2022.3226463>
34. Çağatay Demiralp, Michael S Bernstein, and Jeffrey Heer. 2014. Learning perceptual kernels for visualization design. *IEEE transactions on visualization and computer graphics* 20, 12: 1933–1942. <https://doi.org/10.1109/TVCG.2014.2346978>
35. Çağatay Demiralp, Carlos E. Scheidegger, Gordon L. Kindlmann, David H. Laidlaw, and Jeffrey Heer. 2014. Visual Embedding: A Model for Visualization. *IEEE Computer Graphics and Applications* 34, 1: 10–15. <https://doi.org/10.1109/MCG.2014.18>
36. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
37. Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. 2020. Image Quality Assessment: Unifying Structure and Texture Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*: 1. <https://doi.org/10.1109/TPAMI.2020.3045810>
38. Yiren Ding, Jack Wilburn, Hilson Shrestha, Akim Ndlovu, Kiran Gadhave, Carolina Nobre, Alexander Lex, and Lane Harrison. 2023. reVISit: Supporting Scalable Evaluation of Interactive Visualizations. In *2023 IEEE Visualization and Visual Analytics (VIS)*, 31–35. <https://doi.org/10.1109/VIS54172.2023.00015>
39. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Retrieved from <https://arxiv.org/abs/2010.11929>

40. Madison A Elliott, Christine Nothelfer, Cindy Xiong, and Danielle Albers Szafr. 2021. A Design Space of Vision Science Methods for Visualization Research. *IEEE transactions on visualization and computer graphics* 27, 2: 1117–1127. <https://doi.org/10.1109/TVCG.2020.3029413>
41. Wilfried Elmenreich. 2002. An introduction to sensor fusion. *Vienna University of Technology, Austria* 502, 1–28: 37.
42. Ahmet M Eskicioglu and Paul S Fisher. 1995. Image quality measures and their performance. *IEEE Transactions on communications* 43, 12: 2959–2965. <https://doi.org/10.1109/26.477498>
43. Stephen Few. 2004. Show me the numbers. *Analytics Pres.*
44. Martin Fink, Jan-Henrik Haunert, Joachim Spoerhase, and Alexander Wolff. 2013. Selecting the Aspect Ratio of a Scatter Plot Based on Its Delaunay Triangulation. *IEEE Transactions on Visualization and Computer Graphics* 19, 12: 2326–2335. <https://doi.org/10.1109/TVCG.2013.187>
45. François Fleuret, Ting Li, Charles Dubout, Emma K Wampler, Steven Yantis, and Donald Geman. 2011. Comparing machines and humans on a visual categorization test. *Proceedings of the National Academy of Sciences* 108, 43: 17621–17625. <https://doi.org/10.1073/pnas.1109168108>
46. Amy Rae Fox. 2023. Theories and models in graph comprehension. *Visualization Psychology*: 39–64.
47. Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. 2023. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*. <https://doi.org/10.48550/arXiv.2306.09344>
48. Johannes Fuchs, Petra Isenberg, Anastasia Bezerianos, and Daniel Keim. 2017. A Systematic Review of Experimental Studies on Data Glyphs. *IEEE Transactions on Visualization and Computer Graphics* 23, 7: 1863–1879. <https://doi.org/10.1109/TVCG.2016.2549018>
49. Jonah Gabry and Tristan Mahr. 2025. bayesplot: Plotting for Bayesian Models. <https://doi.org/10.32614/CRAN.package.bayesplot>
50. Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. 2019. Visualization in Bayesian workflow. *Journal of the Royal Statistical Society Series A: Statistics in Society* 182, 2: 389–402. <https://doi.org/https://doi.org/10.1111/rssa.12378>
51. Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A Neural Algorithm of Artistic Style. <https://doi.org/10.48550/arXiv.1508.06576>
52. Robert Geirhos, Kristof Meding, and Felix A. Wichmann. 2020. Beyond Accuracy: Quantifying Trial-by-Trial Behaviour of CNNs and Humans by Measuring Error Consistency. In *Advances*

- in *Neural Information Processing Systems*, 13890–13902. Retrieved January 29, 2024 from <https://proceedings.neurips.cc/paper/2020/hash/9f6992966d4c363ea0162a056cb45fe5-Abstract.html>
53. Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. 2018. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*. <https://doi.org/10.48550/arXiv.1811.12231>
 54. Andrew Gelman and Jennifer Hill. 2007. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.
 55. Andrew Gelman, Jennifer Hill, and Masanao Yajima. 2012. Why We (Usually) Don't Have to Worry About Multiple Comparisons. *Journal of Research on Educational Effectiveness* 5, 2: 189–211. <https://doi.org/10.1080/19345747.2011.618213>
 56. Douglas J Gillan and Robert Lewis. 1994. A componential model of human interaction with graphs: 1. Linear regression modeling. *Human Factors* 36, 3: 419–440. <https://doi.org/10.1177/001872089403600303>
 57. Bernd Girod. 1993. What's wrong with mean-squared error?. In *Digital images and human vision*. 207–220. <https://doi.org/10.1109/MDSP.1991.639240>
 58. Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, and Anastasia Bezerianos. 2018. Comparing similarity perception in time series visualizations. *IEEE transactions on visualization and computer graphics* 25, 1: 523–533. <https://doi.org/10.1109/TVCG.2018.2865077>
 59. Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*. <https://doi.org/10.48550/arXiv.1412.6572>
 60. Shreyank N Gowda and Chun Yuan. 2019. ColorNet: Investigating the importance of color spaces for image classification. In *Computer Vision—ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part IV* 14, 581–596. https://doi.org/10.1007/978-3-030-20870-7_36
 61. Guo-Dong Guo, Anil K Jain, Wei-Ying Ma, and Hong-Jiang Zhang. 2002. Learning similarity measure for natural image retrieval with relevance feedback. *IEEE Transactions on Neural Networks* 13, 4: 811–820. <https://doi.org/10.1109/TNN.2002.1021882>
 62. Daniel Haehn, James Tompkin, and Hanspeter Pfister. 2019. Evaluating ‘Graphical Perception’ with CNNs. *IEEE Transactions on Visualization and Computer Graphics* 25, 1: 641–650. <https://doi.org/10.1109/TVCG.2018.2865138>

63. Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825: 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
64. Lane Harrison, Fumeng Yang, Steven Franconeri, and Remco Chang. 2014. Ranking visualizations of correlation using weber's law. *IEEE Transactions on Visualization and Computer Graphics* 20, 12: 1943–1952. <https://doi.org/10.1109/TVCG.2014.2346979>
65. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
66. Martin N Hebart, Adam H Dickter, Alexis Kidder, Wan Y Kwok, Anna Corriveau, Caitlin Van Wicklin, and Chris I Baker. 2019. THINGS: A database of 1,854 object concepts and more than 26,000 naturalistic object images. *PloS one* 14, 10: e0223792. <https://doi.org/10.1371/journal.pone.0223792>
67. Martin N Hebart, Charles Y Zheng, Francisco Pereira, and Chris I Baker. 2020. Revealing the multi-dimensional mental representations of natural objects underlying human similarity judgements. *Nature human behaviour* 4, 11: 1173–1185. <https://doi.org/10.1038/s41562-020-00951-3>
68. Jeffrey Heer and Maneesh Agrawala. 2006. Multi-Scale Banking to 45 Degrees. *IEEE transactions on visualization and computer graphics* 12, 5: 701–708. <https://doi.org/10.1109/TVCG.2006.163>
69. Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 203–212. <https://doi.org/10.1145/1753326.1753357>
70. Michael Correll AND Jeffrey Heer. 2017. Black Hat Visualization. In *Workshop on Dealing with Cognitive Biases in Visualisations (DECISive)*, IEEE VIS. Retrieved from <http://idl.cs.washington.edu/papers/blackhatvis>
71. Mary Hegarty. 2011. The cognitive science of visual-spatial displays: Implications for design. *Topics in cognitive science* 3, 3: 446–474. <https://doi.org/https://doi.org/10.1111/j.1756-8765.2011.01150.x>
72. Bernhard Hommel, Craig S. Chapman, Paul Cisek, Heather F. Neyedli, Joo-Hyun Song, and Timothy N. Welsh. 2019. No One Knows What Attention Is. *Attention, Perception, & Psychophysics* 81, 7: 2288–2303. <https://doi.org/10.3758/s13414-019-01846-w>

73. Hidehito Honda, Rina Kagawa, and Masaru Shirasuna. 2022. On the Round Number Bias and Wisdom of Crowds in Different Response Formats for Numerical Estimation. *Scientific Reports* 12, 1: 8167. <https://doi.org/10.1038/s41598-022-11900-7>
74. Matt-Heun Hong, Jessica K. Witt, and Danielle Albers Szafr. 2021. The Weighted Average Illusion: Biases in Perceived Mean Position in Scatterplots. <https://doi.org/10.48550/arXiv.2108.03766>
75. Helia Hosseinpour, Laura E Matzen, Kristin M Divis, Spencer C Castro, and Lace Padilla. 2024. Examining Limits of Small Multiples: Frame Quantity Impacts Judgments with Line Graphs. *IEEE Transactions on Visualization and Computer Graphics*.
76. Kevin Hu, Snehal Kumar 'neil' S Gaikwad, Madelon Hulsebos, Michiel A Bakker, Emanuel Zraggen, César Hidalgo, Tim Kraska, Guoliang Li, Arvind Satyanarayan, and Çağatay Demiralp. 2019. VizNet: Towards A Large-Scale Visualization Learning and Benchmarking Repository. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*, 1–12. <https://doi.org/10.1145/3290605.3300892>
77. Oliver Huang and Carolina Nobre. 2025. ViStruct: Simulating Expert-Like Reasoning Through Task Decomposition and Visual Attention. In *2025 IEEE Visualization and Visual Analytics (VIS)*, 216–220. <https://doi.org/10.1109/VIS60296.2025.00049>
78. Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, 1501–1510. <https://doi.org/10.48550/arXiv.1703.06868>
79. Darrell Huff. 2023. *How to lie with statistics*. Penguin UK.
80. Jessica Hullman, Steven Drucker, Nathalie Henry Riche, Bongshin Lee, Danyel Fisher, and Eytan Adar. 2013. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on visualization and computer graphics* 19, 12: 2406–2415. <https://doi.org/10.1109/TVCG.2013.119>
81. Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360*. <https://doi.org/10.48550/arXiv.1602.07360>
82. Ulf Geir Indahl, Tormod Næs, and Kristian Hovde Liland. 2018. A similarity index for comparing coupled matrices. *Journal of Chemometrics* e3049. <https://doi.org/10.1002/cem.3049>
83. Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* 20, 11: 1254–1259.

84. Laurence Jacobs, Charles Keown, Reginald Worthley, and Kyung-Il Ghymn. 1991. Cross-cultural colour comparisons: Global marketers beware!. *International marketing review* 8, 3. <https://doi.org/10.1108/02651339110137279>
85. Nicole Jardine, Brian D Ondov, Niklas Elmqvist, and Steven Franconeri. 2020. The Perceptual Proxies of Visual Comparison. *IEEE transactions on visualization and computer graphics* 26, 1: 1012–1021. <https://doi.org/10.1109/TVCG.2019.2934786>
86. Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II* 14, 694–711. https://doi.org/10.1007/978-3-319-46475-6_43
87. Alex Kale, Matthew Kay, and Jessica Hullman. 2021. Visual Reasoning Strategies for Effect Size Judgments and Decisions. *IEEE transactions on visualization and computer graphics* 27, 2: 272–282. <https://doi.org/10.1109/TVCG.2020.3030335>
88. Alboukadel Kassambara. 2023. ggpubr: 'ggplot2' Based Publication Ready Plots. Retrieved from <https://rpkgs.datanovia.com/ggpubr/>
89. 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 *Proceedings of the 2016 CHI conference on human factors in computing systems*, 4521–4532.
90. Matthew Kay. 2023. tidybayes: Tidy Data and Geoms for Bayesian Models. <https://doi.org/10.5281/zenodo.1308151>
91. Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. 2014. Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS computational biology* 10, 11: e1003915. <https://doi.org/10.1371/journal.pcbi.1003915>
92. Junkyung Kim, Matthew Ricci, and Thomas Serre. 2018. Not-So-CLEVR: learning same–different relations strains feedforward neural networks. *Interface focus* 8, 4: 20180011. <https://doi.org/doi.org/10.1098/rsfs.2018.0011>
93. Nam Wook Kim, Zoya Bylinskii, Michelle A. Borkin, Krzysztof Z. Gajos, Aude Oliva, Fredo Durand, and Hanspeter Pfister. 2017. BubbleView: An Interface for Crowdsourcing Image Importance Maps and Tracking Visual Attention. *ACM Trans. Comput.-Hum. Interact.* 24, 5: 36:1–36:40. <https://doi.org/10.1145/3131275>

94. Younghoon Kim and Jeffrey Heer. 2018. Assessing effects of task and data distribution on the effectiveness of visual encodings. In *Computer Graphics Forum*, 157–167.
95. Younghoon Kim, Kanit Wongsuphasawat, Jessica Hullman, and Jeffrey Heer. 2017. Graphscape: A model for automated reasoning about visualization similarity and sequencing. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, 2628–2638. <https://doi.org/10.1145/3025453.3025866>
96. Gordon Kindlmann and Carlos Scheidegger. 2014. An Algebraic Process for Visualization Design. *IEEE Transactions on Visualization and Computer Graphics* 20, 12: 2181–2190. <https://doi.org/10.1109/TVCG.2014.2346325>
97. Robert Kosara. 2013. Continuous Values and Baselines. Retrieved December 7, 2023 from <https://eagereyes.org/blog/2013/baselines>
98. Stephen M. Kosslyn. 1989. Understanding Charts and Graphs. *Applied Cognitive Psychology* 3, 3: 185–225. <https://doi.org/10.1002/acp.2350030302>
99. Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. 2008. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience* 2: 249. <https://doi.org/10.3389/neuro.06.004.2008>
100. Alex Krizhevsky, Geoffrey Hinton, and others. 2009. Learning multiple layers of features from tiny images. Retrieved from <http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf>
101. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25. <https://doi.org/10.1145/3065386>
102. Jonas Kubilius, Martin Schrimpf, Aran Nayebi, Daniel Bear, Daniel LK Yamins, and James J DiCarlo. 2018. Cornet: Modeling the neural mechanisms of core object recognition. *BioRxiv*: 408385. <https://doi.org/10.1101/408385>
103. Manoj Kumar, Neil Houlsby, Nal Kalchbrenner, and Ekin D. Cubuk. 2022. Do Better ImageNet Classifiers Assess Perceptual Similarity Better?. <https://doi.org/10.48550/arXiv.2203.04946>
104. Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. 2011. Empirical studies in information visualization: Seven scenarios. *IEEE transactions on visualization and computer graphics* 18, 9: 1520–1536. <https://doi.org/10.1109/TVCG.2011.279>

105. Jill H Larkin and Herbert A Simon. 1987. Why a Diagram Is (Sometimes) Worth Ten Thousand Words. *Cognitive science* 11, 1: 65–100. <https://doi.org/10.1111/j.1551-6708.1987.tb00863.x>
106. Eric C Larson and Damon M Chandler. 2010. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of electronic imaging* 19, 1: 11006. <https://doi.org/10.1117/1.3267105>
107. Thai Le, Cecilia Aragon, Hilaire J Thompson, and George Demiris. 2014. Elementary graphical perception for older adults: a comparison with the general population. *Perception* 43, 11: 1249–1260.
108. Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. 2006. Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, 1–5. <https://doi.org/https://doi.org/10.1145/1168149.1168168>
109. Qisheng Li, Sung Jun Joo, Jason D. Yeatman, and Katharina Reinecke. 2020. Controlling for Participants' Viewing Distance in Large-Scale, Psychophysical Online Experiments Using a Virtual Chinrest. *Scientific Reports* 10, 1: 904. <https://doi.org/10.1038/s41598-019-57204-1>
110. Zhimin Li, Haichao Miao, Xinyuan Yan, Valerio Pascucci, Matthew Berger, and Shusen Liu. 2025. See or recall: A sanity check for the role of vision in solving visualization question answer tasks with multimodal llms. *arXiv preprint arXiv:2504.09809*. <https://doi.org/https://doi.org/10.48550/arXiv.2504.09809>
111. Gerald Lee Lohse. 1993. A Cognitive Model for Understanding Graphical Perception. *Human-Computer Interaction* 8, 4: 353–388. https://doi.org/10.1207/s15327051hci0804_3
112. Sheng Long and Matthew Kay. 2024. To cut or not to cut? a systematic exploration of y-axis truncation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 1–12.
113. Sheng Long, Angelos Chatzimpampas, Emma Alexander, Matthew Kay, and Jessica Hullman. 2025. Seeing Eye to AI? Applying Deep-Feature-Based Similarity Metrics to Information Visualization. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–20.
114. Yuxin Ma, Anthony KH Tung, Wei Wang, Xiang Gao, Zhigeng Pan, and Wei Chen. 2018. Scatternet: A deep subjective similarity model for visual analysis of scatterplots. *IEEE transactions on visualization and computer graphics* 26, 3: 1562–1576. <https://doi.org/10.1109/TVCG.2018.2875702>
115. Jock Mackinlay. 1986. Automating the Design of Graphical Presentations of Relational Information. *ACM transactions on graphics* 5, 2: 110–141. <https://doi.org/10.1145/22949.22950>

116. David Marr. 2010. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press. <https://doi.org/https://doi.org/10.7551/mitpress/9780262514620.001.0001>
117. George Mather. 2016. *Foundations of sensation and perception*. Psychology Press.
118. Richard McElreath. 2020. Statistical Rethinking: A Bayesian Course with Examples in R and STAN.
119. Douglas L. Medin, Robert L. Goldstone, and Dedre Gentner. 1993. Respects for Similarity. *Psychological Review* 100, 2: 254–278. <https://doi.org/10.1037/0033-295X.100.2.254>
120. Dominik Moritz, Lace M Padilla, Francis Nguyen, and Steven L Franconeri. 2023. Average estimates in line graphs are biased toward areas of higher variability. *IEEE Transactions on Visualization and Computer Graphics* 30, 1: 306–315. <https://doi.org/10.1109/TVCG.2023.3326589>
121. Dominik Moritz, Chenglong Wang, Greg L Nelson, Halden Lin, Adam M Smith, Bill Howe, and Jeffrey Heer. 2018. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE transactions on visualization and computer graphics* 25, 1: 438–448.
122. Tamara Munzner. 2014. *Visualization analysis and design*. CRC press.
123. Carolina Nobre, Kehang Zhu, Eric Mörtz, Hanspeter Pfister, and Johanna Beyer. 2024. Reading between the pixels: Investigating the barriers to visualization literacy. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 1–17. <https://doi.org/https://doi.org/10.1145/3613904.3642760>
124. Brian D. Ondov, Fumeng Yang, Matthew Kay, Niklas Elmqvist, and Steven Franconeri. 2021. Revealing Perceptual Proxies with Adversarial Examples. *IEEE Transactions on Visualization and Computer Graphics* 27, 2: 1073–1083. <https://doi.org/10.1109/TVCG.2020.3030429>
125. Jeroen Ooms. 2014. The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. *arXiv:1403.2805 [stat.CO]*. <https://doi.org/10.32614/CRAN.package.jsonlite>
126. Lace M Padilla, Sarah H Creem-Regehr, Mary Hegarty, and Jeanine K Stefanucci. 2018. Decision making with visualizations: a cognitive framework across disciplines. *Cognitive research: principles and implications* 3, 1: 29. <https://doi.org/https://doi.org/10.1186/s41235-018-0120-9>
127. Liudas Panavas, Amy E Worth, Tarik Crnovrsanin, Tejas Sathyamurthi, Sara Cordes, Michelle A Borkin, and Cody Dunne. 2022. Juvenile graphical perception: A comparison between children and adults. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–14.

128. Aditeya Pandey, Uzma Haque Syeda, and Michelle A Borkin. Towards Identification and Mitigation of Task-Based Challenges in Comparative Visualization Studies.
129. Anshul Vikram Pandey, Josua Krause, Cristian Felix, Jeremy Boy, and Enrico Bertini. 2016. Towards understanding human similarity perception in the analysis of large sets of scatter plots. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 3659–3669. <https://doi.org/10.1145/2858036.2858155>
130. Anshul Vikram Pandey, Katharina Rall, Margaret L Satterthwaite, Oded Nov, and Enrico Bertini. 2015. How Deceptive Are Deceptive Visualizations? An Empirical Analysis of Common Distortion Techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 1469–1478. <https://doi.org/10.1145/2702123.2702608>
131. Alexandra Papoutsaki. 2015. Scalable webcam eye tracking by learning from user interactions. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, 219–222.
132. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32. <https://doi.org/10.48550/arXiv.1912.01703>
133. David Peebles and Peter C-H Cheng. 2003. Modeling the effect of task and graphical representation on response latency in a graph reading task. *Human Factors* 45, 1: 28–46. <https://doi.org/10.1518/hfes.45.1.28.27225>
134. David Peebles and Peter C. -H Cheng. 2002. Extending Task Analytic Models of Graph-Based Reasoning: A Cognitive Model of Problem Solving with Cartesian Graphs in ACT-R/PM. *Cognitive Systems Research* 3, 1: 77–86. [https://doi.org/10.1016/S1389-0417\(01\)00046-8](https://doi.org/10.1016/S1389-0417(01)00046-8)
135. Gustav Grund Pihlgren, Konstantina Nikolaidou, Prakash Chandra Chhipa, Nosheen Abid, Rajkumar Saini, Fredrik Sandin, and Marcus Liwicki. 2023. A Systematic Performance Analysis of Deep Perceptual Loss Networks: Breaking Transfer Learning Conventions. *arXiv preprint arXiv:2302.04032*. <https://doi.org/10.48550/arXiv.2302.04032>
136. Steven Pinker. 1990. A theory of graph comprehension. In *Artificial intelligence and the future of testing*, Roy Freedle (ed.). Psychology Press, Hillsdale, NJ, US, 73–126.
137. Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and others. 2015. Image database

- TID2013: Peculiarities, results and perspectives. *Signal processing: Image communication* 30: 57–77. <https://doi.org/10.1016/j.image.2014.10.009>
138. Michael I Posner. 1980. Orienting of attention. *Quarterly journal of experimental psychology* 32, 1: 3–25.
 139. Ekta Prashnani, Hong Cai, Yasamin Mostofi, and Pradeep Sen. 2018. PieAPP: Perceptual Image-Error Assessment through Pairwise Preference. <https://doi.org/10.48550/arXiv.1806.02067>
 140. Rifat Ara Proma, Michael Correll, Ghulam Jilani Quadri, and Paul Rosen. 2024. Visual Stenography: Feature Recreation and Preservation in Sketches of Line Charts. Retrieved from https://ieevis.org/year/2024/program/poster_v-vis-posters-1100.html
 141. Rifat Ara Proma, Michael Correll, Ghulam Jilani Quadri, and Paul Rosen. 2025. Visual Stenography: Feature Recreation and Preservation in Sketches of Noisy Line Charts. *IEEE Transactions on Visualization and Computer Graphics*: 1–16. <https://doi.org/10.1109/TVCG.2025.3626128>
 142. Zening Qu and Jessica Hullman. 2016. Evaluating visualization sets: Trade-offs between local effectiveness and global consistency. In *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization*, 44–52. <https://doi.org/10.1145/2993901.2993910>
 143. Zening Qu and Jessica Hullman. 2017. Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE transactions on visualization and computer graphics* 24, 1: 468–477. <https://doi.org/10.1109/TVCG.2017.2744198>
 144. Ghulam Jilani Quadri and Paul Rosen. 2022. A Survey of Perception-Based Visualization Studies by Task. *IEEE Transactions on Visualization and Computer Graphics* 28, 12: 5026–5048. <https://doi.org/10.1109/TVCG.2021.3098240>
 145. Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. 2024. Vision language models are blind: Failing to translate detailed visual features into words. *arXiv preprint arXiv:2407.06581*. <https://doi.org/https://doi.org/10.48550/arXiv.2407.06581>
 146. Ronald A Rensink. 2021. Visualization as a stimulus domain for vision science. *Journal of Vision* 21, 8: 3.
 147. Jacob Ritchie, Daniel Wigdor, and Fanny Chevalier. 2019. A Lie Reveals the Truth: Quasimodes for Task-Aligned Data Presentation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13. <https://doi.org/10.1145/3290605.3300423>

148. Brett D Roads and Bradley C Love. 2021. Enriching imagenet with human similarity judgments and psychological embeddings. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 3547–3557. <https://doi.org/10.48550/arXiv.2011.11015>
149. Brett D Roads and Bradley C Love. 2024. Modeling similarity and psychological space. *Annual Review of Psychology* 75, 1: 215–240. <https://doi.org/10.1146/annurev-psych-040323-115131>
150. Brett D Roads and Michael C Mozer. 2017. Improving human-machine cooperative classification via cognitive theories of similarity. *Cognitive science* 41, 5: 1394–1411. <https://doi.org/10.1111/cogs.12400>
151. Iris van Rooij, Berna Devezer, Joshua Skewes, Sashank Varma, and Todd Wareham. 2024. What makes a good theory? Interdisciplinary perspectives. *Computational Brain & Behavior* 7, 4: 503–507. <https://doi.org/https://doi.org/10.1007/s42113-024-00225-5>
152. Teemu Säilynoja, Andrew R Johnson, Osvaldo A Martin, and Aki Vehtari. 2025. Recommendations for visual predictive checks in Bayesian workflow. *arXiv preprint arXiv:2503.01509*. <https://doi.org/https://doi.org/10.48550/arXiv.2503.01509>
153. Bahador Saket, Alex Endert, and Cagatay Demiralp. 2019. Task-Based Effectiveness of Basic Visualizations. *IEEE transactions on visualization and computer graphics* 25, 7: 2505–2512. <https://doi.org/10.1109/TVCG.2018.2829750>
154. Mehul P Sampat, Zhou Wang, Shalini Gupta, Alan Conrad Bovik, and Mia K Markey. 2009. Complex wavelet structural similarity: A new image similarity index. *IEEE transactions on image processing* 18, 11: 2385–2401. <https://doi.org/10.1109/TIP.2009.2025923>
155. Abhraneel Sarma, Sheng Long, Michael Correll, and Matthew Kay. 2024. Tasks and Telephones: threats to experimental validity due to misunderstandings of visualisation tasks and strategies position paper. In *2024 IEEE Evaluation and Beyond-Methodological Approaches for Visualization (BELIV)*, 33–40. <https://doi.org/10.1109/BELIV64461.2024.00009>
156. Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2016. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics* 23, 1: 341–350. <https://doi.org/https://doi.org/10.1109/TVCG.2016.2599030>
157. Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, and others. 2018. Brain-score: Which artificial neural network for object recognition is most brain-like?. *BioRxiv*: 407007. <https://doi.org/10.1101/407007>

158. Hamid R Sheikh. 2003. Image and video quality assessment research at LIVE. Retrieved from <http://live.ece.utexas.edu/research/quality/>
159. Danqing Shi, Yao Wang, Yunpeng Bai, Andreas Bulling, and Antti Oulasvirta. 2025. Chartist: Task-driven Eye Movement Control for Chart Reading. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–14. <https://doi.org/https://doi.org/10.1145/3706598.3713128>
160. David Simkin and Reid Hastie. 1987. An information-processing analysis of graph perception. *Journal of the American Statistical Association* 82, 398: 454–465. <https://doi.org/10.1080/01621459.1987.10478448>
161. Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. <https://doi.org/10.48550/arXiv.1409.1556>
162. Chad Skelton. 2018. Bar charts should always start at zero. but what about line charts?. Retrieved from <http://www.chadskelton.com/2018/06/bar-charts-should-always-start-at-zero.html>
163. Stan Development Team. 2023. RStan: the R interface to Stan. Retrieved from <https://mc-stan.org/>
164. The Stdlib Authors. 2016. stdlib. Retrieved from <https://stdlib.io/>
165. Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C Love, Erin Grant, Jascha Achterberg, Joshua B Tenenbaum, and others. 2023. Getting aligned on representational alignment. *arXiv preprint arXiv:2310.13018*. <https://doi.org/10.48550/arXiv.2310.13018>
166. Danielle Albers Szafir, Steve Haroz, Michael Gleicher, and Steven Franconeri. 2016. Four Types of Ensemble Coding in Data Visualizations. *Journal of Vision* 16. <https://doi.org/10.1167/16.5.11>
167. Justin Talbot, John Gerth, and Pat Hanrahan. 2011. Arc Length-Based Aspect Ratio Selection. *IEEE Transactions on Visualization and Computer Graphics* 17, 12: 2276–2282. <https://doi.org/10.1109/TVCG.2011.167>
168. Justin Talbot, John Gerth, and Pat Hanrahan. 2012. An empirical model of slope ratio comparisons. *IEEE Transactions on Visualization and Computer Graphics* 18, 12: 2613–2620.
169. Justin Talbot, Vidya Setlur, and Anushka Anand. 2014. Four Experiments on the Perception of Bar Charts. *IEEE transactions on visualization and computer graphics* 20, 12: 2152–2160. <https://doi.org/10.1109/TVCG.2014.2346320>

170. Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114. <https://doi.org/10.48550/arXiv.1905.11946>
171. Panayiotis Theodossiou. 1998. Financial Data and the Skewed Generalized T Distribution. *Management Science* 44, 12: 1650–1661. Retrieved from <https://www.jstor.org/stable/2634700>
172. Warren S Torgerson. 1965. Multidimensional scaling of similarity. *Psychometrika* 30, 4: 379–393. <https://doi.org/10.1007/BF02289530>
173. Edward R Tufte. 2001. *The visual display of quantitative information*. Graphics press Cheshire, CT.
174. Amos Tversky. 1977. Features of similarity. *Psychological review* 84, 4: 327. <https://doi.org/10.1037/0033-295X.84.4.327>
175. Tomer Ullman. 2024. The illusion-illusion: Vision language models see illusions where there are none. *arXiv preprint arXiv:2412.18613*. <https://doi.org/https://doi.org/10.48550/arXiv.2412.18613>
176. Kevin Ushey, JJ Allaire, and Yuan Tang. 2024. reticulate: Interface to 'Python'. Retrieved from <https://rstudio.github.io/reticulate/>
177. Matthew Varona, Maryam Hedayati, Matthew Kay, and Carolina Nobre. 2025. Theory is Shapes. In *Proceedings of alt.vis 2025*. <https://doi.org/10.48550/arXiv.2510.01382>
178. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30.
179. Rafael Veras and Christopher Collins. 2019. Discriminability tests for visualization effectiveness and scalability. *IEEE transactions on visualization and computer graphics* 26, 1: 749–758. <https://doi.org/10.1109/TVCG.2019.2934432>
180. Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17: 261–272. <https://doi.org/10.1038/s41592-019-0686-2>

181. Catherine Wah, Subhransu Maji, and Serge Belongie. 2015. Learning Localized Perceptual Similarity Metrics for Interactive Categorization. In *2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 502–509. <https://doi.org/10.1109/WACV.2015.73>
182. Huichen Will Wang, Mitchell Gordon, Leilani Battle, and Jeffrey Heer. 2024. Dracogpt: Extracting visualization design preferences from large language models. *IEEE Transactions on Visualization and Computer Graphics* 31, 1: 710–720. <https://doi.org/https://doi.org/10.1109/TVCG.2024.3456350>
183. Huichen Will Wang, Jane Hoffswell, Sao Myat Thazin Thane, Victor S. Bursztyn, and Cindy Xiong Bearfield. 2025. How Aligned are Human Chart Takeaways and LLM Predictions? A Case Study on Bar Charts with Varying Layouts. *IEEE Transactions on Visualization and Computer Graphics* 31, 1: 536–546. <https://doi.org/10.1109/TVCG.2024.3456378>
184. Jian Wang, Luca Pagani, Richard K Leach, Wenhan Zeng, Bianca M Colosimo, and Liping Zhou. 2017. Study of weighted fusion methods for the measurement of surface geometry. *Precision Engineering* 47: 111–121. <https://doi.org/https://doi.org/10.1016/j.precisioneng.2016.07.012>
185. Yunhai Wang, Zeyu Wang, Lifeng Zhu, Jian Zhang, Chi-Wing Fu, Zhanglin Cheng, Changhe Tu, and Baoquan Chen. 2018. Is There a Robust Technique for Selecting Aspect Ratios in Line Charts?. *IEEE Transactions on Visualization and Computer Graphics* 24, 12: 3096–3110. <https://doi.org/10.1109/TVCG.2017.2787113>
186. Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13, 4: 600–612. <https://doi.org/10.1109/TIP.2003.819861>
187. Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, 1398–1402. <https://doi.org/10.1109/ACSSC.2003.1292216>
188. Felix A Wichmann and Robert Geirhos. 2023. Are deep neural networks adequate behavioral models of human visual perception?. *Annual review of vision science* 9, 1: 501–524. <https://doi.org/https://doi.org/10.1146/annurev-vision-120522-031739>
189. Hadley Wickham and Lionel Henry. 2026. purrr: Functional Programming Tools. Retrieved from <https://purrr.tidyverse.org/>
190. Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemond, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson,

- Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. 2019. Welcome to the tidyverse. *Journal of Open Source Software* 4, 43: 1686. <https://doi.org/10.21105/joss.01686>
191. Hadley Wickham. 2016. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. Retrieved from <https://ggplot2.tidyverse.org/>
 192. Claus O Wilke and Brenton M Wiernik. 2020. ggtext: Improved text rendering support for ‘ggplot2’. *R package version 0.1.1*. Retrieved from <https://wilkelab.org/ggtext/>
 193. Jessica K Witt. 2019. Graph Construction: An Empirical Investigation on Setting the Range of the y-Axis. *Meta-Psychology*.
 194. Jeremy M Wolfe. 2020. Visual search: How do we find what we are looking for?. *Annual review of vision science* 6, 1: 539–562. <https://doi.org/10.1146/annurev-vision-091718-015048>
 195. Eugene Wu and Remco Chang. 2024. Design-Specific Transformations in Visualization. *arXiv preprint arXiv:2407.06404*. <https://doi.org/https://doi.org/10.48550/arXiv.2407.06404>
 196. Wayne Wu. 2024. We know what attention is!. *Trends in Cognitive Sciences* 28, 4: 304–318.
 197. Cindy Xiong, Cristina R. Ceja, Casimir J.H. Ludwig, and Steven Franconeri. 2020. Biased Average Position Estimates in Line and Bar Graphs: Underestimation, Overestimation, and Perceptual Pull. *IEEE Transactions on Visualization and Computer Graphics* 26, 1: 301–310. <https://doi.org/10.1109/TVCG.2019.2934400>
 198. Brenda W. Yang, Camila Vargas Restrepo, Matthew L. Stanley, and Elizabeth J. Marsh. 2021. Truncating Bar Graphs Persistently Misleads Viewers. *Journal of Applied Research in Memory and Cognition* 10, 2: 298–311. <https://doi.org/10.1016/j.jarmac.2020.10.002>
 199. Fumeng Yang, Lane T. Harrison, Ronald A. Rensink, Steven L. Franconeri, and Remco Chang. 2019. Correlation Judgment and Visualization Features: A Comparative Study. *IEEE Transactions on Visualization and Computer Graphics* 25, 3: 1474–1488. <https://doi.org/10.1109/TVCG.2018.2810918>
 200. 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). <https://doi.org/10.1145/3544548.3581111>
 201. Nathan Yau. 2015. Bar Chart Baselines Start at Zero. Retrieved December 6, 2023 from <https://flowingdata.com/2015/08/31/bar-chart-baselines-start-at-zero/>

202. Lei Yuan, Steve Haroz, and Steven Franconeri. 2019. Perceptual Proxies for Extracting Averages in Data Visualizations. *Psychonomic Bulletin & Review* 26, 2: 669–676. <https://doi.org/10.3758/s13423-018-1525-7>
203. Jeff Zacks and Barbara Tversky. 1999. Bars and Lines: A Study of Graphic Communication. *Memory & cognition* 27, 6: 1073–1079. <https://doi.org/10.3758/bf03201236>
204. Zehua Zeng and Leilani Battle. 2023. A Review and Collation of Graphical Perception Knowledge for Visualization Recommendation. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–16. <https://doi.org/10.1145/3544548.3581349>
205. Zehua Zeng, Phoebe Moh, Fan Du, Jane Hoffswell, Tak Yeon Lee, Sana Malik, Eunye Koh, and Leilani Battle. 2021. An evaluation-focused framework for visualization recommendation algorithms. *IEEE Transactions on Visualization and Computer Graphics* 28, 1: 346–356. <https://doi.org/10.48550/arXiv.2109.02706>
206. Zehua Zeng, Junran Yang, Dominik Moritz, Jeffrey Heer, and Leilani Battle. 2023. Too Many Cooks: Exploring How Graphical Perception Studies Influence Visualization Recommendations in Draco. *IEEE Transactions on Visualization and Computer Graphics*: 1–11. <https://doi.org/10.1109/TVCG.2023.3326527>
207. Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. <https://doi.org/10.48550/arXiv.1801.03924>
208. Mingqian Zhao, Huamin Qu, and Michael Sedlmair. 2019. Neighborhood Perception in Bar Charts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*, 1–12. <https://doi.org/10.1145/3290605.3300462>

APPENDIX

A.1 CHAPTER THREE RELATED

A.1.1 brms FORMULAE

For the two **comparison** tasks, the corresponding brms formula is

```
correct ~ 1 + (1 | pid) + (1 | rel:dist:cut), family =  
bernoulli.
```

For the two **estimation** tasks, the corresponding brms formula is

```
bf(diff ~ 1 + (1|pid) + (1|cut:rel:dist), hu ~ 1 + (1|pid)  
+ (1|cut:rel:dist)), family = hurdle_lognormal()
```

We used the following priors for all of our models:

- $\bar{\beta} \sim \text{Normal}(0, 1.5)$
- $\beta_{\text{PID}} \sim \text{Normal}(0, \sigma_{\text{PID}})$
- $\beta_{\text{cut:dist:rel}} \sim \text{Normal}(0, \sigma_{\text{cut:dist:rel}})$
- $\sigma_{\text{PID}}, \sigma_{\text{cut:dist:rel}}, \sigma \sim \text{Exponential}(1.5)$

A.1.2 MATH CALCULATIONS

For discordant data relation $\Delta A > \Delta B$, $\frac{\Delta A}{A_1} < \frac{\Delta B}{B_1}$:

PROOF. Suppose for contradiction that truncation $t > 0$ is non-monotonic, i.e., t reverses the data relation. This implies that

$$\begin{aligned}
\frac{\Delta A}{A_1 - t} &> \frac{\Delta B}{B_1 - t} \\
\underbrace{(\Delta B - \Delta A)t}_{<0} &> \underbrace{\Delta B A_1 - \Delta A B_1}_{>0} \\
&\quad \text{(given by data relation)} \\
t &< \underbrace{\frac{\Delta B A_1 - \Delta A B_1}{\Delta B - \Delta A}}_{<0}
\end{aligned}$$

Since truncation can not be negative, we conclude that it is impossible to have a non-monotonic truncation on data relation $\Delta A > \Delta B, \frac{\Delta A}{A_1} < \frac{\Delta B}{B_1}$. ■

Similarly, for discordant data relation $\Delta A < \Delta B, \frac{\Delta A}{A_1} > \frac{\Delta B}{B_1}$:

PROOF. Suppose for contradiction that truncation $t > 0$ is non-monotonic, i.e., t reverses the data relation. This implies that

$$\begin{aligned}
\frac{\Delta A}{A_1 - t} &< \frac{\Delta B}{B_1 - t} \\
\underbrace{(\Delta B - \Delta A)t}_{>0} &< \underbrace{\Delta B A_1 - \Delta A B_1}_{<0} \\
&\quad \text{(given by data relation)} \\
t &< \underbrace{\frac{\Delta B A_1 - \Delta A B_1}{\Delta B - \Delta A}}_{<0}
\end{aligned}$$

Since truncation can not be negative, we conclude that it is impossible to have a non-monotonic truncation on data relation $\Delta A < \Delta B, \frac{\Delta A}{A_1} > \frac{\Delta B}{B_1}$. ■

A.2 CHAPTER FOUR RELATED

A.2.1 NEURAL NETWORK ARCHITECTURES AND EXTRACTION

POINTS

We followed the implementation of Zhang et al. [207] when extracting deep features from AlexNet, SqueezeNet, and VGG16. We followed the implementation of Kumar et al. [103] when extracting deep features from ResNets and EfficientNets. See Cref{tab:extraction_points} for

details. Other works, such as Pihlgren et al. [135], have extracted four points for each network such that the chosen points represent “early, semi-early, middle, and late” layers in the convolutional layers. While Pihlgren et al.’s approach offers a more systematic way to sample features across network depth, we opted to align our extraction points with Zhang et al. [207] and Kumar et al. [103] to enable direct comparisons, though it’s worth noting that there is no broad consensus in the field regarding optimal feature extraction locations.

Network Architecture	Feature extraction points, ImageNet top-1 accuracy, and model size	Top-1 accuracy (%)	Model size (MB)	Num of Parameters
AlexNet	1st, 2nd, 3d, 4th, and 5th ReLU	56.522	233.1	61,100,840
VGG16	2nd, 4h, 7th, 10th, and 13th ReLU	71.592	527.8	138,357,544
SqueezeNet	1st ReLU, 2nd, 4th, 5th, 6th, 7th, and 8th Fire	58.178	4.7	1,235,496
ResNet18,	1st Conv2d, 1st	69.758	44.7	11,689,512
ResNet50	MaxPool2d, 2nd, 3rd, and 4th Block Stack	76.13	97.8	25,557,032
EfficientNet B0,	1st Conv2d, 2nd, 3rd, 4th, and 6th MB-Conv (mobile inverted bottle-neck)	77.692	20.5	5,288,548
EfficientNet B5		83.444	116.9	30,389,784

A.2.2 NEURAL NETWORK WEIGHTS

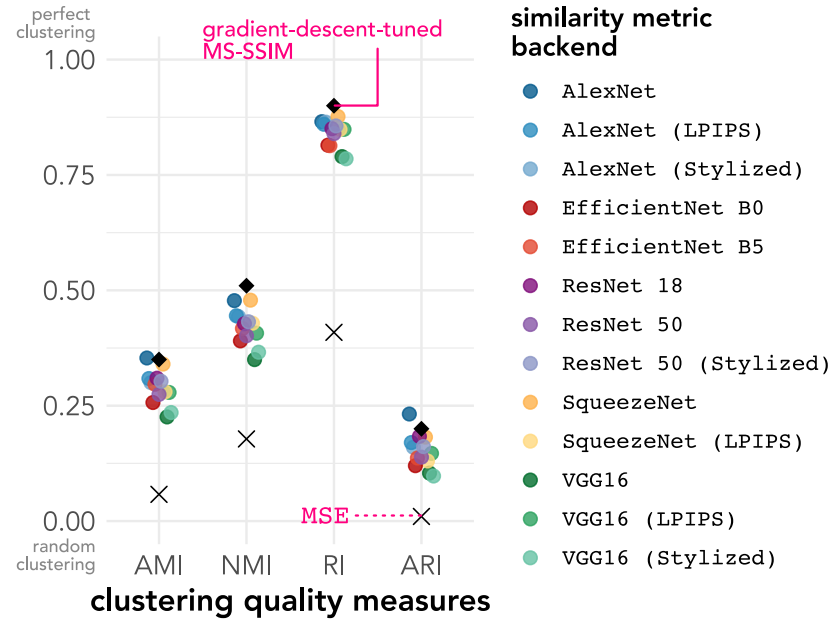
We obtained Stylized ImageNet weights from Geirhos et al. [53]. Stylized ImageNet is generated by AdaIn style transfer [78]⁸³. We utilize the Stylized ImageNet weights for AlexNet, VGG16, and ResNet50 trained by Geirhos et al. [53] at <https://github.com/rgeirhos/texture-vs-shape/>. For ResNet-50, Geirhos et al. [53] used the standard ResNet-50 architecture from PyTorch [132] (i.e., the `torchvision.models.resnet50` implementation). Geirhos et al. [53] used batch size of 256 and trained on Stylized ImageNet for 60 epochs with Stochastic Gradient Descent (`torch.optim.SGD`) using a momentum term of 0.5, weight decay $1e-4$ and a learning rate of 0.1 which was multiplied by a factor of 0.1 after 20 and 40 epochs of training. For AlexNet and VGG16, they used model architectures from `torchvision.models` and trained the networks under the identical circumstances as ResNet-50. Identical hyperparameter setting except for the learning rate — the learning rate for AlexNet was set to 0.001 and for VGG16 was 0.01 initially. Both learning rates were multiplied by 0.1 after 20 and 40 epochs of training (60 epochs in total). We obtained CIFAR-10 weights [100] trained for ResNet18 and ResNet50 from https://github.com/huyvnphan/PyTorch_CIFAR10. We obtained the SimCLR weights [20] trained for ResNet18 and ResNet50 from <https://github.com/sthalles/SimCLR>. We use the same feature extraction points as specified in Table 3.

⁸³... the code is available at <https://github.com/rgeirhos/Stylized-ImageNet>

A.2.3 PRE-PROCESSING WITH RESOLUTION 224×224

Similar to prior observations [103], we also found that using standard ImageNet resolution (224×224) yields slightly worse performance compared to a resolution of 64×64 .

Figure 50: Black diamonds represent Veras and Collins' [179] results using gradient-descent-tuned MS-SSIM. Points are slightly offset for clarity. Results when replicating Pandey et al's experiment using resolution $224 \times 224 \times 3$.



A.2.4 CLUSTERING OF SCATTERPLOTS

Figure 51: Consensus clustering of scatterplots from Pandey et al. [129].

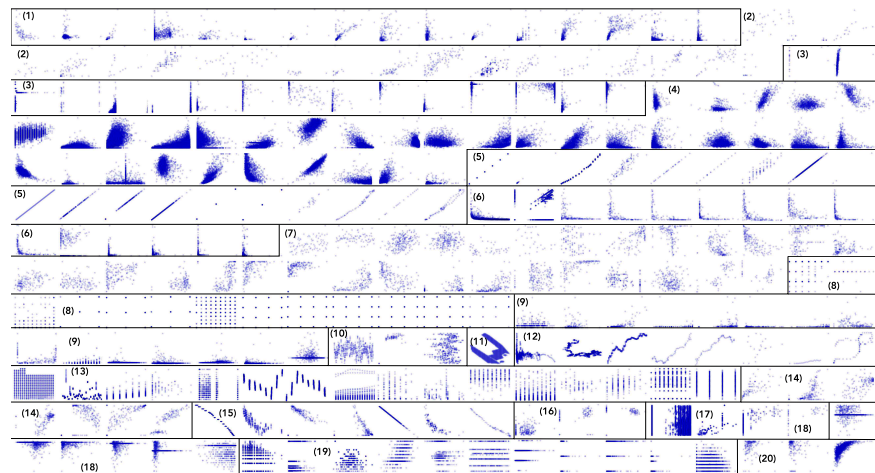
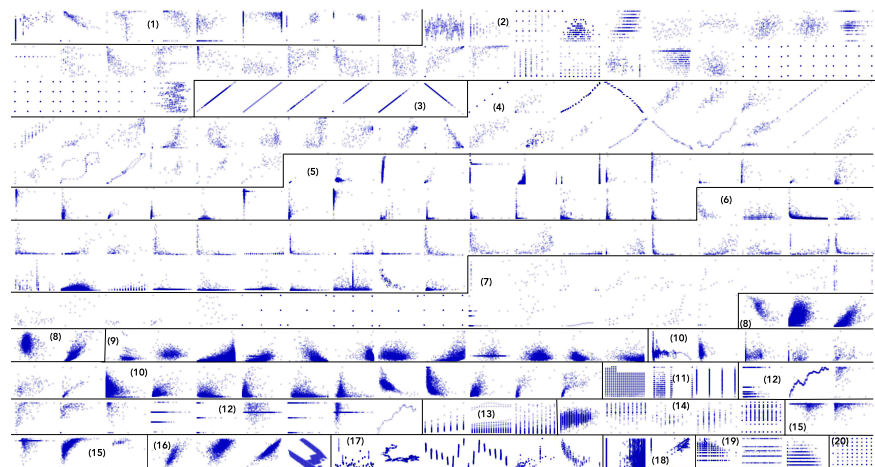


Figure 52: Clustering labels of scatterplots from Pandey et al. [129] using SqueezeNet (LPIPS) as the backbone network for the deep-feature-based similarity metric.



Backbone	RI	ARI	AMI	NMI
Veras and Collins	0.90	0.20	0.35	0.51
Mean Squared Error	0.409	0.010	9.0579	0.178
AlexNet	0.8878905	0.2378972	0.4078301	0.5328779
AlexNet (Stylized ImageNet)	0.8517824	0.1814445	0.3521323	0.4876277
AlexNet (LPIPS)	0.8935190	0.2419678	0.3943937	0.5241815
VGG16	0.8688325	0.1978015	0.3345852	0.4622434
VGG16 (Stylized ImageNet)	0.8759751	0.2099710	0.3494193	0.4757726
VGG16 (LPIPS)	0.8363122	0.1822110	0.3643023	0.4839279
SqueezeNet	0.8917086	0.2286150	0.3841792	0.5179834
SqueezeNet (LPIPS)	0.8939479	0.2433410	0.4053757	0.5353521
EfficientNet B0	0.8833152	0.2424203	0.4153032	0.5400465
EfficientNet B5	0.8613278	0.1856533	0.3402646	0.4677815
ResNet 18	0.8375959	0.1617837	0.3518211	0.4793912
ResNet 50	0.8516507	0.1918074	0.3637146	0.4807449
ResNet 50 (Stylized ImageNet)	0.8531648	0.2176265	0.3707914	0.4893624

Table 4: Clustering quality measures obtained from 64 × 64, highlighting the top one performing backbones for each measure.

A.3 CHAPTER SIX RELATED

A.3.1 FULL HIERARCHICAL MODEL SPECIFICATIONS

We follow the hierarchical notation system of Gelman and Hill [54] and denote participants' selection on the X-Axis and Y-Axis with $\hat{x}, \hat{y}$ respectively and the true answer with x, y . Unless otherwise specified, standard deviation parameters (σ) follow $\mathcal{N}^+(0, 1)$ and other parameters follow $\mathcal{N}(0, 1)$ as (hyper-)prior distributions. The full model specifications are in the quarto and html files in supplemental materials hosted at <https://osf.io/prtfq/>.

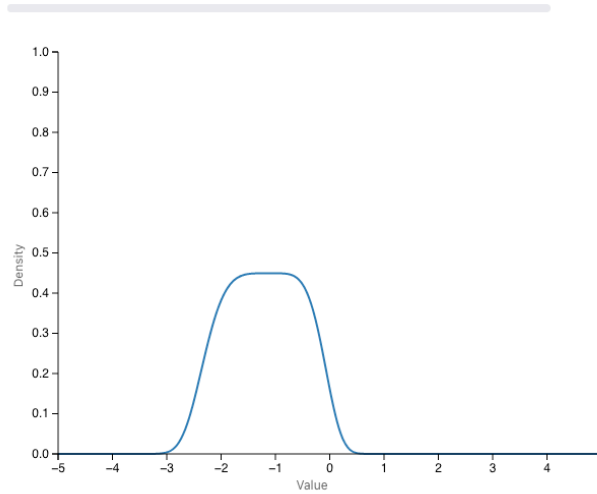
Figure 53: An example of the experiment interface. The generalized skewed- t distribution is parameterized with $\mu = -1.1$, $\sigma = 0.7$, $\lambda = -0.1$, $p = 4.6$, $q = 31.7$.

A.3.2 SCREENSHOT OF EXPERIMENT INTERFACE

Click the slider to initialize a selection.

Drag the slider to move the ball along the curve until a vertical line through it divides the area into two equal halves.

Adjust using left arrow $\leftarrow$ and right arrow $\rightarrow$ keys until the areas match.



A.3.3 SKEWED GENERALIZED T DISTRIBUTION

The skewed generalized t distribution (SGT), introduced by Theodosiou [171], is a univariate, unimodal distribution. We chose it as our data-generating distribution for its flexibility, minimal distributional parameters, and capacity to model a wider range of data characteristics than Gaussian, Laplace, and many other common distributions. Specifically, the SGT allows independent control over skewness (asymmetry), kurtosis (tail heaviness), and scale, enabling us to generate distributions that range from highly symmetric to strongly skewed, and from light-tailed (platykurtic) to heavy-tailed (leptokurtic). This flexibility is crucial for testing human perception across diverse distributional shapes that participants might encounter in real-world data visualization contexts.

The SGT probability density function is defined as

$$f_{\text{SGT}(x;\mu,\sigma,\lambda,p,q)} = \frac{p}{2\nu\sigma q^{\frac{1}{p}} B\left(\frac{1}{p}, q\right) \left[1 + \frac{|x-\mu|^p}{q(\nu\sigma)^p(1+\lambda \cdot \text{sign}(x-\mu))^p}\right]^{q+\frac{1}{p}}}$$

where μ is *mode*, σ is *scale*, λ is *skewness*, and p and q are kurtosis parameters. $B(\cdot, \cdot)$ is the beta function and ν is a variance adjustment parameter, defined as

$$\nu = \left(q^{-\frac{1}{p}} \left[(3\lambda^2 + 1) \frac{B\left(\frac{3}{p}, q - \frac{2}{p}\right)}{B\left(\frac{1}{p}, q\right)} - 4\lambda^2 \left(\frac{B\left(\frac{2}{p}, q - \frac{1}{p}\right)}{B\left(\frac{1}{p}, q\right)} \right)^2 \right] \right)^{-\frac{1}{2}}$$

To ensure sufficient variation in the visualized data range of $x \in [-5, 5]$, $y \in [0, 1]$, we sampled the values for the parameters as $\mu \sim \text{Uniform}[-2, 2]$, $\sigma \sim \text{Uniform}[0.5, 2.5]$, $\lambda \sim \mathcal{N}(0, 0.33)$, $p \sim \text{Uniform}[2, 4]$ and $q \sim \text{Uniform}[1, 50]$.

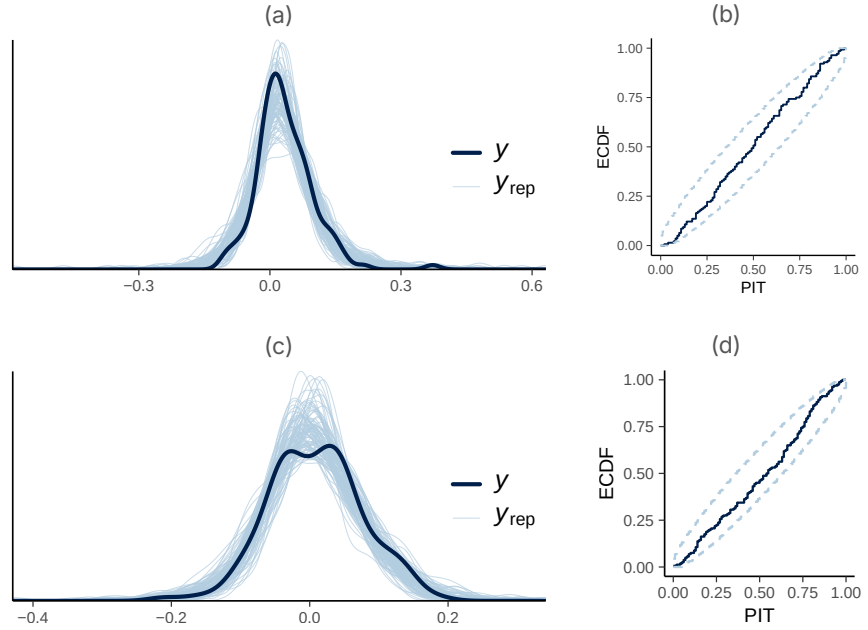
A.3.4 ADDITIONAL POSTERIOR CHECK RELATED PLOTS

A.3.4.1 CHECKS FOR EXPERIMENT 1

Following best practices, we plot and examine the posterior predictive check plots and the Probability Integral Transform-Empirical Cumulative Distribution Function (PIT-ECDF) calibration plots. Throughout our results, posterior predictive draws recover the shape of the participants' responses. For details on how to interpret posterior predictive plots, refer to Gabry et al [\[50\]](#).

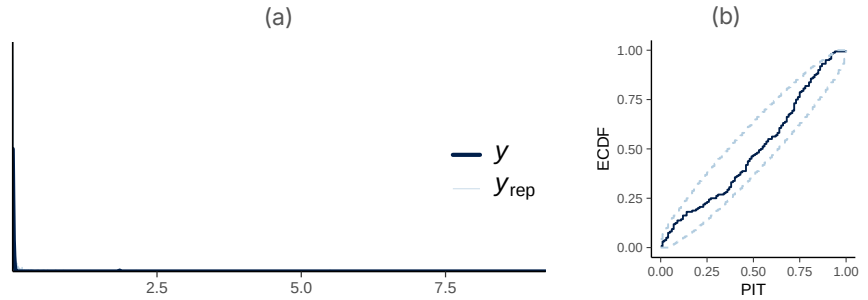
A.3.4.1.1 REMAINING TWO PROJECTION OPERATORS

Figure 54: Plots for **ProjToAxis_x** and **ProjToAxis_y** : (a), (c) Posterior predictive checks with 100 draws for x -Axis and y -Axis respectively. (b), (d) PIT-ECDF calibration plot for x -Axis and y -Axis respectively.



A.3.4.1.2 HIGHEST POINT Y OPERATOR

Figure 55: Plots for **HighestPoint_y** : (a) Posterior predictive check with 100 draws, (b) PIT-ECDF calibration.



A.3.4.1.3 GAUSSIAN MODEL FOR HighestPoint_x

The mathematical model for modeling the signed error for the highest point task (§ 5.4.2.2) on the x -axis is as follows:

$$\text{error}_i \sim \mathcal{N}(\mu_{\text{PID}[i]}, \sigma_{\text{PID}[i]}^2)$$

$$\mu_j \sim \mathcal{N}(\bar{\mu}, \bar{\sigma}^2), j \in [16]$$

$$\sigma_j \sim \mathcal{N}(\alpha, \beta^2), j \in [16]$$

$$\bar{\mu}, \alpha \sim \mathcal{N}(0, 1)$$

$$\bar{\sigma}, \beta \sim \mathcal{N}^+(0, 1)$$

Note that the Gaussian model does not quite capture the response distribution, but the good PIC-ECDF performance tells us that it's still a relatively good fit, and thus it works for the **BisectArea** operator in Section 5.4.2.4.

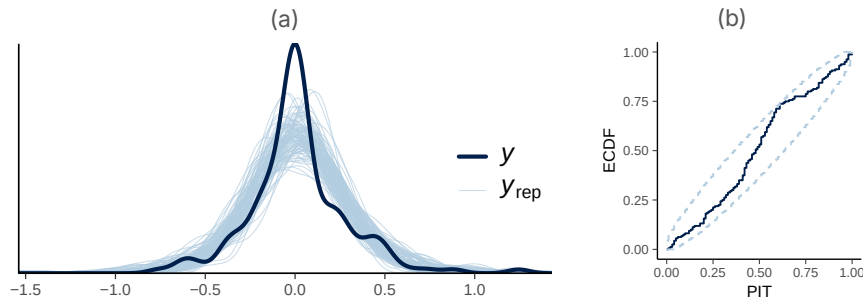


Figure 56: Plots for **Highest-Point_x** task for error on the x -Axis, except *here we are using a Normal distribution instead of a Laplace distribution* to model: (a) PIT-ECDF calibration chart, (b) Posterior predictive check with 100 draws

A.3.4.1.4 MAX SLOPE OPERATOR

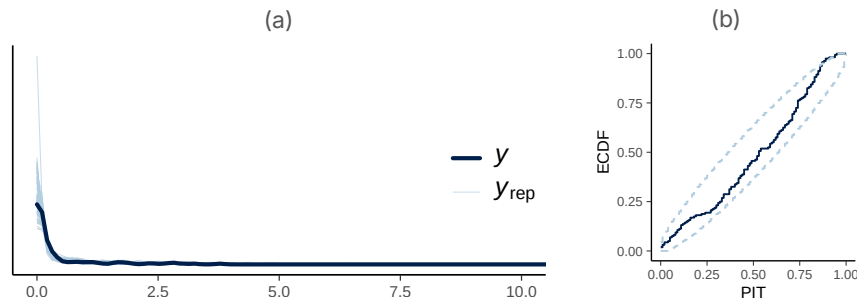
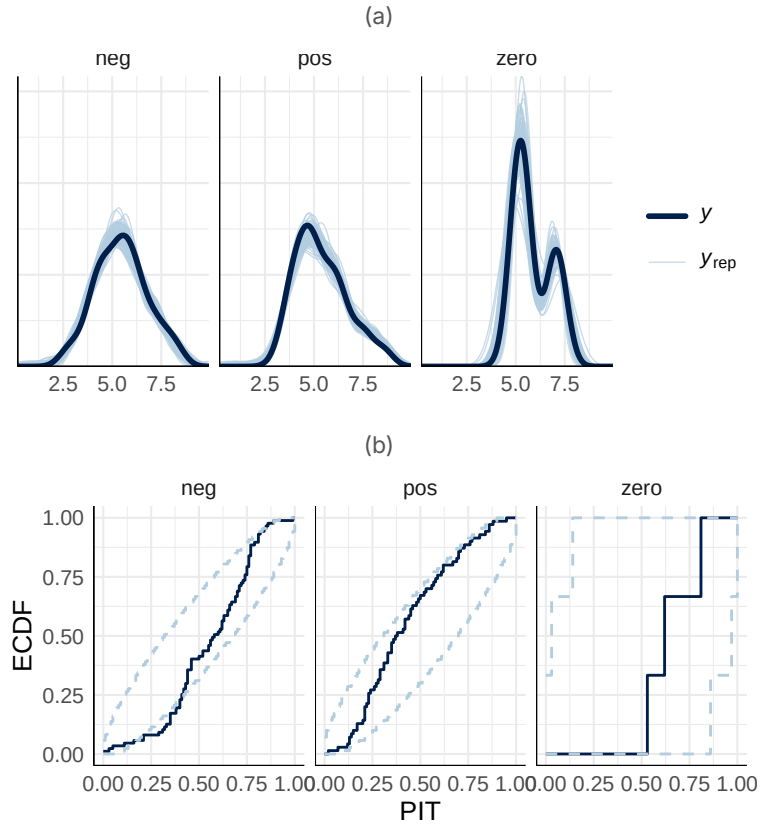


Figure 57: Plots for **MaxS-lope** : (a) Posterior predictive check with 100 draws, (b) PIT-ECDF calibration plot.

A.3.4.1.5 BISECT AREA OPERATOR

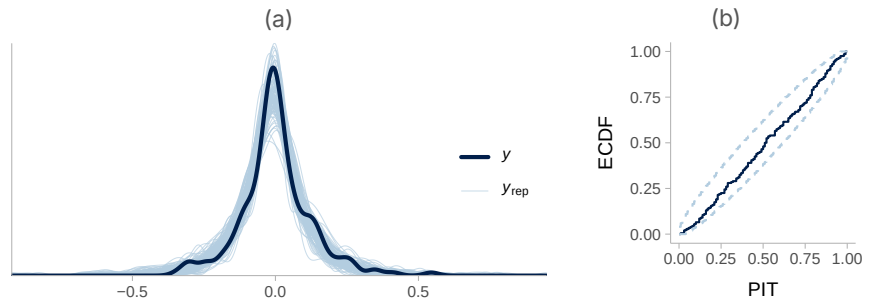
Figure 58: Plots for **Bi-sectArea**. (a) Posterior predictive checks with 100 draws, where the plot is faceted to have three parts, where the label indicates whether the sign of λ , the skewness parameter, is negative, positive, or zero, and (b) PIT-ECDF calibration chart, where the plot is faceted to have three parts, where the label indicates whether the sign of λ , the skewness parameter, is negative, positive, or zero. (a) and (b) demonstrate good fits



A.3.5 CHECKS FOR EXPERIMENT 2

A.3.5.1 PART 1 – PROJECT TO AXIS Y

Figure 59: Plots for **ProjToAxisY** for Experiment 2. (a) Posterior predictive checks with 100 draws for the y -Axis. (b) PIT-ECDF calibration plot for y -Axis.



A.3.5.2 PART 2 – PROJECT-ONCE

Note that the following are all plots generated using bayesplot [\[49\]](#).

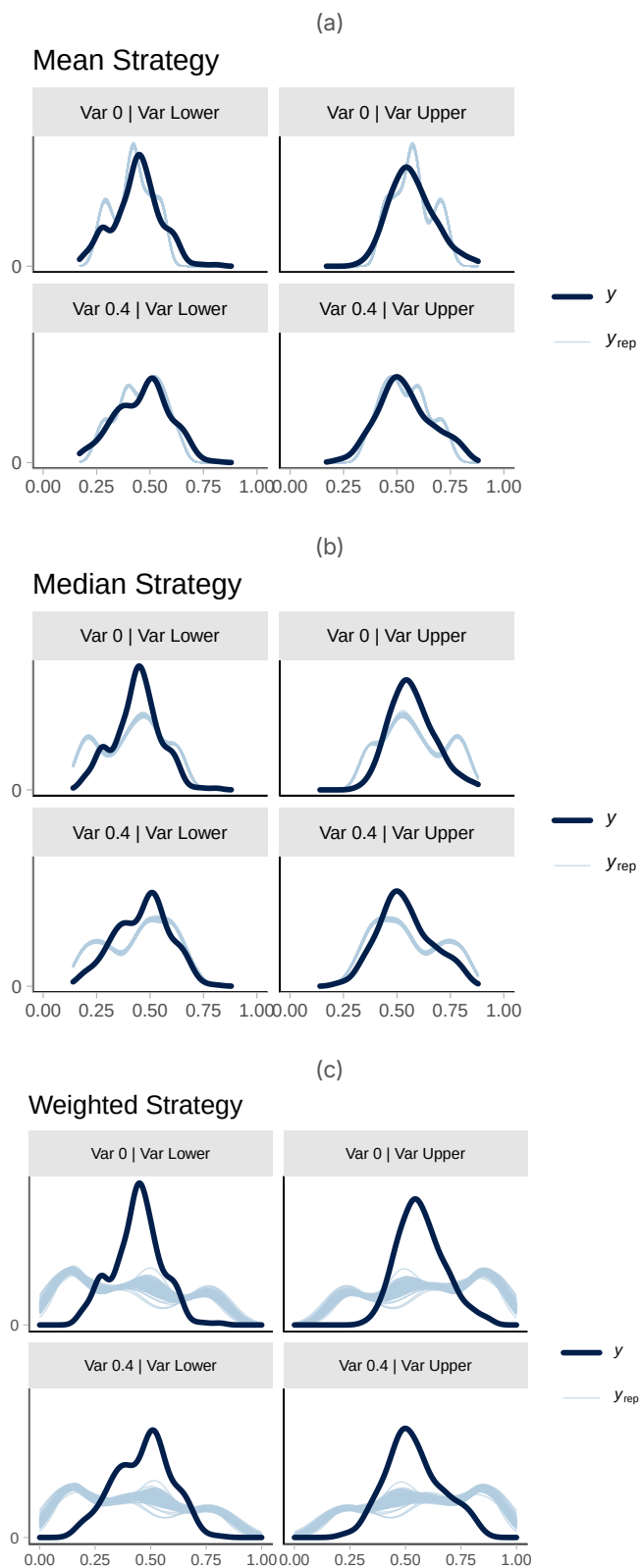


Figure 60: Predictions composed under **project-once** strategy for (a) mean, (b) median, and (c) weighted.

A.3.5.3 PART 2 – PROJECT-TWICE

Note that the following are all plots generated using `bayesplot` [\[49\]](#).

Figure 61: Predictions composed under **project-twice** strategy for (a) mean, (b) median, and (c) weighted.

