Uncovering How Scatterplot Features Skew Visual Class Separation

S. Sandra Bae*
sandra.bae@colorado.edu
ATLAS Institute
University of Colorado
Boulder, CO, USA

Takanori Fujiwara*
takanori.fujiwara@liu.se
Dept. of Science and Technology
Linköping University
Norrköping, Sweden

Chin Tseng*
chint@cs.unc.edu
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC, USA

Danielle Albers Szafir
danielle.szafir@cs.unc.edu
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC, USA

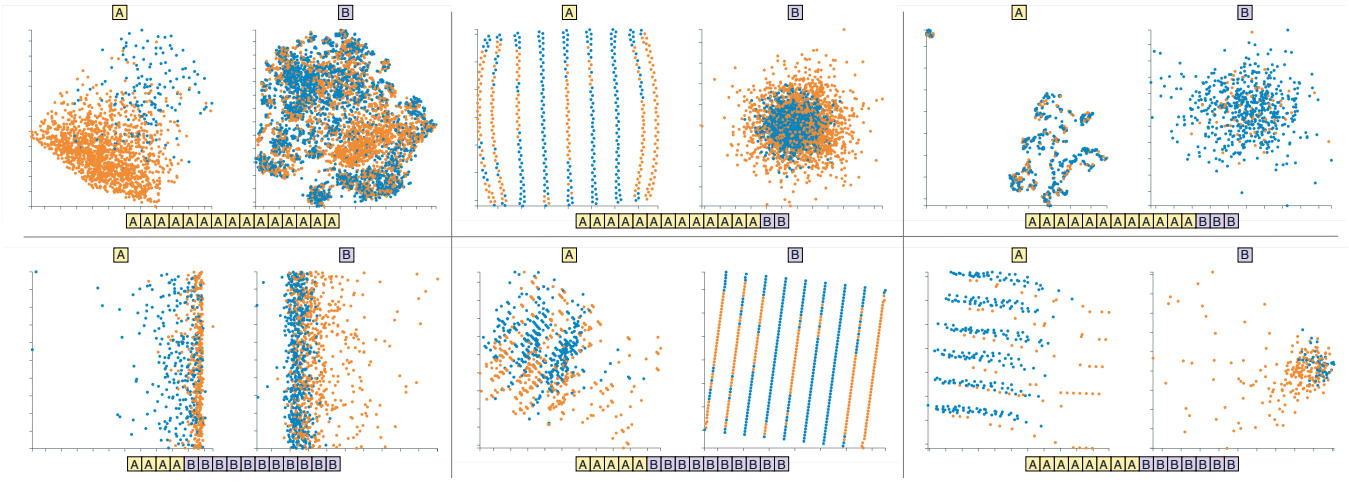


Figure 1: Examples showing people's selection on the task: *Which scatterplot (A or B) has blues and oranges that are more separated?* In this study, we aim to find which set of scatterplot features impact people's perception in Visual Class Separation (VCS) tasks. Yellow and purple boxes represent each participant's answers. For example, all participants selected Scatterplot A in the top left pair, while 8 participants selected A and the other 7 participants selected B in the bottom right.

Abstract

Multi-class scatterplots are essential for visually comparing data, such as examining class distributions in dimensionality reduction and evaluating classification models. Visual class separation (VCS) measures quantify human perception but are largely derived from and evaluated with datasets reflecting limited types of scatterplot features (e.g., data distribution, similar class densities). Quantitatively identifying which scatterplot features are influential to VCS tasks can enable more robust guidance for future measures. We analyze the alignment between VCS measures and people's perceptions of class separation through a crowdsourced study using 70 scatterplot features relevant to class separation. To cover a wide range of scatterplot features, we generated a set of multi-class scatterplots from 6,947 real-world datasets. Our results highlight that multiple combinations of features are needed to best explain VCS. From our analysis, we develop a composite feature model that identifies key scatterplot features for measuring VCS task performance.

*These authors contributed equally to this work and are listed alphabetically.

This work is licensed under a Creative Commons Attribution 4.0 International License.
CHI '25, Yokohama, Japan
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713976

CCS Concepts

• **Human-centered computing → Visualization design and evaluation methods**; **Empirical studies in visualization**.

Keywords

Visualization, Multi-class scatterplots, Classification complexity, Visual quality measures, Scagnostics, Quantitative study

ACM Reference Format:

S. Sandra Bae, Takanori Fujiwara, Chin Tseng, and Danielle Albers Szafir. 2025. Uncovering How Scatterplot Features Skew Visual Class Separation. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26-May 1, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3706598.3713976

1 Introduction

Multi-class scatterplots [38] are used for a range of analytical tasks, including examining class distributions in dimensionality reduction [18, 35, 77] and estimating the classification quality of machine learning models [41, 73, 103]. For these high-level tasks, an analyst must *visually separate* the labeled data to understand distribution differences between different classes (e.g., determine whether two classes are seen as (dis)similar for analytical reasoning). Following Bernard et al.'s definition [14], a *visual class separation* (VCS) task aims to *quantify how well distributions of predefined classes in scatterplots are separated* (see Fig. 1 for examples). Though this task is related to visual clustering [1, 51], which involves perceiving and identifying spatially proximate points (referred to as *clusters*), it differs in its use of predefined (and therefore explicitly encoded) classes. Our work aims to enhance VCS for multi-class scatterplots, as human performance on VCS directly influences the scientific and practical insights obtained through class comparison [2, 35].

Given the importance of VCS, developers can use various VCS measures to guide their visualization designs [9, 79, 94]. These measures aim to predict how well people can visually separate classes (e.g., is Class A well separated from Class B?). Developers use these measures to recommend visually interesting pairs of variables [79], automatically select dimensionality reduction results [9, 15, 83], and develop perception-based dimensionality reduction methods [94]. These measures can also contribute to predicting and avoiding perceptual biases when visually estimating a machine learning model's classification quality [73]. Despite their usefulness, existing VCS measures are evaluated with a limited variety of scatterplots (e.g., classes following close-to-normal distributions) [9, 14, 79, 94]. This limited set of evaluations raises questions as to how human perception may differ when performing VCS tasks that are beyond these evaluation constraints. For instance, machine learning and real-world datasets encompass non-Gaussian distributions (see Sec. 4.2), which existing VCS models do not account for.

Scatterplots have a multitude of features that may influence VCS performance, such as the number of points plotted, the density of points, and shapes formed from the contours of points (Fig. 1). Sedlmair et al.'s taxonomy [78] synthesizes these features and speculates as to how these features may influence the perception of class separation. However, we lack guidance on how we can *quantitatively* measure each feature and evaluate its influence on class separation. A quantitative approach can offer actionable and generalizable methods to understand the influence of different features on VCS measures and to develop more reliable metrics. Toward this goal, this paper investigates two research questions: (RQ1) What are the key scatterplot features that influence human perception of VCS? and (RQ2) Do existing VCS measures align with human perception? To answer these questions, we introduce quantitative measures of multi-class scatterplot features related to VCS.

We conducted a crowdsourced study using 294 scatterplots across and 70 features related to VCS. These 294 scatterplots are a subset of the 6,947 scatterplots we generated from real-world datasets, which cover a wide range of the multi-class scatterplot feature values. We scope our work to multi-class scatterplots with two classes and use hue-based encoding for class information. This approach aligns with prior methods investigating VCS measures [9, 76]. We analyzed the data to uncover associations among visual class separation measures, multi-class scatterplot features, and participants' perceived class separation.

Our results highlight how features related to classification complexity [58], within-class [89] attributes, and between-class attributes [9] heavily impact people's judgment on VCS tasks. Additionally, our results highlight the need for future work to fully understand the effect of existing VCS measures on human performance. We use feature selection to derive a set of 26 features that strongly correlate with people's perception of class separation based on our study data and use these features to generate a composite feature model that outperforms the state-of-the-art VCS measure. This derived measure suggests a more robust approach to predicting class separation and helps identify key features of data distribution that influence separation.

The primary contributions of this paper are:

- statistical analyses that provide quantitative insights into how human perception is influenced by multi-class scatterplot features;
- a composite feature model that better predicts human performance than existing VCS measures;
- new quantitative measures of multi-class scatterplot features;
- a methodology to generate multi-class scatterplots and resultant datasets that more comprehensively evaluate VCS measures.

2 Related Works

We survey prior work about visual diagnostic measures, VCS measures, and human perception for scatterplots.

2.1 Visualization Diagnostic Measures

Diagnostic measures in visualization have a rich history. Early attempts to assess and improve visualization quality focused on graphical aspects. For example, Tufte [88] introduced the *data-ink ratio*, which quantifies the proportion of a visualization's "ink" (or pixels) devoted to data. Similarly, the graph drawing community developed *graph aesthetics* to design better layouts for node-link diagrams, including maximum symmetry [28, 55], minimum edge crossing [33], and minimum edge bends [82]. Brath [17] and Miller et al. [63] advocate for diagnostic measures that can "predict successful visualizations based on objective quantities that can be easily measured" [63]. Since then, research has introduced various diagnostic measures that quantitatively capture analytical patterns in a visualization, often termed *visual quality metrics* or VQMs [12, 16]. Bertini et al.'s [16] survey highlights six types of quantitative VQMs: clustering [4, 51, 83, 97], correlation [4, 69, 83], outlier [4, 69, 97],

complex patterns [79, 83, 97], image quality [31, 69], and feature preservation [79, 83].

Many of these metrics are grounded in studies of scatterplots. Scatterplots support various exploratory data analysis tasks, such as checking correlations, outliers, clusters, and data distributions [75]. Scatterplots are also frequently used with high dimensional data, comparing key properties of different groups within a dataset. Before performing such analytical tasks, analysts often need to select pairs of variables from the working dataset or apply dimensionality reduction to generate 2D scatterplots. As the number of variables increases to tens, hundreds, or even thousands, examining all possible combinations of variables or dimensionality reduction results becomes exceedingly time-consuming and even unfeasible. Scatterplot diagnostic measures can expedite the process of finding the right relationships for comparing key groups to create scatterplots summarizing patterns of interest.

Scagnostics [89] is arguably the most well-known scatterplot diagnostic measure and characterizes scatterplots according to a series of features (e.g., skewed, clumpy, convex, skinny). Variants of scagnostics measures [24, 61, 95, 97] and interactive analysis systems that use scagnostics [25, 97] further improve scagnostics' robustness and add new analytical capabilities. There are also diagnostics measures more specifically for scatterplots generated by dimensionality reduction methods. These measures focus on tasks related to distances among points, such as visual clustering tasks [1, 51] and VCS tasks [9, 79, 94].

2.2 Visual Class Separation Measures

VCS measures can help generate visualizations that effectively support visual analytics tasks [9, 15, 94]. Current VCS measures derive from class centroid-based or nearest neighbor-based approaches. The class centroid-based approach decides the degree of class separation by agreement between each point's belonging class and each point's closest class centroid. This approach is taken by the distance consistency measure (*DSC*) [79] and its variants (e.g., *density-aware DSC*) [94]. Nearest-neighbor approaches measure the degree of class separation based on the class label of the closest points to a given target point. This second approach selects neighbors and computes neighbors' class similarity using a range of metrics. Aupetit & Sedlmair [9] noted that the best-performing overall measure is *GONG 0.35 DIR CPT*, which uses a nearest-neighbor approach. This measure selects neighbors based on a γ-observable neighbor graph [8] with $\gamma = 0.35$ and directed edges and then computes their similarity based on the proportion of belonging to one specified target class. Distributions consistency (or *DC*) [79] partitions a scatterplot by a grid and defines neighbors as points in the same grid cell. Wang et al. [94] also designed a nearest neighbor graph-based measure (named *density-aware KNNG*) that considers the distance between each neighbor.

Though each existing measure is evaluated in its respective work, subsequent studies reevaluate these measures using different datasets [9, 78, 84]. Among these studies, Sedlmair et al.'s taxonomy [78] is most closely related to our work. The taxonomy applies two existing measures, *DSC* and *DC*, to 816 scatterplots in part to identify cases where *DSC* and *DC* did not fit the authors' perception of class separation. They used this analysis to derive a taxonomy of scatterplot features that likely influence VCS tasks, consisting of four categories (*scale*, *point distance*, *shape*, *position*) and corresponding *within-class* and *between-class* factors. For example, the combination of these parameters introduces each class' scatterplot area size and the variance of their sizes.

We extend the analysis introduced by Sedlmair et al. [78] through expanded (1) scatterplot feature measures, (2) data diversity, and (3) evaluation methodology to offer additional insight into VCS. While the taxonomy enumerates several key multi-class scatterplot features, these features lack methods to quantitatively measure them. We offer a concrete implementation of these measures by introducing a set of formal quantitative definitions. Second, the 816 scatterplots used in their study are made by applying 4 dimensionality reduction methods to 31 real-world and 44 synthetic datasets, resulting in a limited coverage of feature values. We expand upon these datasets to be more inclusive of feature variety, with a specific emphasis on non-Gaussian distributions, while applying modern dimensionality reduction methods to over 800 real-world datasets to reflect VCS situations that analysts may currently encounter. Third, we conduct a crowdsourced study in place of the original qualitative analysis approach to increase the generalizability of the evaluation given the expected variance from perceptual variability [26, 57, 105]. Perceptual variability highlights variance among individuals: people may behave differently even for the same perceptual tasks (e.g., information retrieval/search [22, 91]). The Axiom of Perceptual Variability [7] implies that the two authors/coders' judgment may fail to account for the distribution of perceived VCS across the wider population. A crowdsourced study ($n = 150$) helps account for perceptual variability in evaluating class separability. These extensions enable us to quantitatively evaluate how well VCS measures capture human perception in a wider range of scenarios. The resulting models can support more robust and situationally-aware approaches to visualization design.

2.3 Scatterplot Features in Task Perception

VCS measures quantify how well people can perceive the differences between classes. Visualization often uses knowledge and methods from visual perception to drive design across a broad range of tasks [30, 43, 46, 71, 96, 99, 100]. By understanding how people perceive different patterns in their data, we can make informed choices about what visualizations are most likely to support the needs of a given task or dataset. Perceptual organization and ensemble coding are two notable perceptual operations related to class separability. Both perceptual operations allow us to quickly estimate the gist of a scene, getting the big picture about a data distribution to help orient people to group properties (see Sec. 2.2).

Perceptual organization creates hierarchical visual representations from lower-level components [93]. In class separability, the lower-level components are individual instances (i.e., scatterplot points) that we perceive as higher-level groups. Theories of perceptual organization in visualization have been heavily influenced by Gestalt principles of grouping [93]. For instance, the principle of *similarity* states that objects with similar shapes or colors are perceived as part of the same groups. The principle of *proximity* suggests that elements that are closer to each other than they are to other items are perceived as a group. The principle of *continuity*

states that elements will group together if they lie on the same contour. These principles can help us understand how humans perceive visualizations when performing a specific task, including VCS.

Ensemble coding allows individuals to quickly extract information on sets of objects based on the distribution of visual features (e.g., orientation [10], size [6], or color [72]). These characteristics are rapidly and efficiently estimated before active attention, capturing group- or set-level properties rather than individual details about a given object. Szafir et al. [81] highlight the importance of ensemble coding in visualizations. For example, ensemble coding allows people to quickly estimate the position of a group of scatterplot points without attending to each point individually. Consequently, people can summarize a data distribution by rapidly estimating over an entire set of visual marks (e.g., mean size, color of glyphs [6, 53]).

A growing number of interdisciplinary studies use vision science methods to study these perceptual operations for scatterplot design [30, 71, 104]. Much of this research has explored how people accomplish different scatterplot tasks [75], including those that do not involve class information such as correlation [101], causality [100], target location [40], trend estimation [56], similar scatterplot search [67], and visual clustering [51, 70]. Several works focus on multi-class scatterplot tasks that require analysis across classes [38]. Etemadpour et al. [32] investigated scatterplot features' influence on VCS tasks, using eye tracking to examine visual attention in 20 synthetic scatterplots varying in point density, drawing area size, and shape to evaluate the influence of Gestalt principles on class separation. Divis et al. [27] confirmed the influence of each class' density of points and drawing area size on class separation. These studies indicate that human visual attention can act differently based on multi-class scatterplot features. We further investigate potentially influential features stated by Sedlmair et al.'s [78] and analyze the discrepancy between existing VCS measures and human perceptions.

3 Multi-class Scatterplot Features

We describe multi-class scatterplot features[1] to systematically choose real-world datasets and understand these features' contribution to VCS tasks. Sedlmair et al.'s taxonomy [78] suggests conceptual features that may influence human perception of VCS. As Sedlmair et al. do not provide concrete implementations of these conceptual features, we introduce a set of formal quantitative definitions for instantiating these features. We distinguish between the conceptual features and the instantiated features by denoting them with typewriter (e.g., `conceptual feature`) and math italic fonts (e.g., $instantiated\ feature$), respectively (see Fig. 2). Given how visual class separation tasks can be viewed as parallel to performing classification of 2D data, we further include classification complexity measures [45, 58] developed in the machine learning community (see Sec. 3.2). These classification complexity measures can extract features that Sedlmair et al.'s taxonomy does not cover (e.g., structures of class boundaries).

[1]The source code implementing these features: https://github.com/takanori-fujiwara/multiclass-scatterplot-features

3.1 Feature Implementation

The first half of our multi-class scatterplot features stems from concepts proposed by Sedlmair et al.'s visual classification taxonomy [78]. This taxonomy describes 16 conceptual features, which are binned into four categories: Scale, Point Distance, Shape, and Position. See Fig. 2 for a visual summary.

- Scale refers to the number of points (i.e., data scale) in addition to the proportion of the chart area covered by the points (i.e., scatterplot-area scale).
- Point Distance refers to the distribution of points within a class, namely (i) how dense or sparse a class is and (ii) whether there are outliers that are distant from the majority of the points.
- Shape refers to perceived visual configuration based on the distribution of points within a class, specifically recognizing the spatial relationships among the points. This category mainly references the Gestalt grouping principles [93].
- Position describes a class's overall position, such as the class centroid and the margin between classes.

Each category contains two feature types: within-class features (i.e., characteristics of each class) and between-class features (i.e., interactions between classes).

3.1.1 Background. We implement various multi-class scatterplot features by extending methods employed in scagnostics [89, 98] to multi-class scatterplots. To help understand how we applied scagnostics for multi-class scatterplots, we provide the necessary background for these methods below.

Outlier Removal for Measuring Features As discussed in Wilkinson et al.'s design rationale for scagnostics [98], scatterplot features are highly influenced by outliers. Outlier removal is necessary to robustly measure various scatterplot features. We follow Wilkinson et al.'s outlier removal procedure [89, 98]. For each class, we first build a minimum spanning tree (MST) [39] for all points within a scatterplot. We consider a point to be an outlier if all of its edges in the MST have a length greater than a threshold, ω. ω is defined as $\omega = P_{75} + 1.5(P_{75} - P_{25})$, where P_i is the i-th percentile of the MST's edge lengths. We apply this outlier removal procedure before conducting any additional preprocessing or feature analysis.

Convex and α-Hull Construction Our multi-class scatterplot features require knowing each class's convex and α-hulls [29] to capture visual configurations of points in a multi-class scatterplot. After outlier removal (see above), we derive a class's convex hull as the smallest convex polygon that contains all points of that class. We then compute the α-hull—the non-convex polygon enclosing all points—with the area of the hull parameterized by α. More precisely, the α-hull is defined as the intersection of a set of circles with radius $1/\alpha$ that contains all the points [29]. Following Wilkinson et al. [98], we use $\alpha = 1/P'_{90}$, where P'_{90} is the 90th percentile of edge lengths of the MST after outlier removal. However, we found that when the circle diameter is smaller than the outlier removal threshold (i.e., $2P'_{90} < \omega$), we cannot generate a hull containing all points in some cases. For these cases, we iteratively update α by $\alpha \leftarrow 0.95\alpha$ until we successfully generate an α-hull.

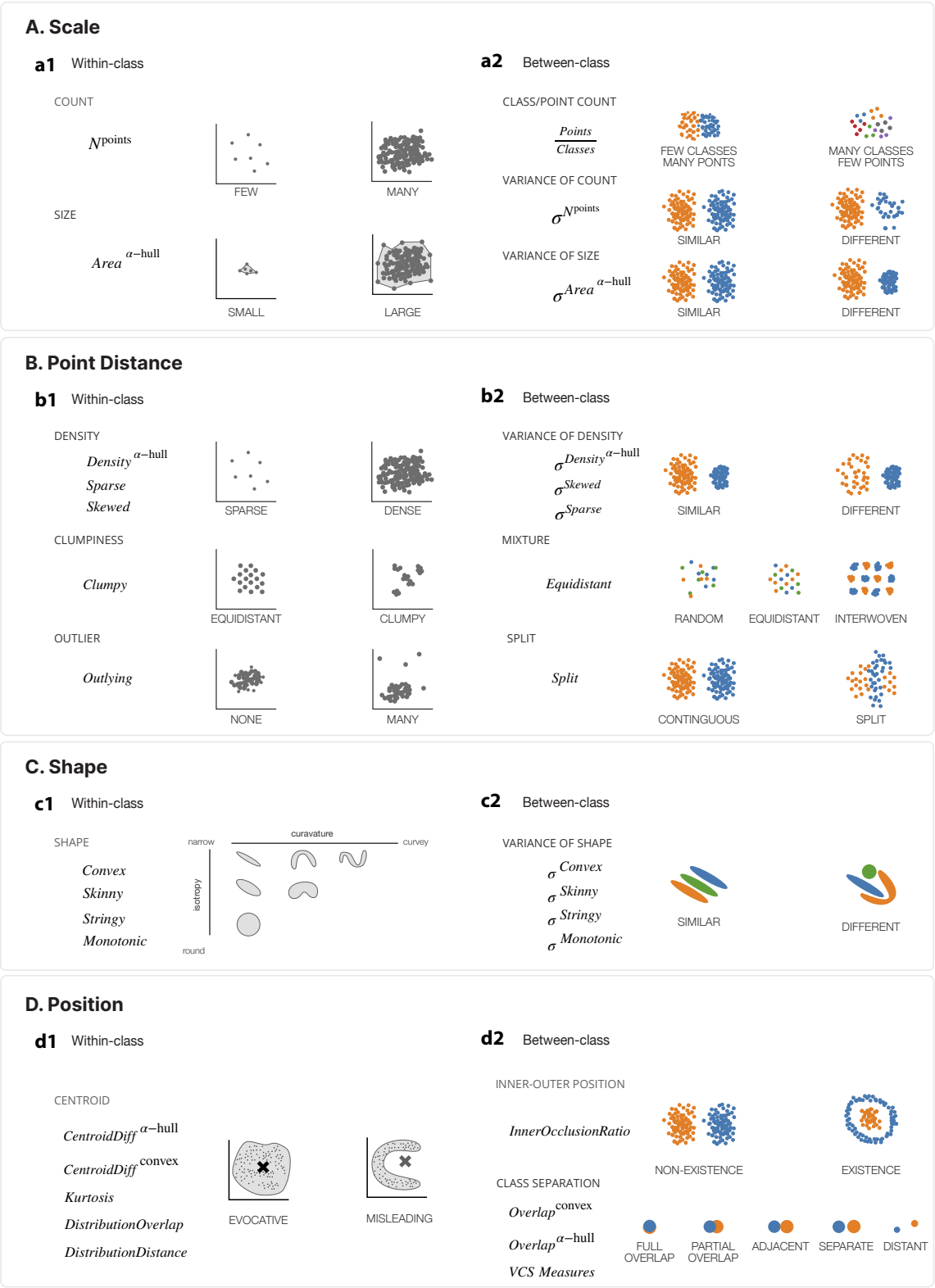


Figure 2: Visual summary of multi-class scatterplot features in Sec. 3.1. Each category (A, B, C, D) contains *within-class* and *between-class* features. Icons adopted and modified from Sedlmair et al's taxonomy [78]. Note that VCS Measures refers to four existing measures: *DSC*, *GONG* 0.35 *DIR CPT*, *density−awareDSC*, *density−awareKNNG*

Scagnostic Measures Although scagnostic measures [98] are designed for single-class scatterplots, we can still apply these measures *separately* to each class in a multi-class scatterplot to capture within-class features (e.g., quantifying the degree of outlier ratios for each class). We can also derive various between-class features by comparing each class's scagnostic measures (e.g., the standard deviation of two classes' measures). We note that scagnostic measures do not provide any information on whether the classes overlap or not. Class overlaps are captured by our new measures (e.g., *Split* in Sec. 3.1.3, $Overlap^{\text{convex}}$ in Sec. 3.1.5), existing VCS measures, and classification complexity measures (cf. Sec. 3.2). To model features described below, we use 8 scagnostic measures (*Skewed*, *Sparse*, *Clumpy*, *Outlying*, *Convex*, *Skinny*, *Stringy*, *Monotonic*). Though multiple variants of these measures exist [95, 97, 98], we employ a commonly used one from Wilkinson et al. [98]. Wilkinson et al. suggest hexagonal binning to improve computational performance, but we do not apply hexagonal binning as Wang et al. [97] note that some measures are overly sensitive to this binning (e.g., *Outlying* and *Clumpy*).

3.1.2 Scale. Scale relates both the point number and overall area spanned by the points.

Within-class Features

`Count` is defined by the number of points within a class. We adhere to this straightforward definition and denote it as N^{points}.

`Size` is described as the spread of points in a class in terms of 2D area. We instantiate `Size` as an area of the α-hull described in Sec. 3.1.1 and denote it as $Area^{\alpha\text{-hull}}$.

Between-class Features

`Class-Point Count` is the ratio between the overall number of points and the number of classes available in the dataset. We compute this relationship, $Points/Classes$, with: $\sum_i N_i^{\text{points}}/N^{\text{classes}}$ where N^{classes} is the number of classes and N_i^{points} is the number of points in the i-th Class ($i \in \{1, \cdots, N^{\text{classes}}\}$).

`Variance of Count` is a conceptual feature that informs the dispersion of `Count` of each class. We instantiate this feature by computing the *standard deviation* of classes' `Count`, denoted by $\sigma^{N^{\text{points}}}$. When having only two classes (i.e., our study scope), the standard deviation is equivalent to half of the absolute value of the difference between the class counts.

`Variance of Size` informs the dispersion of `Size`. We compute the standard deviation, $\sigma^{Area^{\alpha\text{-hull}}}$.

3.1.3 Point Distance. Point distance captures distributional patterns in the spacing between individual points, consisting of their density, clumpiness, and outlierness.

Within-class Features

`Density` is the ratio between two Scale features: `Count` and `Size`. We instantiate this feature as $Density^{\alpha\text{-hull}} = N^{\text{points}}/Area^{\alpha\text{-hull}}$. When $N^{\text{points}}/Area^{\alpha\text{-hull}}$ is small, points in a class are sparsely distributed in a scatterplot and vice-versa. However, this characterization assumes uniform point distributions within the α-hull. To characterize density in greater detail, we expand Sedlmair et al.'s concept of `Density` by considering *distance distributions*, which can inform whether a scatterplot class has region(s) where points are densely packed. We achieve this by adding two scagnostic measures—*Skewed* and *Sparse*. These measures are defined as:

$$Skewed = \frac{P'_{90} - P'_{50}}{P'_{90} - P'_{10}}$$

$$Sparse = P'_{90}$$

Based on Wilkinson et al.'s [98] definition, high *Skewed* indicates a scatterplot class has significant density variance over its area. *Sparse* determines whether points are positioned in only a limited number of locations within the area.

`Clumpiness` refers to a scagnostic concept describing a class's inter-point distribution [89]. We use scagnostics' *Clumpy*. High *Clumpy* indicates sets of points are placed relatively far away compared to others. Given *Clumpy* is also computed after the outlier removal (Sec. 3.1.1), high *Clumpy* generally implies a scatterplot class has a locally dense region of points. For more details, refer to Wilkinson et al. [98].

`Outlier` quantifies distant points from the majority of a class. We compute scagnostics' *Outlying* for each class. *Outlying* is derived when performing outlier removal. From edge lengths of the MST used to perform outlier removal, *Outlying* is computed as the ratio of the sum of edge lengths that are longer than ω to the sum of all edge lengths in the MST. When outlier points are extremely far away from the other point, *Outlying* is high.

Between-class Features

`Variance of Density` measures the dispersion of `Density` across classes. We compute the standard deviation for each corresponding measure: $\sigma^{Density^{\alpha\text{-hull}}}$, σ^{Skewed}, σ^{Sparse}.

`Mixture` describes characteristics of points when classes are fully or partly overlapping. Sedlmair et al. identified three patterns in mixtures: random, equidistant, and interwoven patterns. A random pattern corresponds to the case where there is no clear structure seen in the overlapped area. An equidistant pattern shows similar or equal distances among the points for each different class. The interwoven pattern is similar to the equidistant pattern but differs in having similar or equal distances among *groups* of points for each different class. We introduce a measure, *Equidistant*, to determine whether a mixture pattern is random or equidistant. We also discuss how *Equidistant* can also be used to identify the interwoven pattern.

To capture the equidistant pattern, for each point in a class, we first find the smallest circle that contains a point of *another* class. Let $\mathbf{x}_i^a$ denote i-th point belonging to Class a. Then, we can compute a radius corresponding to the smallest circle at $\mathbf{x}_i^a$ that contains any point in Class b as follows:

$$r_i^{a\to b} = \min\{\|\mathbf{x}_i^a - \mathbf{x}_j^b\| \mid \mathbf{x}_j^b \in \text{points in Class } b\}$$

We compute $r_i^{a\to b}$ for all points in Class a, and obtain a vector, $\mathbf{r}^{a\to b}$, containing all corresponding radii. Then, *Equidistant* between Classes a and b can be defined as:

$$Equidistant_{a,b} = \frac{2}{\text{SD}\left(\frac{\mathbf{r}^{a\to b}}{\text{mean}(\mathbf{r}^{a\to b})}\right) + \text{SD}\left(\frac{\mathbf{r}^{b\to a}}{\text{mean}(\mathbf{r}^{b\to a})}\right)}$$

Using the above equation, our objective is to determine how strictly classes within a scatterplot have an equidistant relationship. To do so, we compute the standard deviations (SD(·)) of $\mathbf{r}^{a\to b}$ and $\mathbf{r}^{b\to a}$ after normalizing these radii by their mean. Normalization is applied to avoid impacts from scaling differences of radii by each class. Then, we compute the average of these two standard deviations and use its inverse as an equidistance measure between Class a and b, $Equidistant_{a,b}$. As a summary measure, we can take a mean of $Equidistant_{a,b}$ of all possible pairs of classes:

$$Equidistant = \text{mean}\left\{ Equidistant_{a,b} \;\middle|\; \begin{array}{l} a \in \{1, \cdots, N^{\text{classes}}\} \\ b \in \{1, \cdots, N^{\text{classes}}\} \\ a \neq b \end{array} \right\}$$

We do not introduce an instantiated method to measure the interwoven pattern because this pattern can be captured by using *Equidistant* in conjunction with *Clumpy*. If an interwoven pattern exists, a multi-class scatterplot will have high *Equidistant* and high *Clumpy*.

`Split` distinguishes how clearly the points in a class are split into regions physically distanced from another class and its points. We instantiate *Split* as follows: Let $S_a^{\alpha\text{-hull}}$ be a set of points in the α-hull of Class a. We first subtract α-hull of Class b from α-hull of Class a: $S_a^{\alpha\text{-hull}} - S_b^{\alpha\text{-hull}}$. This output is the geometric difference between both classes. Then, from this subtracted geometry, we judge whether $S_a^{\alpha\text{-hull}}$ is separated by Class b into multiple regions. Note that this judgment can be easily performed by using existing libraries such as Shapely [37] (e.g., with Shapely, checking whether $(S_a^{\alpha\text{-hull}} - S_b^{\alpha\text{-hull}})$ is a "MultiPolygon"). Then, we define *Split* of Class a by Class b as:

$$Split_{a,b} = \max\left(\delta_{a-b}\frac{\text{area}\left(S_a^{\alpha\text{-hull}} - S_b^{\alpha\text{-hull}}\right)}{\text{area}\left(S_a^{\alpha\text{-hull}}\right)}, \delta_{b-a}\frac{\text{area}\left(S_b^{\alpha\text{-hull}} - S_a^{\alpha\text{-hull}}\right)}{\text{area}\left(S_b^{\alpha\text{-hull}}\right)}\right)$$

where area(·) computes the area of an input geometry. $\delta_{a-b} = 1$ if $(S_a^{\alpha\text{-hull}} - S_b^{\alpha\text{-hull}})$ has multiple split regions and $\text{area}(S_a^{\alpha\text{-hull}} - S_b^{\alpha\text{-hull}})/\text{area}(S_a^{\alpha\text{-hull}})$ is over a threshold; otherwise, $\delta_{a-b} = 0$. We set the threshold to 0.1 by default and use it to measure split. The threshold helps eliminate cases where a minor fraction of a class is split by another class. By taking the ratio of the subtracted and original areas, $Split_{a,b}$ quantitatively indicates the extent these two classes are split apart. When $Split_{a,b}$ is high, one class is clearly split by another class within a tight margin. Similar to *Equidistant*, the mean of $Split_{a,b}$ from all possible class pairs can be used as a summary measure, resulting in *Split*.

3.1.4 Shape.

Within-class Features

`Shape` describes the perceived visual configuration of the class points. `Shape` has two possible axes: isotropy and curvature. Isotropy indicates the directional pull of the visual configuration with values ranging from narrow (i.e., non-isotropic) to round (i.e., isotropic). Curvature describes the nonlinearity of the visual configuration. We again use scagnostic measures as modeled features for `Shape`. Specifically, we use *Convex*, *Skinny*, *Stringy*, and *Monotonic*. *Convex* is defined as $Convex = Area^{\alpha\text{-hull}}/Area^{\text{convex}}$ where $Area^{\alpha\text{-hull}}$ and $Area^{\text{convex}}$ are areas of the α- and convex hulls of a class, respectively. *Convex* measures the nonlinearity of the visual configuration using α-hull area. When the α-hull area is relatively small compared to the convex hull, nonlinearity is high. *Skinny* is the ratio of α-shape's perimeter to α-shape's area: $Skinny = 1 - \sqrt{4\pi Area^{\alpha\text{-hull}}}/(\text{perimeter of the } \alpha\text{−shape})$. Similar to *Skinny*, *Stringy* measures how narrow of a width the visual configuration is but differs as it is designed to have a high value when there is no "branch-like shape" (refer to Wilkinson et al. [98]). *Monotonic* is the squared Spearman correlation coefficient of x- and y-coordinates of points of a class. Thus, *Monotonic* indicates whether or not the visual configuration has a narrow, monotonic pattern.

Between-class Features

`Variance of Shape` measures how different `Shape` is for each class. We compute the standard deviation for each corresponding measure: σ^{Convex}, σ^{Skinny}, $\sigma^{Stringy}$, $\sigma^{Monotonic}$.

3.1.5 Position.

Position summarizes the relative locations of a class overall with respect to other classes, including its centroid and properties of its overall distribution.

Within-class Features

`Centroid` describes how misleading a class's center position is for estimating separation. Sedlmair et al. claimed that if points in a class do not follow a Gaussian distribution, the center position can be misled due to a mismatch between data and graphical centers (e.g., see Fig. 2-d1). We design five measures to quantify how misleading a class's center position is. These five measures cover two considerations: differences in the centroids of the hulls of the classes and differences in the overall distribution of points. Two measures, $CentroidDiff^{\alpha\text{-hull}}$ and $CentroidDiff^{\text{convex}}$, define a mismatch between the data centroid and the visual configuration centroid. The other three measures, *Kurtosis*, *DistributionOverlap*, and *DistributionDistance*, capture the (non-)Gaussian properties of point distributions.

$CentroidDiff^{\alpha\text{-hull}}$ is defined as the Euclidean distance between the geometric center of the α-hull of a class and the centroid of the points in that class. Similarly, we define $CentroidDiff^{\text{convex}}$ by using the *convex hull* of a class instead of α-hull.

Our three additional measures focus on the (non-)Gaussian distributions of points. These three measures are inspired by Independent Component Analysis [23, 49], where quantitative measures of non-Gaussianity are used when decomposing a multivariate signal. For *Kurtosis*, we follow Hyvärinen & Oja's kurtosis definition [49]. After kurtosis is computed for each x- and y-coordinates of points in a class, we use the absolute sum of these two kurtoses as *Kurtosis*. Higher *Kurtosis* infers higher non-Gaussianity. For *DistributionOverlap*, we first construct a 2D Gaussian distribution that mirrors the mean and variance of points in a class. From this distribution, we draw the same number of random samples as the number of points in the class (i.e., N^{points} samples). We then compute the similarity between the samples from the 2D Gaussian distributions and the points in

the class by taking their histogram intersection [65]. To decide the bin width of the histograms, we apply the method employed by NumPy[2] to combined data of the samples and points (specifically, the minimum bin width of those derived with Freedman–Diaconis and Sturges rules). We further divide the histogram intersection by the number of histogram bins for normalization. Given how this process involves randomness when drawing the samples, we repeat this process 10 times by default and define *DistributionOverlap* as the mean of resultant histogram intersections. If points in a class follow the Gaussian distribution, *DistributionOverlap* is close to 1. *DistributionDistance* measures how a class's point distribution is (dis)similar to a Gaussian distribution. *DistributionDistance* is computed in the same manner as *DistributionOverlap* except we substitute histogram intersection with the Hellinger distance [44].

Between-class Features

`Inner-Outer Position` describes the spatial proximal relationship between one class and the other classes within a scatterplot. For example, a class can be fully surrounded by another class. To capture this relationship between Class a and others, we introduce $InnerOcclusionRatio_a$, a measure that quantifies how much of the class with a smaller convex hull overlaps with the convex hull of the larger class. $InnerOcclusionRatio_a$ can be defined as:

$$InnerOcclusionRatio_a = \frac{\text{area}\left(S_a^{\text{convex}} \cap \left(\bigcup_{b \neq a} S_b^{\text{convex}}\right)\right)}{\min\left(\text{area}\left(S_a^{\text{convex}}\right), \text{area}\left(\bigcup_{b \neq a} S_b^{\text{convex}}\right)\right)}$$

where S_a^{convex} is a set of points in a convex hull of Class a, $\cap$ produces the intersected region of two convex hulls, and $\cup$ generates the combined region of two convex hulls (i.e., $\bigcup_{b \neq a} S_b^{\text{convex}}$ is the combined region of the convex hulls from all other classes). We define *InnerOcclusionRatio* as the maximum of $InnerOcclusionRatio_a$ of all classes. *InnerOcclusionRatio* identifies whether at least one class is surrounded by others.

`Class Separation` describes the spatial overlap and distance between a pair of classes (i.e., 2 classes), resulting in either full overlap, partial overlap, adjacency, separate, or distant. Sedlmair et al. [78] note that this feature will be strongly influenced by all of the other features described above. **Quantitatively examining this spatial relationship is the objective of VCS measures.** Existing VCS measures include *DSC* (a class centroid-based measure), *GONG* 0.35 *DIR CPT* (a nearest neighbor-based measure), and their variants (see Sec. 2.2). In addition to these measures, we introduce two simple shape-based measures, $Overlap^{\text{convex}}$ and $Overlap^{\alpha\text{-hull}}$. We define these measures as:

$$Overlap^{\text{convex}} = \sum_{a \in \{1, \cdots, N^{\text{classes}}\}} \sum_{b > a} \text{area}\left(S_a^{\text{convex}} \cap S_b^{\text{convex}}\right)$$

$$Overlap^{\alpha\text{-hull}} = \sum_{a \in \{1, \cdots, N^{\text{classes}}\}} \sum_{b > a} \text{area}\left(S_a^{\alpha\text{-hull}} \cap S_b^{\alpha\text{-hull}}\right)$$

These measures quantify the total overlapped area of the convex hulls and α-hulls for all pairs of classes, respectively, focusing on how much *spatial overlap* classes share. This aspect differs from how existing VCS measures rely on *distances* between points in different classes.

[2]Corresponding to using bin="auto" for histogram_bin_edges: https://numpy.org/doc/stable/reference/generated/numpy.histogram_bin_edges.html

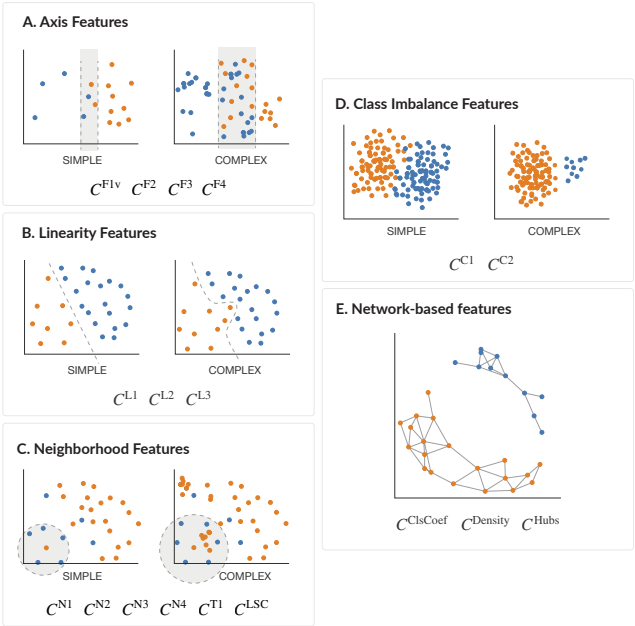


Figure 3: Visual summaries of five categories of classification complexity measures discussed in Sec. 3.2. All classification complexity measures are represented as C^{Idenfier}, and Identifier **corresponds to the common abbreviations used in the literature [45, 58].**

3.1.6 Implementation. We implemented all the above measures including scagnostics, existing VCS measures (e.g., *GONG* 0.35 *DIR CPT*), and newly designed ones (e.g., *Equidistant*) with Python 3. We used NumPy/SciPy [92] for matrix calculation, Shapely [37] and Alpha Shape Toolbox [13] for geometric operations (including α-hull generation), and scikit-learn [68] for neighbor graph generation required to implement VCS measures.

3.2 Features Corresponding to Classification Complexity

We also include multi-class scatterplot features that encompass the classification complexity measures surveyed by Lorena et al. [58] to supplement the features in Sec. 3.1 as well as to examine how machine learning-based features align with human perceptions through our user study. The survey discusses 22 classification complexity measures grouped into six categories: axis, linearity, neighborhood, dimensionality, and class imbalance measures. We provide a summary description for these categories and their measures, except for *dimensionality*. We exclude this category because our multi-class scatterplots have only two dimensions (i.e., x- and y-coodinates), and the complexity of dimensionality is almost always constant (i.e., 2 dimensions) and negligible. In contrast to Sedlmair et al.'s taxonomy, Lorena et al. provide an implementation library [5] that we used directly to implement the complexity measures. See Fig. 3 for visual summaries of the five categories.

- *Axis measures* characterize how easily classes can be separated based on the multi-class scatterplot's axes/directions. Directions can be x-direction, y-direction, or a direction at an arbitrary angle. For example, if points in different classes have few overlaps along the x-direction, they are likely easily separated across the x-axis. Note that Lorena et al. originally named this category as "feature" measures, but we use the term "axis" to avoid confusion with other features that we discuss in the paper.
- *Linearity measures* quantify how easily points in different classes can be separated by a line. If the linearity is high, multi-class scatterplots can be easily classified with a simple boundary.
- *Neighborhood measures* characterize classification difficulty based on how different classes mix within each point's local neighborhoods near the class boundaries. In contrast to linearity measures, neighborhood measures can help us understand whether each class in a multi-class scatterplot can be separated using a nonlinear boundary.
- *Network measures* extract structural information from points in each class by constructing a graph where points act as nodes and edges exist only between points closer than an algorithmically determined distance threshold. This information can describe a scatterplot's density or whether distinct clusters exist.
- *Class imbalance measures* compute the imbalance in the number of points per class.

For more comprehensive details, we refer readers to Lorena et al.'s survey. We note that some classification complexities have significant overlaps with measures described in Sec. 3.1. For example, the class imbalance measures are heavily related to `Variance of Count`. Despite their similarity, the classification complexity measures can help augment the measures discussed in Sec. 3.1. For instance, linearity and neighborhood measures can quantify the shape of boundaries of each class's points, which is not captured by `Variance of Shape`.

4 Methodology

We performed a crowdsourced study measuring how scatterplot features influence people's judgment on VCS tasks. This study allows us to characterize the effects of different scatterplot features. We hypothesized that:

H1: Multiple scatterplot features will influence human perceptions for VCS tasks. We reason **H1** given how scatterplot features are designed to quantify and capture data characteristics; thereby these features should also be reflected in people's perceptions.

H2: Scatterplot features related to class separation and between-class features will have higher associations with human VCS perceptions than other features. Given that our primary task relates to class separation, features that are designed to measure class separation, such as VCS measures and classification complexity measures, should more strongly align with people's perception than other features (e.g. N^{points}). Also, between-class features should outperform within-class features as between-class features emphasize the differences in their values.

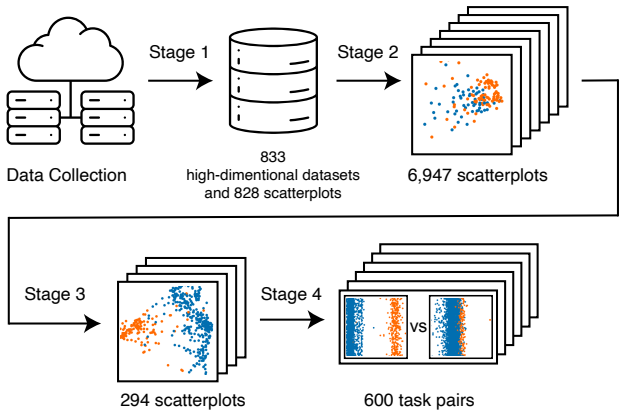


Figure 4: The four stages of our stimuli generation (Sec. 4.2). Stages 1–3 focus on scatterplot selection from real-world datasets. Stage 4 focuses on choosing task pairs for the user study.

4.1 Task

VCS tasks using scatterplots have been studied extensively [9, 76–79, 94]. Our study employed a two-alternative forced choice design task. Participants were presented with a pair of two-class scatterplots side-by-side, and they were asked to select which scatterplot is more visually separated for the two classes. This task aims to understand how features and their values influence human perception of VCS.

4.2 Stimulus Generation

We rendered scatterplots using D3 on a 400 × 400 pixel white background with two orthogonal black axes with 10 unlabeled ticks (Fig. 1). Each scatterplot contained two classes, color-coded with blue and orange in the D3 Category10 color palette. Each point was visualized with a 2-pixel radius. In internal piloting, we found that this radius made points in different classes distinguishable while minimizing overdraw in our tested datasets. The z-order for points was randomly sampled to avoid potential bias from overdraw.

Inspired by Pandey et al.'s method to generate a set of similar scatterplots [67], we employed a four-stage process (Fig. 4) to systematically choose pairs of datasets to render as horizontally juxtaposed scatterplot pairs.

4.2.1 Stage 1: Data Collection from Online Data Repositories. In the first stage, we collected datasets from multiple online dataset repositories. Adhering to our general motivation, we aimed to include a variety of datasets such that our scatterplots would holistically encompass different feature values, which will be generated at Stage 4 of our stimuli generation process. The dataset repositories we included are:

Data Source 1: Sedlmair and Aupetit's two-class scatterplots. This repository[3] provides 828 2D two-class scatterplots. These 2D scatterplots are generated by applying four different dimensionality reduction methods to 75 datasets (31 real, 44 synthetic). The four dimensionality reduction methods used in

[3]https://sepme.vda.univie.ac.at/

this repository are principal components analysis (PCA) [47, 52], robust PCA [85], multidimensional scaling (MDS) [86], and t-SNE [90]. We note that existing VCS measure evaluations largely relied on this repository [9, 76–78, 94]. We collected all 828 scatterplots.

Data Source 2: UCI Machine Learning Repository The UCI Machine Learning Repository[4] contains popular datasets for performing machine learning tasks, such as classification and clustering. This repository accounts for 13 of 31 real-world datasets that Sedlmair and Aupedit [9] used for their 2D two-class scatterplot generation. Given our need for predefined classes, we collected all datasets suitable for a classification task by using the `ucimlrepo` package.[5] The `ucimlrepo` package provides parameter controls for dataset queries. For instance, we limited the number of data points within a dataset ($< 10,000$) to avoid excessive computation costs for the subsequent stages of our stimuli generation process. In the end, we collected 194 high-dimensional datasets from this repository.

Data Source 3: VisuMap Datasets. This repository[6] contains 25 real-world datasets from various domain applications (e.g., financial industry and bioinformatics). 7 of 31 real-world datasets used in Sedlmair and Aupedit's two-class scatterplot generation are also from this repository. We collected all 21 datasets that have predefined labels.

Data Source 4: Jeon et al.'s clustering validation datasets. This repository[7] provides 96 labeled datasets curated from multiple different data repository sources: Kaggle,[8] the UCI Machine Learning Repository, and research papers. We collected all 96 of them.

Data Source 5: OpenML Datasets. This repository[9] provides over 5,000 datasets curated from various online sources (e.g., Kaggle, Rdatasets,[10] DataBrewer,[11] public GitHub repositories). Similar to Data Source 2, we only collected labeled datasets with less than 10,000 data points and datasets with no missing values. This query resulted in 522 datasets.

In summary, we collected 833 high-dimensional datasets and 828 scatterplot datasets. We provide details of these in the supplementary materials. These 833 high-dimensional datasets are further processed in the next stage to generate two-class scatterplots.

4.2.2 Stage 2: Scatterplot Generation Using Dimensionality Reduction Methods. Similar to how scatterplots were generated in Data Source 1, we applied 10 different dimensionality reduction (DR) methods to each high-dimensional dataset. However, in contrast to Data Source 1, we used a wider variety of datasets (over 800 real-world datasets vs. 31 real-world datasets) and also selected DR methods that (1) correspond to methods already used for Data Source 1, (2) cover state-of-the-art methods, and (3) consider the analytical context when using multi-class scatterplots (i.e., analysts want to compare predefined groups).

[4]https://archive.ics.uci.edu/
[5]https://github.com/uci-ml-repo/ucimlrepo
[6]https://visumap.com/index.html?VisuMapDatasets
[7]https://hyeonword.com/clm-datasets/
[8]https://www.kaggle.com/
[9]https://docs.openml.org/contributing/Datasets/
[10]https://vincentarelbundock.github.io/Rdatasets/datasets.html
[11]https://github.com/rmax/databrewer

Figure 5: Examples of dimensionality reduction results. These results are generated by applying PCA, MDS, t-SNE, UMAP, and PHATE to the texture dataset [19] available in the OpenML repository. We can see the differences in visual configurations among the t-SNE, UMAP and PHATE results. While t-SNE tends to generate more rounded shapes, PHATE shows more narrow, curvy shapes. The UMAP result has narrower shapes than the t-SNE's with a wide white space.

First, we selected DR methods already employed by Data Source 1: PCA, MDS, and t-SNE. Given how ordinary PCA consumes a large memory space for high-dimensional data, we applied a more scalable variant, specifically, incremental PCA [74]. Second, we included more recently developed methods such as UMAP [62] and PHATE [64]. These DR methods are widely used in bioinformatics as they preserve local, continuous relationships between data points in high-dimensional space. This preservation is critical for perception-dependent analysis tasks (e.g., single-cell trajectory inference) [20, 21]. Despite using the same 833 high-dimensional datasets, UMAP and PHATE generate significantly different visual configurations of scatterplot points compared to PCA, MDS, and t-SNE (see Fig. 5). The inclusion of UMAP and PHATE in our study can add visual configurations that are unique to these methods. Third, we incorporated DR methods designed for comparative analysis of predefined data groups, specifically, linear discriminant analysis (LDA) [50], contrastive PCA [2], ccPCA [34], and unified linear comparative analysis (ULCA) [36]. Comparative analysis using these methods usually involves VCS tasks (e.g., how well separated is each group? How much do the groups overlap?) [36]. Thus, this third inclusion can reveal how multi-class scatterplot features influence VCS tasks. We further included Gaussian random projection [3] to generate scatterplots that may have features the aforementioned methods cannot generate. In total, we used 10 different DR methods.

A multi-classscatterplot was generated for each combination of dataset and DR method. We first eliminated duplicates of collected datasets from the different data sources (e.g., Data Sources 1, 2, 4, and 5 may have the same datasets collected from the UCI machine learning repository) by identifying datasets that have the same numbers of data points and dimensions. We performed a Z-score normalization to each dataset before applying DR. For each method, we used default parameters employed by scikit-learn [68] or the original authors' implementations. However, we note three exceptions. First, since LDA can only output ($N^{\text{classes}} - 1$) dimensions, we assigned random y-coordinates for datasets with $N^{\text{classes}} = 2$. Second, given that ULCA accepts various parameters to flexibly compare data groups (e.g., how much their separation should be emphasized), we assigned these parameters randomly. Lastly, we only applied MDS, cPCA, ccPCA, and ULCA to datasets with less than 1,000 data points due to their relatively high time or space complexity. After this DR process, we binarized data labels by following

a similar procedure as Sedlmair and Aupetit [76]. We assigned Class 0 as one class label by a random sampling that prioritizes selecting a class with more data points. We then assigned Class 1 to all of the other class labels. At the end of Stage 2, we obtained 6,947 two-class scatterplots (6,119 derived from the high-dimensional datasets and 828 scatterplots from Data Source 1).

4.2.3 Stage 3: Feature and Clustering-based Scatterplot Selection. In the third stage, we selected a subset of 6,947 scatterplots to maintain a reasonable number of stimuli for the user study. For each two-class scatterplot, we computed all instantiated features and class complexity measures described in Sec. 3. In total, each two-class scatterplot is represented with 70 features. The breakdown includes 32 *within-class* features, 20 *between-class* features, and 18 *classification complexity measures*.

We computed all 16 within-class features for both Class 0 and Class 1 and extracted the minimum and maximum of each feature as multi-class scatterplot features (e.g., the minimum of N^{points} (denoted by $N^{\text{points}}_{\text{min}}$) and the maximum of N^{points} ($N^{\text{points}}_{\text{max}}$)). This process produced 32 features. We computed all 20 between-class features using our implementation method in Fig. 2. Of these, four existing VCS measures are part of the 20 features: *DSC* [79], *GONG* 0.35 *DIR CPT* [9], *density-awareDSC* [94], and *density-awareKNNG* [94]. These VCS measures are instantiations of class centroid-based and nearest neighbor-based approaches, which can help inform `Class Separation` from multiple aspects (refer to Sec. 2.2). Note that *GONG* 0.35 *DIR CPT* inherently outputs a different score based on which class is selected as the target class (Class 0 or Class 1) [9]. Hence, we obtained two features corresponding to either case when selecting Class 0 or Class 1 as a target. Lastly, we computed 18 classification complexity measures spanning the five categories (i.e., axis, linearity, neighborhood, network, and class imbalance measures).

We then sampled representative scatterplots with clustering methods. Our aim for this sampling is to select a small set of scatterplots with a wide variety of feature values. Using spectral clustering [66], we first generated 100 microclusters from the 6,947 scatterplots. Then, we selected 3 samples from each microcluster by applying k-means clustering with $k = 3$ and selecting those most closely placed to each k-means cluster center. This process generated 300 sampled scatterplots. However, we manually removed 6 scatterplots because of heavy overdraw within the scatterplot (e.g., all points are located only at a few different coordinates). From this filtering process, we selected 294 scatterplots.

Given how there are many features (70 features) relative to the number of scatterplots, we considered two aspects when applying spectral clustering: reducing the number of features and considering the influence of the curse of dimensionality for distance computations. First, to reduce the number of features, we applied correlation-based feature selection. We first removed redundant features by choosing only one feature from similar features that have over 0.8 Pearson's or Spearman's absolute correlation coefficient with each other, reducing 70 features to 40. We then applied PCA to the remaining features and selected the minimum number of principal components (PCs) that preserved 95% of the original data's variance, resulting in 26 PCs (i.e., compressed features). Second, as the affinity matrix for spectral clustering, we used an adjacency matrix corresponding to a k-nearest neighbor graph of scatterplots. Using a k-nearest neighbor graph can mitigate the issues caused by varying local densities and the unreliability of Euclidean distances in high-dimensional space [59]. This consideration related to the distance relationships in high-dimensional space also led us to the approach of generating many microclusters first and then selecting a small number of representatives from each microcluster.

4.2.4 Stage 4: Generation of Multi-class Scatterplot Stimuli Pairs Based on Feature Groups. From Stage 3, we extracted 294 representative scatterplots. However, these scatterplots cover a wide range of values for each feature, and applying an exhaustive comparison over 70 features is challenging and infeasible. Therefore, we grouped the features that have strong correlations to narrow down our scope. Note that the correlation coefficients used for the previous stage were for feature selection to perform clustering, while this stage uses the correlation coefficients for grouping of features. In Stage 4, within 294 representative scatterplots, we first calculated Pearson's correlation coefficients for each pair of 70 features. We then selected pairs of features that have an absolute correlation coefficient over 0.5. We further merged the pairs if they share at least one feature (e.g. we merged a pair of Features A, B and a pair of Features B, C into one group as these pairs share Feature B). This grouping process produced *20 feature groups*. Each feature group has a different number of multi-class scatterplot features, ranging from 1 to 24. We list all the 20 feature groups in Table 1.

To further analyze how feature values can impact performance, we used the 20 feature groups as the baseline when selecting a *task pair*. We aimed to systemically select task pairs that have feature value differences as control variables. We selected task pairs in three steps through an iterative process. In the first step, we selected task pairs such that each pair has significantly different values for scatterplot features categorized in that feature groups (e.g., for Feature Group 1 in Table 1, a pair has both large and small $Overlap^{\text{convex}}$, while also having large differences in all values of $Area^{\alpha\text{-hull}}_{\text{max}}$, $Area^{\alpha\text{-hull}}_{\text{min}}$, C^{F2}, and $Overlap^{\alpha\text{-hull}}$). Second, for each feature group, we further identified task pairs that have similar values for features that are *not* in the current feature group (e.g., when considering Feature Group 1, a pair should have similar $Density^{\alpha\text{-hull}}$ and $Clumpy$, etc.). Lastly, we prioritized selecting task pairs that were not frequently chosen to avoid an unbalanced selection of some scatterplots. We repeated this process to select 30 task pairs for each of the 20 feature groups, resulting in 600 task pairs of scatterplots in total. The number of selections for each scatterplot ranged from 2 to 7 ($mean = 4.08, \sigma = 0.9$). We randomly divided the 600 task pairs into 10 batches (i.e., each batch contains 60 tasks). For a given task pair, two scatterplots were placed side-by-side in a random order.

4.3 Procedure

Our experiment had three phases: (i) informed consent and color-blindness screening, (ii) task description and tutorial, and (iii) the formal study.

In the first phase, participants provided their informed consent after reading our consent form following our IRB protocol and provided their demographic data. They were also asked to complete an Ishihara color-blind screening [42]. After successfully passing

Table 1: The 20 feature groups of the 70 multi-class scatterplot features. Each group contains a set of features that have high Pearson's correlation coefficients ($r > 0.5$) based on our dataset. These feature groups are used to generate task pairs for the user study. See Sec. 4.2 on how these feature groups were generated.

Group	List of features
Group 1	(1) $Area^{\alpha\text{-hull}}_{\max}$ (2) $Area^{\alpha\text{-hull}}_{\min}$ (3) $Overlap^{\text{convex}}$ (4) $Overlap^{\alpha\text{-hull}}$ (5) C^{F2}
Group 2	(1) $CentroidDiff^{\text{convex}}_{\max}$ (2) $CentroidDiff^{\text{convex}}_{\min}$
Group 3	(1) $Density^{\alpha\text{-hull}}_{\max}$ (2) $Density^{\alpha\text{-hull}}_{\min}$ (3) $\sigma^{Density^{\alpha\text{-hull}}}$
Group 4	(1) $Clumpy_{\max}$ (2) $Clumpy_{\min}$ (3) $CentroidDiff^{\alpha\text{-hull}}_{\min}$ (4) $CentroidDiff^{\alpha\text{-hull}}_{\min}$ (5) $DistributionOverlap_{\max}$ (6) $DistributionOverlap_{\min}$ (7) $DistributionDistance_{\max}$ (8) $DistributionDistance_{\min}$
Group 5	(1) $\sigma^{Monotonic}$
Group 6	(1) $N^{\text{points}}_{\max}$ (2) $N^{\text{points}}_{\min}$ (3) $\sigma^{N^{\text{points}}}$ (4) $Points/Classes$ (5) $Sparse_{\min}$
Group 7	(1) $Area^{\alpha\text{-hull}}_{\max}$ (2) $Area^{\alpha\text{-hull}}_{\min}$ (3) $\sigma^{Area^{\alpha\text{-hull}}}$ (4) $Skewed_{\min}$ (5) $Outlying_{\max}$ (6) $Outlying_{\min}$ (7) $Convex_{\max}$ (8) $Convex_{\min}$ (9) $Overlap^{\text{convex}}$ (10) $Overlap^{\alpha\text{-hull}}$
Group 8	(1) $Overlap^{\text{convex}}$ (2) C^{F2} (3) C^{F3} (4) C^{F4} (5) C^{LSC} (6) C^{Density}
Group 9	(1) $Kurtosis_{\max}$ (2) $Kurtosis_{\min}$
Group 10	(1) $N^{\text{points}}_{\max}$ (2) $\sigma^{N^{\text{points}}}$ (3) $Points/Classes$ (4) C^{Hubs} (5) C^{C1} (6) C^{C2}
Group 11	(1) $Monotonic_{\max}$ (2) $Monotonic_{\min}$
Group 12	(1) $Points/Classes$ (2) $Sparse_{\max}$ (3) $Sparse_{\min}$ (4) $Skinny_{\max}$ (5) $Skinny_{\min}$
Group 13	(1) $Split$ (2) $InnerOcclusionRatio$ (3) DSC (4) $GONG$ 0.35 $DIR\ CPT_{t0}$ (5) $GONG$ 0.35 $DIR\ CPT_{t1}$ (6) $density{-}awareDSC$ (7) $density{-}awareKNNG$ (8) C^{F2} (9) C^{F3} (10) C^{F4} (11) C^{L1} (12) C^{L2} (13) C^{L3} (14) C^{N1} (15) C^{N2} (16) C^{N3} (17) C^{N4} (18) C^{T1} (19) C^{LSC} (20) C^{Density} (21) C^{F1v} (22) C^{Hubs} (23) C^{C1} (24) C^{C2}
Group 14	(1) $Sparse_{\max}$ (2) $Sparse_{\min}$ (3) $Convex_{\max}$ (4) $Convex_{\min}$ (5) σ^{Convex} (6) $Skinny_{\max}$ (7) $Skinny_{\min}$
Group 15	(1) $Equidistant$
Group 16	(1) $N^{\text{points}}_{\max}$ (2) $\sigma^{N^{\text{points}}}$ (3) C^{L2} (4) C^{L3} (5) C^{N4} (6) C^{Density} (7) C^{Hubs} (8) C^{C1} (9) C^{C2}
Group 17	(1) $Stringy_{\max}$ (2) $Stringy_{\min}$ (3) $\sigma^{Stringy}$
Group 18	(1) $Area^{\alpha\text{-hull}}_{\max}$ (2) $Area^{\alpha\text{-hull}}_{\min}$ (3) $Skewed_{\max}$ (4) $Skewed_{\min}$ (5) $Outlying_{\max}$ (6) $Outlying_{\min}$ (7) $Convex_{\max}$ (8) $Convex_{\min}$ (9) $Overlap^{\alpha\text{-hull}}$ (10) $Overlap^{\text{convex}}$
Group 19	(1) σ^{Convex} (2) $Skinny_{\min}$ (3) σ^{Skinny}
Group 20	(1) $Area^{\alpha\text{-hull}}_{\max}$ (2) $Area^{\alpha\text{-hull}}_{\min}$ (3) $Sparse_{\max}$ (4) $Outlying_{\max}$ (5) $Convex_{\max}$ (6) $Convex_{\min}$ (7) $Skinny_{\max}$ (8) $Skinny_{\min}$ (9) $Overlap^{\alpha\text{-hull}}$ (10) C^{ClsCoef}

the color-blind screening, participants were led to a tutorial page that described our target experimental task.

In the second phase, we provided a series of icons to illustrate different degrees of VCS for the tutorial. We chose to use icons instead of scatterplots to prevent biasing the participants' responses. Participants were required to successfully answer three easy tutorial tasks by indicating *which scatterplot (A or B) has blues and oranges that are more separated?* This tutorial task uses a two-alternative forced choice (2AFC) and is representative of the experimental task, serving to remove any possible ambiguity from the task description. After successfully answering all three tutorial tasks, participants moved on to the formal study phase.

During the formal study phase, participants assessed VCS for 64 pairs of stimuli (60 formal trials and 4 engagement checks). Each participant was randomly assigned to one of the 10 batches. The task pairs in each batch were presented in a randomized order. We collected 15 participants for each batch. Participants had 20 seconds to respond to each task pair. If they did not respond within that time window, their answer was considered to be `N/A`, and the study moved on to the next task pair. Participants either clicked on the provided radio buttons or used the left/right keyboard arrow keys to input their responses. We employed four engagement checks to ensure valid participation. These engagement checks were stimuli that had a clear class separation with a *GONG* 0.35 *DIR CPT* score [9] of at least 0.6 difference. We randomly placed these engagement checks throughout the 60 formal task pairs.

4.4 Participants

We recruited 152 participants from Amazon's Mechanical Turk. Two participants failed three out of four engagement checks and were excluded from the analysis due to insufficient attention or lack of understanding of the task. We analyzed data from the remaining 150 participants (91 male, 59 female; 24–65 years of age). All participants were from the United States and Canada, had at least a 95% approval rating, and reported having normal or correct-to-normal vision. The experiment lasted an average of 15 minutes.

4.5 Analysis

We used participants' selection differences over each task pair as our primary dependent measure. To measure how scatterplot features influence people's selection on VCS tasks, we divided our experimental data into two parts: *all-data* (includes 70 features) and *by-group* (see Table 1). For each task pair, we computed the difference in feature values (subtracting feature values of the less frequently selected scatterplot from those of the more frequently selected scatterplot) as our independent variables. We analyzed the resulting data using F-tests with standard least squares and linear regression. All post-hoc analyses used Tukey's Honest Significant Difference Test (HSD, $\alpha = 0.05$).

5 Results

We measure how different scatterplot features can influence people's perception of VCS tasks and identify which features are important for such tasks. Our analysis focuses on three primary questions: (Q1) How do existing VCS measures perform with respect to human judgments? (Q2) Which individual scatterplot features are significant to VCS tasks? (Q3) What combination of multiple features are significant to VCS tasks? We report significant effects of features relative to our two hypotheses. We provide full data tables in our supplemental materials. We discuss significant results and statistical analysis based on the scatterplot features using both traditional inferential measures and 95% bootstrapped confidence intervals (± 95% CI) for fair statistical communication.

5.1 Analysis 1: Human Perception versus Visual Class Separation Measures

We analyze participants' responses to first evaluate their consistency across task pairs. This analysis is necessary given how there is no ground truth to compare participants' answers against. Hence, analyzing participants' responses can help establish a baseline of the general patterns of human perception for VCS tasks. We conduct three analyses for this objective: understanding alignment among participants, computing alignment between participants' responses and VCS measures, and understanding how the magnitude of computed VCS measures informs perceived separation.

5.1.1 How well aligned are people with each other? First, we computed the difference in participants' selection within a task pair (i.e., selection difference). A task pair requires a binary selection (i.e., a two-alternative forced choice between scatterplot A or B). Selections across all 600 task pairs would highlight a preference majority for certain features. A higher selection difference value would indicate a majority alignment among participants' perceived class separation. We computed this selection difference by taking the absolute value of the difference of the number of times participants selected A versus B. The result is shown in Fig. 6. Across all 600 task pairs, the average selection difference for each task pair is 9.716 ($\sigma = 4.591$) from the possible maximum of 15. This result implies that while there is a general agreement on how VCS in the scatterplots was perceived, there is not a universal consensus.

5.1.2 How well do people's perceptions align with GONG 0.35 DIR CPT? To further investigate this divergence, we used *GONG* 0.35 *DIR CPT* as a proxy ground truth to compare alignment against. As discussed in Sec. 2.2, Aupetit & Seldmiar [9] noted that *GONG* 0.35 *DIR CPT* is the best-performing VCS measure for their tested set of 828 scatterplots. Participants' averaged 67.6% accuracy ($\sigma = 31.2$; 95% $CI = [65.1, 70.1]$) when using *GONG* 0.35 *DIR CPT* as a proxy ground truth. This result highlights a 32.4% mismatch between *GONG* 0.35 *DIR CPT* measures and human perception, indicating that *GONG* 0.35 *DIR CPT* likely fails to account for a subset set of multi-class scatterplot features that are important for human VCS.

5.1.3 Do large GONG 0.35 DIR CPT differences lead to higher agreement with human judgments? To understand mismatches between

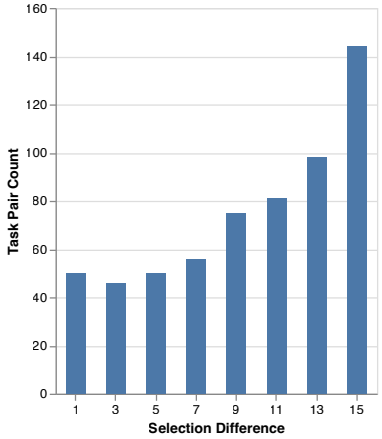


Figure 6: Selection difference for 600 task pairs. Each task pair has 15 responses. For example, the selection difference of 11 represents 13 people selecting one class and 2 selecting another. 80 task pairs had this 13:2 ratio in their responses.

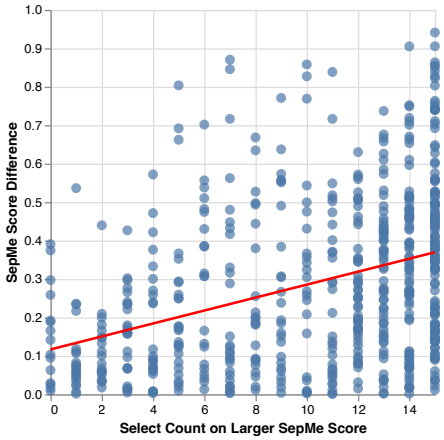


Figure 7: Comparison of the number of selections on a scatterplot having a larger *GONG* 0.35 *DIR CPT* score (x-axis) and the *GONG* 0.35 *DIR CPT* score difference (y-axis) for task pairs. The red line depicts the line of best fit, showing a weak correlation ($r = 0.36$).

human perception and *GONG* 0.35 *DIR CPT*, we performed a secondary analysis to determine whether scatterplot pairs with larger differences in *GONG* 0.35 *DIR CPT* scores (i.e., stronger predicted separation) would lead to greater agreement amongst participants. We posit that a larger difference in *GONG* 0.35 *DIR CPT* for a task pair should result in clearer separation and, therefore, more consistent human responses. For example, consider two hypothetical task pairs where each scatterplot has a *GONG* 0.35 *DIR CPT* score: the first task pair has *GONG* 0.35 *DIR CPT* scores of $[0.8, 0.2]$ and the other has $[0.4, 0.2]$. In this case, the first task pair should lead to higher agreement.

We compared responses with *GONG* 0.35 *DIR CPT* scores by calculating the difference in *GONG* 0.35 *DIR CPT* scores for each task pair and then computing the Pearson correlation coefficient between the score differences and the number of participants selecting the scatterplot with the larger *GONG* 0.35 *DIR CPT* score (Fig. 7). *GONG* 0.35 *DIR CPT* scores were weakly correlated with participant response frequencies ($r = 0.36$). This result also indicates that *GONG* 0.35 *DIR CPT* may insufficiently capture human perception for VCS tasks.

5.1.4 Summary of Analysis 1: People's assessment of visual class separation is largely in alignment but fails to be in universal agreement for some conditions. Existing VCS measures such as *GONG* 0.35 *DIR CPT* are not well-aligned with people's perceptions of relative class separation.

5.2 Analysis 2: Identifying Individual Features Significant to Visual Class Separation

Building upon Analysis 1, we investigate which individual features impact human judgments on VCS tasks that existing VCS measures insufficiently account for by using two tests (F-test and F-regression) applied across all the experimental data. Please check our supplementary materials for details of these two test results.

5.2.1 Which individual features influence human performance for VCS tasks? First, we performed an F-test for all 70 features across the 600 task pairs. Only three features (C^{F1v}, C^{N1}, and C^{C1}) have significant effects on selection difference ($p < 0.05$, refer to Fig. 8). All three features are classification complexity measures (Sec. 3.2). C^{F1v} is an axis-based feature that evaluates classification difficulty based on the distance of class centroids relative to the positional distribution of points within each class (specifically, the maximum Fisher's discriminant ratio [58]). It generally describes `Class Separation` as the degree of overlap between the two classes. See Fig. 9 as an example, which presents two task pairs with small and large values of C^{F1v}. C^{N1} is a neighborhood feature that computes the fraction of points near the class boundary. People viewed scatterplots with smaller C^{F1v} and C^{N1} values as more separated (Fig. 8). C^{C1} is a class imbalance feature that computes the entropy of class proportions. While the F-tests indicates the significant effects C^{C1}, it does not show clear effects as C^{F1v} and C^{N1}, as shown in Fig. 8.

Both C^{F1v} and C^{N1} consider the overlap between two classes and the degree of complexity (e.g., computational or geometric) required to separate them. Lorena et al. [58] note that lower values in these measures indicate that the data represents simpler classification problems, which can be easily separated using statistical approaches (Fig. 3-a, c). Though these classification complexity measures were originally intended for machine learning models, these results highlight how these measures also reasonably match human perception.

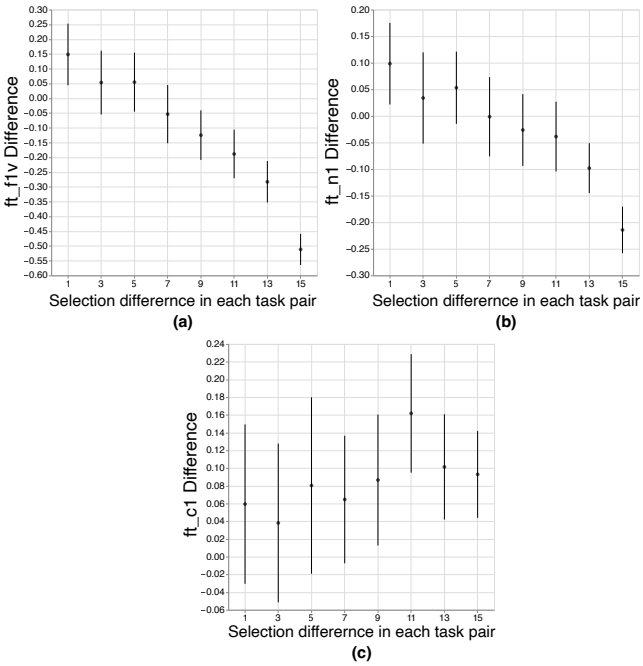


Figure 8: The average feature differences for selection differences per task pair. Three features: C^{F1v}, C^{N1}, and C^{C1} best capture human perception for VCS tasks based on our F-test analysis. The figures show that people tend to consider a scatterplot to be more visually separated with smaller C^{F1v} and C^{N1} features. Error bars represent 95% confidence intervals.

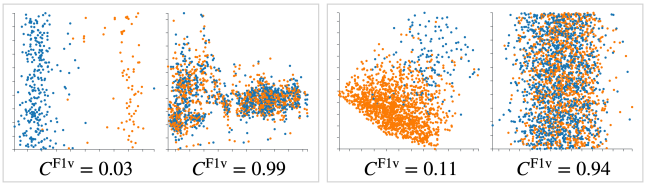


Figure 9: Two task pair examples that have large value differences for C^{F1v}. All 15 participants selected the scatterplots that have smaller C^{F1v} values for these two examples.

To evaluate the significance of individual features, we also performed F-regression for all 70 features across the 600 task pairs. As listed in Table 2, 37 features have at least a $p < 0.05$, indicating slightly over half of the multi-class scatterplot features (37/70) may influence VCS tasks. Out of these 37 features, 27 are the instantiated features from Sedlmair et al. [78], confirming how the conceptual features proposed by their taxonomy influence VCS tasks. These results confirm **H1**: scatterplot features impact people's perception of VCS tasks, and that multiple features correlate with class separability.

Table 2: 37 features with significant effects on human perception for VCS tasks (Sec. 5.2.1). These features are identified with F-regression ($p < 0.05$). The Pearson's correlation coefficients larger than 0.3 or smaller than -0.3 (i.e., $|r| > 0.3$) are bolded.

Feature name	Feature category	r
C^{F1v}	Axis	**-0.504**
DSC	Class Separation	**0.465**
$GONG\ 0.35\ DIR\ CPT_{t1}$	Class Separation	**0.439**
$density{-}awareDSC$	Class Separation	**0.408**
C^{F4}	Axis	**-0.399**
C^{F3}	Axis	**-0.366**
C^{N1}	Neighborhood	**-0.328**
C^{N3}	Neighborhood	**-0.324**
$density{-}awareKNNG$	Class Separation	**0.323**
C^{T1}	Neighborhood	**-0.312**
C^{LSC}	Neighborhood	**-0.302**
C^{L1}	Neighborhood	-0.297
C^{L3}	Neighborhood	-0.294
$Split$	Split	0.271
$InnerOcclusionRatio$	Inner-Outer Position	-0.269
C^{L2}	Neighborhood	-0.268
C^{Density}	Network	-0.246
$Points/Classes$	Class/Point Count	-0.229
$N^{\text{points}}_{\text{max}}$	Count	-0.227
$Skinny_{\text{min}}$	Shape	-0.214
C^{N4}	Neighborhood	-0.204
$\sigma^{N^{\text{points}}}$	Variance of Count	-0.191
σ^{Convex}	Variance of Shape	-0.184
$Sparse_{\text{min}}$	Density	0.179
$Sparse_{\text{max}}$	Density	0.174
$Kurtosis_{\text{min}}$	Centroid	-0.173
$Skinny_{\text{max}}$	Shape	-0.164
$GONG\ 0.35\ DIR\ CPT_{t0}$	Class Separation	0.164
σ^{Skinny}	Variance of Shape	0.153
$CentroidDiff^{\text{convex}}$	Centroid	-0.141
$Convex_{\text{max}}$	Shape	0.137
$Clumpy_{\text{max}}$	Clumpiness	0.129
$N^{\text{points}}_{\text{min}}$	Count	-0.123
$Outlying_{\text{min}}$	Outlier	-0.116
$DistributionDistance_{\text{max}}$	Centroid	0.111
$Overlap^{\text{convex}}$	Centroid	-0.110
$\sigma^{Area^{\alpha\text{-hull}}}$	Variance of Size	0.105

We assess how well these features individually correlate with VCS using Pearson's correlation. 11 of the 37 features have a Pearson correlation coefficient of $r > 0.3$ (at least moderate correlation) and $p < 0.05$. 7 of these 11 features are *axis* and *neighborhood* measures and are derived from the classification complexity measures from Lorena et al. [58]. The remaining 4 are VCS measures. This result further supports **H2**, illustrating how classification complexity and VCS measures can quantify perceived separability.

5.2.2 Considering feature group data, which individual features influence human perception for VCS tasks? The 20 feature groups in Table 1 each represent a set of features that are highly correlated to each other (Sec. 4.2.4), but we lack insight as to whether these features are correlated to *participants' scatterplot selection*. The F-test and F-regression in Sec. 5.2.1 help determine which features

Table 3: Selected F-regression test results computed for each feature group (Sec. 5.2.2). The seven feature groups that have at least one feature has p-value < 0.05 from our F-regression computation. The table shows the corresponding Pearson's correlation coefficients for features considered significant to the VCS tasks (i.e., $p < 0.05$).

Feature group	Feature name	Feature category	r
Group 4	$Clumpy_{\text{max}}$	Clumpiness	0.446
	$DistributionOverlap_{\text{max}}$	Centroid	-0.412
	$DistributionOverlap_{\text{min}}$	Centroid	-0.406
	$DistributionDistance_{\text{max}}$	Centroid	0.399
	$DistributionDistance_{\text{min}}$	Centroid	0.395
Group 8	C^{LSC}	Neighborhood	-0.503
	C^{F3}	Axis	-0.481
	C^{F4}	Axis	-0.465
	C^{Density}	Network	-0.442
	$Overlap^{\text{convex}}$	Class Separation	-0.425
Group 10	$N^{\text{points}}_{\text{max}}$	Count	-0.430
	$Points/Classes$	Class/Point Count	-0.429
	$\sigma^{N^{\text{points}}}$	Variance of Count	-0.429
Group 12	$Points/Classes$	Class/Point Count	-0.639
	$Skinny_{\text{min}}$	Shape	-0.635
	$Skinny_{\text{max}}$	Shape	-0.589
	$Sparse_{\text{min}}$	Density	0.515
	$Sparse_{\text{max}}$	Density	0.440
Group 13	C^{Density}	Network	0.517
Group 14	σ^{Convex}	Variance of Shape	0.530
	$Skinny_{\text{min}}$	Shape	-0.445
	$Skinny_{\text{max}}$	Shape	-0.393
Group 20	$Skinny_{\text{max}}$	Shape	-0.519
	$Area^{\alpha\text{-hull}}_{\text{min}}$	Size	0.504
	$Convex_{\text{min}}$	Shape	0.484
	$Convex_{\text{max}}$	Shape	0.445
	$Skinny_{\text{min}}$	Shape	-0.434

are most closely related to VCS but can introduce noise given how all 70 features are treated as independent variables. We performed an F-regression test for each *feature group* to mitigate the influence of other feature groups—and inevitably other features—introducing a more robust measure for identifying the influence of independent features for VCS, aligning with our study's task pair design.

7 of the 20 feature groups have at least one feature that significantly correlates with perceived `Class Separation`. Table 3 shows these 7 feature groups and their corresponding r values. For example, in Group 4, *Clumpy*, *DistributionOverlap*, and *DistributionDistance* focus on capturing the non-Gaussianity of point distributions. We can infer that participants perceived VCS differently based on whether the distributions are Gaussian or not. In Group 8, C^{LSC}, C^{F3}, and C^{F4} show negative correlations: people think two classes are more separated when general overlap region is small. Group 10 shows that point `Count` also matters, indicating humans perceive VCS differently if there is a class imbalance or a large number of points in a scatterplot. Groups 12, 13, 14, and 20 show that features related to *Skinny*, *Sparse*, *Convex*, and C^{Density} correspond to `Point Distance` and `Shape`. As all VCS measures

are included in Group 13 (see Table 1), the correlation of Group 13 with VCS tasks also indicates that all VCS measures quantify human performance on VCS tasks to a certain degree.

5.2.3 Summary of Analysis 2: Classification complexity features largely impact human perception for VCS tasks. Features related to `Count`, `Variance of Count`, `Density`, `Clumpiness`, `Shape`, `Centroid` (or non-Gaussianity), `Class Separation` also have a strong correlation to people's perceived class separation.

5.3 Analysis 3: Identifying Top-Ranking Set of Features

Features that significantly impact human perception of VCS likely orchestrate in ensembles rather than as a single feature. This behavior is suggested based on the inability of any single feature to fully reflect human performance in Analysis 2. Analysis 3 focuses on uncovering the feature combinations that influence VCS. Identifying the set of features that best explain perceived VCS and their corresponding weights can inform future VCS-related studies by informing critical feature combinations and offering new considerations for VCS models. To determine feature sets significant to VCS tasks, we employed a feature selection process and built models based on our experimental data.

5.3.1 What is the top-ranking set of features? First, we employed *SequentialFeatureSelector* from scikit-learn [68] to rank individual candidate features. This machine learning method performs sequential feature selection by choosing to add or remove features in a greedy fashion. Our implementation iteratively added one best-scoring feature while calculating the R^2 score based on currently selected features. We stopped adding features to our selection if R^2 did not improve after adding more features. This process resulted in a set of 23 features, which partially supports **H2**: 9 of these features contain features related to VCS and classification complexity measures (e.g., C^{F1v}, $GONG$ 0.35 $DIR\ CPT_{t0}$, $Overlap^{\text{convex}}$) while the remaining 14 features are within-class features.

We used these 23 features as the base and incorporated two F-regression results from Analysis 2—one for *all data* (Sec. 5.2.1) and *feature group data* (Sec. 5.2.2)—to model our data. Only three additional features, C^{LSC}, C^{F3}, and C^{F4}, were considered significant in both F-regression results. These three features were added to the initial list of 23, resulting in a final set of 26 features (see Table 4).

5.3.2 A composite feature model. We obtain weights for each of the 26 features to model participants' overall VCS scores by using *Epsilon-Support Vector Regression* with the linear kernel (linear SVR). Applying linear SVR, we built a composite feature from the feature set to explore the integrated effects on human judgments. We selected linear SVR, instead of conventional linear regression, as linear SVR is more robust to outliers [11]. To validate the composite feature, we computed Pearson correlation coefficients on selected weighted features and compared them to the computed coefficients from a single-feature analysis. Our results (Table 4) indicate that the Pearson correlation coefficient for the composite feature ($r = 0.696$) is significantly more correlated with participant responses than any individual feature (e.g., C^{F1v}, $r = -0.504$). This result indicates that people are influenced by multiple features related to VCS tasks and that combining multiple features can lead to a better VCS measure.

Table 4: The 26 features selected through the iterative feature selection (Sec. 5.3). Note: uni.—univariate, comp.—composite.

Rank	Feature	Feature type	Weight	r (uni.)	r (comp.)
1	C^{F1v}	classif. complex.	-1.109	-0.504	0.504
2	$GONG$ 0.35 $DIR\ CPT_{t1}$	between-class	-1.279	0.439	0.554
3	$Skinny_{\text{min}}$	within-class	0.959	-0.214	0.618
4	$Kurtosis_{\text{min}}$	within-class	3.489	-0.173	0.639
5	$Clumpy_{\text{max}}$	within-class	-1.137	0.129	0.656
6	$Overlap^{\text{convex}}$	between-class	0.130	-0.110	0.667
7	$InnerOcclusionRatio$	between-class	0.993	-0.269	0.675
8	$DistributionDistance_{\text{min}}$	within-class	-0.929	0.050	0.678
9	$Sparse_{\text{max}}$	within-class	-1.228	0.174	0.681
10	$Outlying_{\text{min}}$	within-class	-1.966	-0.116	0.683
11	$Area_{\text{max}}^{\alpha\text{-hull}}$	within-class	0.445	0.063	0.683
12	$Sparse_{\text{min}}$	within-class	-2.022	0.179	0.684
13	$DistributionDistance_{\text{max}}$	within-class	0.814	0.111	0.685
14	$Kurtosis_{\text{max}}$	within-class	-2.024	0.065	0.686
15	DSC	between-class	-1.246	0.465	0.688
16	$CentroidDiff_{\text{min}}^{\alpha\text{-hull}}$	within-class	0.493	0.020	0.689
17	C^{C1}	classif. complex.	2.403	0.057	0.689
18	$CentroidDiff_{\text{min}}^{\text{convex}}$	within-class	-3.629	-0.141	0.690
19	$Skewed_{\text{min}}$	within-class	-1.888	-0.056	0.691
20	$Skewed_{\text{max}}$	within-class	-0.314	-0.012	0.691
21	$Split$	between-class	-0.586	0.271	0.692
22	C^{C2}	classif. complex.	0.659	-0.067	0.692
23	$\sigma^{Stringy}$	between-class	3.597	0.051	0.693
24	C^{LSC}	classif. complex.	-0.151	-0.503	0.693
25	C^{F3}	classif. complex.	0.999	-0.481	0.693
26	C^{F4}	classif. complex.	-0.828	-0.465	0.695

This compositing approach can also improve existing measures. For example, the correlation coefficient for using $GONG$ 0.35 $DIR\ CPT$ alone is $r = 0.439$. After adding C^{F1v} improves to $r = 0.554$, and adding three more features further increases the correlation to $r = 0.656$ (see Table 4).

The signs of the weights may help us understand how features interact with one another. However, interpreting these signs is not straightforward due to the correlations for each single feature (cf. Table 1). For example, both C^{F1v} and $GONG$ 0.35 $DIR\ CPT_{t1}$ are negative in the final composite feature (Table 4). In contrast, their signs were negative and positive respectively if our composite feature only considers these two features. While iteratively adding features, the sign of $GONG$ 0.35 $DIR\ CPT_{t1}$ flipped from positive to negative after adding $Kurtosis_{\text{min}}$. Hence, instead of focusing on each feature's sign and weight, we rather consider whether the composite feature collectively utilizes the strength of each single feature to better align with human perception.

To evaluate our composite feature model, we employed a Monte Carlo cross-validation [102] with our user study data where we randomly selected 80% of the data as the training data and 20% data as the testing data for 100 iterations. For each iteration, we followed the same approach in Sec. 5.3.1. We used the training set to select the top 23 features and incorporated three features (C^{LSC}, C^{F3}, and C^{F4}) to generate a composite feature. Note that the top 23 features can vary in each iteration and do not necessarily match the set of 23 features mentioned in Sec. 5.3.1. After obtaining the composite feature, we used the composite feature to predict perception for

VCS tasks with the testing set. From these 100 iterations, C^{F1v} was ranked first for all trials. *GONG* 0.35 *DIR CPT*$_{t1}$ was ranked second for 89 trials and ranked third for 10 trials. $Skinny_{\text{min}}$ ranked third for 85 trials and ranked second for 5 trials. These three features were consistently to be the most predictive.

The generated models had an accuracy average of 84.2% (σ = 3.04; 95% $CI = [83.6, 84.8]$) over the 100 trials, significantly outperforming *GONG* 0.35 *DIR CPT*, which had the accuracy of 67.6% (Sec. 5.1.2). Table 5 shows example task pairs that a single scatterplot feature such as C^{F1v} and *GONG* 0.35 *DIR CPT* can predict VCS perception in certain conditions but fail in others. In contrast, our composite feature model provided correct predictions for all tested task pairs.

5.3.3 Summary of Analysis 3: The third analysis validates that we can generate an improved measure of VCS task performance by including multiple features from best-performing feature sets. By integrating results from Analyses 2 and 3, we can generate a composite feature with our final feature set. The cross-validation results show a consistent set of top-ranking features with respect to our composite feature and demonstrate a 16.6% accuracy improvement when compared to only using *GONG* 0.35 *DIR CPT* (cf. Sec. 5.1.2). The improvement illustrates how our composite feature approach can better evaluate human perception of VCS tasks on various data distributions.

6 Discussion

6.1 Visual Class Separation Measures: Key Features and Performance Difference to Human Perception

This paper's goal is twofold: (1) investigate what are key features that influence human perception of VCS; (2) Determine whether existing VCS measures align with human perception. For the first goal, our analysis reveals that top-ranking features come from diverse sources. Notably, within-class features dominated over between-class features. Second, our results show that while there are certain mismatches between existing VCS measures and human perception, our composite feature model can reduce such gap by integrating multiple features.

6.1.1 What are the key scatterplot features? We address our first research question through three sets of analyses (Sec. 5) which also enabled us to create a composite feature model with a final set of 26 features that quantifies perceived VCS better than any individual feature (Table 4). As discussed in Sec. 5.3, predicting human perception in VCS tasks requires models to consider multiple scatterplot features. First, the top five features in our study do not come from a single source—rather they reflect diverse sources. These features are instances of classification complexity measures (#1) [58], VCS measures (#2) [9], scagnostics [89] (#3, #5), and non-Gaussianity (#4). This result largely reflects Sedlmair et al.'s perspective that VCS cannot be singularly evaluated across different scatterplots. Even though VCS tasks may appear straightforward (i.e., is Class A well separated from Class B?), the diversity of these measures indicates the range of intertwined cognitive processes required to accomplish this task. Furthermore, we note that the top-ranking feature is C^{F1v}: an axis-based classification complexity measure. As mentioned in Sec. 3.2, axis-features characterize how easily classes can be separated based on an arbitrary direction in a multi-class scatterplot. Specifically, C^{F1v} measures the maximum linear boundary margin between classes based on the classes' centroids. This approach shares a similarity with the class centroid-based VCS measures, such as *DSC*. However, existing centroid-based VCS measures focus more on the number of points placed close to each class centroid (i.e., point-based), whereas C^{F1v} considers the dispersion of points (i.e., distribution-based). This difference suggests that future VCS measures should incorporate information on the data distributions.

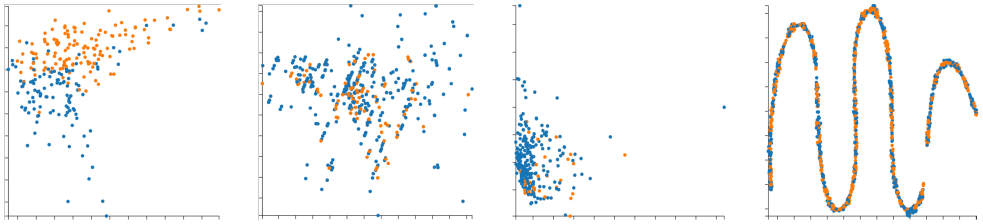
Although the top-ranking feature is C^{F1v}, we note that classification complexity features, as a whole, performed well in correlation tests (Analysis 2) but not in feature selection (Analysis 3). After computing F-regression and Pearson's correlation coefficient with all experimental data (Sec. 5.2), 7 out of 11 selected features are related to classification complexity. However, after applying *SequentialFeatureSelector*, only 3 were selected as key predictors (#1, #17, #22). This result highlights that some classification complexity measures have high *dependence* on each other, resulting in potential redundancy. For example, C^{F1v} conceptually shares many similarities with C^{N1} when two classes fall between "partial overlap" and "separate" (Fig. 2-d2). As a result, once a complexity measure has been selected by a feature selector (e.g., C^{F1v}), similar measures, such as C^{N1}, may not improve the regression model since C^{F1v} satisfies C^{N1}'s role.

We found that the slight majority—14 out of 26 features—are "within-class" features. The result may seem counter-intuitive given how when we conceptually consider VCS, we would think task performance would stem from the feature interactions or differences *between* the two classes. Rather, our results highlight that within-class features, particularly min/max attributes, dominate more than between-class features. We speculate that this result stems from the perceptual operations required for VCS tasks, such as perceptual organization and ensemble coding. The visual features created by the distribution of data along the different dimensions can leverage the global (i.e., ensemble coding) and local (i.e., perceptual organization) visual configuration of a multi-class scatterplot: people may reason over both broader structures characterizing the full class as well as individual points. Szafir et al. [81] discuss how ensemble coding can help people quickly estimate the position of a group of scatterplot points without attending each point individually. Given the short-duration nature of our user study, we speculate similar mechanisms also translate for VCS tasks. The interplay of local encoding (i.e., within-class features) influences the perceived global structure, which can affect people's scatterplot selection.

6.1.2 Do existing VCS measures align with human perception? Of the 70 features, we included four existing VCS measures, including *GONG* 0.35 *DIR CPT*. Past studies highlight how *GONG* 0.35 *DIR CPT* is best-performing VCS measure [9], and it was ranked second within our final set of weighted features. However, results from Analysis 1 show that there was 32% mismatch between *GONG* 0.35 *DIR CPT* and human perception, highlighting the opportunity to improve the effectiveness of VCS measures. Despite the results from Analysis 1, the final set of weighted features highlights the possibility of *GONG* 0.35 *DIR CPT* supplementing

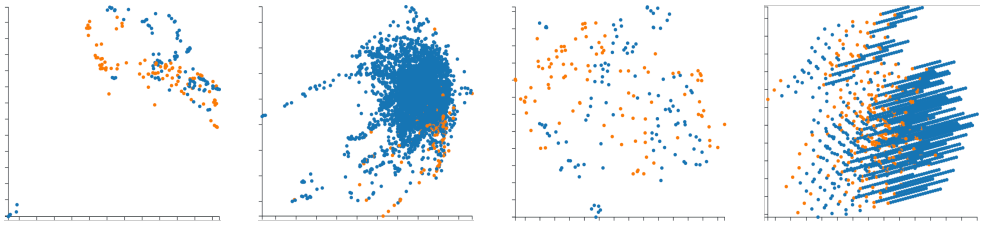
Table 5: Comparison of C^{F1v}, *GONG* 0.35 *DIR CPT*, and our composite feature derived from the 26 features. For C^{F1v}, we compute $(1 - C^{\text{F1v}})$ to make the comparison easier: i.e., larger value, clearer VCS. For task pairs in (a), while C^{F1v} correctly identifies a scatterplot with the majority votes, *GONG* 0.35 *DIR CPT* does not. In (b), we can see the opposite pattern. However, our composite variable provides correct predictions for all these examples.

(a) C^{F1v} agrees with participant votes



	Task pair 1		Task pair 2	
Participant votes	**14**	1	**13**	2
1 - C^{F1v}	**0.72**	0.05	**0.39**	0.00
GONG 0.35 *DIR* CPT_{t1}	0.77	**0.86**	0.11	**0.58**
Our composite feature	**2.52**	-0.19	**-0.54**	-2.24

(b) *GONG* 0.35 *DIR* CPT_{t1} agrees with participant votes



	Task pair 3		Task pair 4	
Participant votes	**14**	1	**14**	1
1 - C^{F1v}	0.06	**0.62**	0.15	**0.45**
GONG 0.35 *DIR* CPT_{t1}	**0.81**	0.10	**0.86**	0.20
Our composite feature	**-1.25**	-3.31	**1.02**	-1.66

C^{F1v}, likely by providing information that C^{F1v} cannot capture (e.g., neighborhood relationships). These features employ neighbor-based and class centroid-based approaches, respectively. Consequently, this result suggests a hybrid design of the two approaches as a potential way to better develop VCS measures and use them for robust measures. Our cross-validation demonstrates how our composite feature model can achieve an 84.2% average accuracy for predicting VCS performance, exceeding 16.6% accuracy compared to the best-performing existing VCS measures.

6.2 Limitations and Future Work

Given the sizes and variance of real-world datasets, we reduced our dataset size to a manageable scale in order to conduct our user study. Despite this down-scale, it was still infeasible to perform an exhaustive combination of task pairs. We mitigated this challenge by picking task pairs that adhered to certain criteria to maximize the range of possible feature values within our study. However, this approach does not provide full coverage of the space of feature values. Future work should extend these results to additional scatterplots, potentially using our results as a means to identify candidate features to explore in greater detail.

We encoded classes for our experimental stimuli using common categorical colors: blue and orange. However, prior studies indicate colors can influence people's graphical perception [80, 87]. Future work should investigate how color encoding or other means for class representation might change perception for VCS tasks.

Our study mainly used scatterplots with less than 10,000 points. Given how people typically analyze much larger datasets when using dimensionality reduction methods, future work should consider the challenges of larger datasets for VCS. As noted in Sec. 6.1.2, increasing the number of points can help uncover how scale affects underlying perceptual mechanisms (e.g., ensemble coding, perceptual organization) and subsequent VCS perceptions. Although our crowdsourced user study used scatterplots with fewer than 10,000 points, our experiments cover a wide range of point counts from as low as 50 up to 9,999. Additionally, our task pairs also cover conditions where there are diverse differences in point counts. While features related to point counts have some impact on VCS tasks, they were not among the top influential features (cf. Sec. 5.2). Future work can use our study as a reference to understand the relationship between performance and scale.

Lastly, our user study results rely on the class complexity measures and our instantiated measures of Sedlmair et al's conceptual features [78]. We acknowledge that there might be essential scatterplot features that were not captured by these measures. Further research effort is required to more comprehensively extract multi-class scatterplot features and understand the relationships to VCS.

One way of uncovering these features is a bottom-up method: conduct the aforementioned perceptual studies to gather data on how humans perform VCS and then develop new measures.

6.3 Call for Future Research

We encourage future work to build upon our efforts and further investigate our second research question. From our study, we were unable to provide a conclusive answer given our two analysis results. Future research can use our 70 features and scatterplot generation method (Sec. 4.2) as an extension of this work.

Aligning to our third key observation, future work should investigate the interrelational effect between local and global structures of a multi-class scatterplot for VCS tasks. Research highlights how this interplay is common for other conventional visualizations, such as networks [48, 60]. Analysts must also simultaneously attend to the local and global structure of a network for a given network task (e.g., estimating the size of a network [54]) We encourage future work to leverage and apply these existing methods as another step towards robustly measuring and developing VCS measures. As a concrete suggestion, we recommend future work to measure how quickly users can orient themselves to the global structure of the multi-class scatterplot, such as the distribution of scatterplot points, sizes, colors, and orientations.

7 Conclusion

We investigate (1) what are the key scatterplot features that influence human perception of VCS and (2) whether existing VCS measures align with human perception. We conducted a crowdsourcing user study with 150 participants to evaluate 294 representative scatterplots and 70 multi-class scatterplot features. Our statistical analyses not only uncovered strong associations among these 70 measures and participants' scatterplot selections but also opens a new set of questions for researchers to further investigate.

Acknowledgments

This research is sponsored in part by the U.S. National Science Foundation through grants IIS-2040489 and IIS-2320920, the Knut and Alice Wallenberg Foundation through Grant KAW 2019.0024, and the CU Boulder Engineering Education and AI-Augmented Learning Interdisciplinary Research Theme Seed Grant. This work was authored (in part) by the National Renewable Energy Laboratory, operated by the Alliance for Sustainable Energy, LLC, for the US Department of Energy (DOE) under contract no. DE-AC36-08GO28308.

References

[1] Mostafa M Abbas, Michaël Aupetit, Michael Sedlmair, and Halima Bensmail. 2019. ClustMe: A visual quality measure for ranking monochrome scatterplots based on cluster patterns. *Comput Graph Forum* 38, 3 (2019), 225–236. https://doi.org/10.1111/cgf.13684

[2] Abubakar Abid, Martin J Zhang, Vivek K Bagaria, and James Zou. 2018. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nat Commun* 9, 1 (2018), 2134. https://doi.org/10.1038/s41467-018-04608-8

[3] Dimitris Achlioptas. 2001. Database-friendly random projections. In *Proc. PODS*. ACM, New York, NY, USA, 274–281. https://doi.org/10.1145/375551.375608

[4] Georgia Albuquerque, Martin Eisemann, Dirk J Lehmann, Holger Theisel, and Marcus Magnor. 2010. Improving the visual analysis of high-dimensional datasets using quality measures. In *Proc. VAST*. IEEE, New York, NY, 19–26. https://doi.org/10.1109/VAST.2010.5652433

[5] Edesio Alcobaça and Felipe Siqueira. 2024. PyMFE. https://pymfe.readthedocs.io/en/latest/index.html. Accessed: 2024-09-04.

[6] Dan Ariely. 2001. Seeing sets: Representation by statistical properties. *Psychol Sci* 12, 2 (2001), 157–162. https://doi.org/10.1111/1467-9280.00327

[7] F. Gregory Ashby and W. William Lee. 1993. Perceptual variability as a fundamental axiom of perceptual science. In *Foundations of Perceptual Theory*, Sergio C. Masin (Ed.). Advances in Psychology, Vol. 99. North-Holland, Amsterdam, Netherlands, 369–399. https://doi.org/10.1016/S0166-4115(08)62778-8

[8] Michaël Aupetit, Pierre Couturier, and Pierre Massotte. 2002. γ-observable neighbours for vector quantization. *Neural Netw* 15, 8-9 (2002), 1017–1027. https://doi.org/10.1016/S0893-6080(02)00076-X

[9] Michael Aupetit and Michael Sedlmair. 2016. SepMe: 2002 New visual separation measures. In *Proc. PacificVis*. IEEE, New York, NY, 1–8. https://doi.org/10.1109/PACIFICVIS.2016.7465244

[10] Burcu Avci and Aysecan Boduroglu. 2021. Contributions of ensemble perception to outlier representation precision. *Atten Percept Psychophys* 83 (2021), 1141–1151. https://doi.org/10.3758/s13414-021-02270-9

[11] Mariette Awad and Rahul Khanna. 2015. Support Vector Regression. In *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Apress, Berkeley, CA, 67–80. https://doi.org/10.1007/978-1-4302-5990-9_4

[12] Michael Behrisch, Michael Blumenschein, Nam Wook Kim, Lin Shao, Mennatallah El-Assady, et al. 2018. Quality metrics for information visualization. *Comput Graph Forum* 37, 3 (2018), 625–662. https://doi.org/10.1111/cgf.13446

[13] Kenneth E. Bellock. 2019. Alpha Shape Toolbox. https://github.com/bellockk/alphashape. Accessed: 2024-09-12.

[14] Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Michael Sedlmair, and Tamara Munzner. 2020. SepEx: Visual Analysis of Class Separation Measures. In *Proc. EuroVA*. Eurographics, Eindhoven, Netherlands, 7–11. https://doi.org/10.2312/eurova.20201079

[15] Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Michael Sedlmair, and Tamara Munzner. 2021. ProSeCo: Visual analysis of class separation measures and dataset characteristics. *Comput Graph* 96 (2021), 48–60. https://doi.org/10.1016/j.cag.2021.03.004

[16] Enrico Bertini, Andrada Tatu, and Daniel Keim. 2011. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Trans Vis Comput Graph* 17, 12 (2011), 2203–2212. https://doi.org/10.1109/TVCG.2011.229

[17] Richard Brath. 1997. Metrics for effective information visualization. In *Proc. InfoVis*. IEEE, New York, NY, 108–111. https://doi.org/10.1109/INFVIS.1997.636794

[18] Matthew Brehmer, Michael Sedlmair, Stephen Ingram, and Tamara Munzner. 2014. Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proc. BELIV*. ACM, New York, NY, 1–8. https://doi.org/10.1145/2669557.266955

[19] Phil Brodatz. 1966. *Textures: A photographic album for artists and designers*. Dover Publications, Garden City, NY.

[20] Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M Ibrahim, et al. 2019. The single-cell transcriptional landscape of mammalian organogenesis. *Nature* 566, 7745 (2019), 496–502. https://doi.org/10.1038/s41586-019-0969-x

[21] Tara Chari and Lior Pachter. 2023. The specious art of single-cell genomics. *PLOS Comput Biol* 19, 8 (2023), e1011288. https://doi.org/10.1371/journal.pcbi.1011288

[22] Chaomei Chen. 2000. Individual differences in a spatial-semantic virtual environment. *J Am Soc Inf Sci* 51, 6 (2000), 529–542. https://doi.org/10.1002/(SICI)1097-4571(2000)51:6<529::AID-ASI5>3.0.CO;2-F

[23] Pierre Comon. 1994. Independent component analysis, a new concept? *Signal Processing* 36, 3 (1994), 287–314. https://doi.org/10.1016/0165-1684(94)90029-9

[24] Tuan Nhon Dang, Anushka Anand, and Leland Wilkinson. 2012. TimeSeer: Scagnostics for high-dimensional time series. *IEEE Trans Vis Comput Graph* 19, 3 (2012), 470–483. https://doi.org/10.1109/TVCG.2012.128

[25] Tuan Nhon Dang and Leland Wilkinson. 2014. ScagExplorer: Exploring scatterplots by their scagnostics. In *Proc. PacificVis*. IEEE, New York, NY, 73–80. https://doi.org/10.1109/PacificVis.2014.42

[26] Russell Davis, Xiaoying Pu, Yiren Ding, Brian D Hall, Karen Bonilla, et al. 2022. The risks of ranking: Revisiting graphical perception to model individual differences in visualization performance. *IEEE Trans Vis Comput Graph* 30, 3 (2022), 1756–1771. https://doi.org/10.1109/TVCG.2022.3226463

[27] Kristin M. Divis, Laura E. Matzen, Michael J. Haass, and Deborah A. Cronin. 2023. Perceptual biases in scatterplot interpretation. In *Visualization Psychology*, Danielle Albers Szafir, Rita Borgo, Min Chen, Darren J. Edwards, Brian Fisher, and Lace Padilla (Eds.). Springer International Publishing, Cham, 273–291. https://doi.org/10.1007/978-3-031-34738-2_12

[28] Peter Eades. 1984. A heuristic for graph drawing. *Congr Numer* 42, 11 (1984), 149–160.

[29] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. 1983. On the shape of a set of points in the plane. *IEEE Trans Inf Theory* 29, 4 (1983), 551–559. https://doi.org/10.1109/TIT.1983.1056714

[30] Madison A Elliott, Christine Nothelfer, Cindy Xiong, and Danielle Albers Szafir. 2020. A design space of vision science methods for visualization research. *IEEE*

Trans Vis Comput Graph 27, 2 (2020), 1117–1127. https://doi.org/10.1109/TVCG.2020.3029413

[31] Geoffrey Ellis and Alan Dix. 2006. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Trans Vis Comput Graph* 12, 5 (2006), 717–724. https://doi.org/10.1109/TVCG.2006.138

[32] Ronak Etemadpour, Bettina Olk, and Lars Linsen. 2014. Eye-tracking investigation during visual analysis of projected multidimensional data with 2D scatterplots. In *Proc. IVAPP*. IEEE, New York, NY, 233–246. https://doi.org/10.5220/0004675802330246

[33] Domenico Ferrari and Lorenzo Mezzalira. 1969. *On drawing a graph with the minimum number of crossings*. Politecnico, Torino, Italy.

[34] Takanori Fujiwara, Oh-Hyun Kwon, and Kwan-Liu Ma. 2020. Supporting Analysis of Dimensionality Reduction Results with Contrastive Learning. *IEEE Trans. Vis. Comput. Graph.* 26, 1 (2020), 45–55. https://doi.org/10.1109/TVCG.2019.2934251

[35] Takanori Fujiwara and Tzu-Ping Liu. 2023. Contrastive multiple correspondence analysis (cMCA): Using contrastive learning to identify latent subgroups in political parties. *PLOS ONE* 18, 7 (2023), e0287180 (20 pages). https://doi.org/10.1371/journal.pone.0287180

[36] Takanori Fujiwara, Xinhai Wei, Jian Zhao, and Kwan-Liu Ma. 2022. Interactive dimensionality reduction for comparative analysis. *IEEE Trans Vis Comput Graph* 28, 1 (2022), 758–768. https://doi.org/10.1109/tvcg.2021.3114807

[37] Sean Gillies, Casper van der Wel, Joris Van den Bossche, Mike W. Taves, Joshua Arnott, and Brendan C. Ward. 2024. Shapely (Version 2.0.6). https://github.com/shapely/shapely. Accessed: 2024-09-12.

[38] Michael Gleicher, Michael Correll, Christine Nothelfer, and Steven Franconeri. 2013. Perception of average value in multiclass scatterplots. *IEEE Trans Vis Comput Graph* 19, 12 (2013), 2316–2325. https://doi.org/10.1109/TVCG.2013.183

[39] John C Gower and Gavin JS Ross. 1969. Minimum spanning trees and single linkage cluster analysis. *J R Stat Soc C: Appl Stat* 18, 1 (1969), 54–64. https://doi.org/10.2307/2346439

[40] Connor C. Gramazio, Karen B. Schloss, and David H. Laidlaw. 2014. The relation between visualization size, grouping, and user performance. *IEEE Trans Vis Comput Graph* 20, 12 (2014), 1953–1962. https://doi.org/10.1109/TVCG.2014.2346983

[41] Nicolas Grossmann, Jürgen Bernard, Michael Sedlmair, and Manuela Waldner. 2021. Does the layout really matter? A study on visual model accuracy estimation. In *Proc. VIS*. IEEE, New York, NY, 61–65. https://doi.org/10.1109/VIS49827.2021.9623326

[42] LeGrand H Hardy, Gertrude Rand, and M Catherine Rittler. 1945. Tests for the detection and analysis of color-blindness. I. The Ishihara test: An evaluation. *JOSA* 35, 4 (1945), 268–275. https://doi.org/10.1001/archopht.1945.00890190297005

[43] Christopher Healey and James Enns. 2011. Attention and visual memory in visualization and computer graphics. *IEEE Trans Vis Comput Graph* 18, 7 (2011), 1170–1188. https://doi.org/10.1109/TVCG.2011.127

[44] Ernst Hellinger. 1909. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *J Reine Angew Math* 1909, 136 (1909), 210–271. https://doi.org/10.1515/crll.1909.136.210

[45] Tin Kam Ho and Mitra Basu. 2002. Complexity measures of supervised classification problems. *IEEE Trans Pattern Anal Mach Intell* 24, 3 (2002), 289–300. https://doi.org/10.1109/34.990132

[46] Matt-Heun Hong, Jessica K Witt, and Danielle Albers Szafir. 2021. The weighted average illusion: Biases in perceived mean position in scatterplots. *IEEE Trans Vis Comput Graph* 28, 1 (2021), 987–997. https://doi.org/10.1109/TVCG.2021.3114783

[47] Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *J Educ Psychol* 24, 6 (1933), 417. https://doi.org/10.1037/h0071325

[48] Weidong Huang, Peter Eades, and Seok-Hee Hong. 2009. Measuring effectiveness of graph visualizations: A cognitive load perspective. *Inf Vis* 8, 3 (2009), 139–152. https://doi.org/10.1057/ivs.2009.10

[49] Aapo Hyvärinen and Erkki Oja. 2000. Independent component analysis: Algorithms and applications. *Neural Netw* 13, 4-5 (2000), 411–430. https://doi.org/10.1016/S0893-6080(00)00026-5

[50] Alan Julian Izenman. 2013. Linear Discriminant Analysis. In *Modern Multivariate Statistical Techniques*. Springer, New York, NY, 237–280. https://doi.org/10.1007/978-0-387-78189-1_8

[51] Hyeon Jeon, Ghulam Jilani Quadri, Hyunwook Lee, Paul Rosen, Danielle Albers Szafir, and Jinwook Seo. 2024. CLAMS: A cluster ambiguity measure for estimating perceptual variability in visual clustering. *IEEE Trans Vis Comput Graph* 30, 1 (2024), 770–780. https://doi.org/10.1109/TVCG.2023.3327201

[52] Ian T Jolliffe. 1986. *Principal Component Analysis and Factor Analysis*. Springer, New York, NY, 115–128. https://doi.org/10.1007/0-387-22440-8_7

[53] Younghoon Kim and Jeffrey Heer. 2018. Assessing effects of task and data distribution on the effectiveness of visual encodings. *Comput Graph Forum* 37, 3 (2018), 157–167. https://doi.org/10.1111/cgf.13409

[54] Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. 2006. Task taxonomy for graph visualization. In *Proc. BELIV*. ACM, New York, NY, 1–5. https://doi.org/10.1145/1168149.1168168

[55] Richard J Lipton, Stephen C North, and Jonathan S Sandberg. 1985. A method for drawing graphs. In *Proc. SCG*. ACM, New York, NY, 153–160. https://doi.org/10.1145/323233.323254

[56] Tingting Liu, Xiaotong Li, Chen Bao, Michael Correll, Changehe Tu, et al. 2021. Data-driven mark orientation for trend estimation in scatterplots. In *Proc. CHI*. ACM, New York, NY, USA, Article 473, 16 pages. https://doi.org/10.1145/3411764.3445751

[57] Zhengliang Liu, R Jordan Crouser, and Alvitta Ottley. 2020. Survey on individual differences in visualization. *Comput Graph Forum* 39, 3 (2020), 693–712. https://doi.org/10.1111/cgf.14033

[58] Ana C. Lorena, Luís P. F. Garcia, Jens Lehmann, Marcilio C. P. Souto, and Tin Kam Ho. 2019. How complex is your classification problem? A survey on measuring classification complexity. *ACM Comput Surv* 52, 5, Article 107 (2019), 34 pages. https://doi.org/10.1145/3347711

[59] Małgorzata Lucińska and Sławomir T Wierzchoń. 2012. Spectral clustering based on k-nearest neighbor graph. In *Proc. CISIM*. Springer, New York, NY, 254–265. https://doi.org/10.1007/978-3-642-33260-9_22

[60] Kim Marriott, Helen Purchase, Michael Wybrow, and Cagatay Goncu. 2012. Memorability of visual features in network diagrams. *IEEE Trans Vis Comput Graph* 18, 12 (2012), 2477–2485. https://doi.org/10.1109/TVCG.2012.245

[61] José Matute, Alexandru C Telea, and Lars Linsen. 2018. Skeleton-based scagnostics. *IEEE Trans Vis Comput Graph* 24, 1 (2018), 542–552. https://doi.org/10.1109/TVCG.2017.2744339

[62] Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. https://doi.org/10.48550/arXiv.1802.03426 arXiv:1802.03426

[63] Nancy Miller, Beth Hetzler, Grant Nakamura, and Paul Whitney. 1997. The need for metrics in visual information analysis. In *Proc. NPIV*. ACM, New York, NY, USA, 24–28. https://doi.org/10.1145/275519.275523

[64] Kevin R Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, et al. 2019. Visualizing structure and transitions in high-dimensional biological data. *Nat Biotechnol* 37, 12 (2019), 1482–1492. https://doi.org/10.1038/s41587-019-0336-3

[65] SM Shahed Nejhum, Jeffrey Ho, and Ming-Hsuan Yang. 2008. Visual tracking with histograms and articulating blocks. In *Proc. CVPR*. IEEE, New York, NY, 1–8. https://doi.org/10.1109/CVPR.2008.4587575

[66] Andrew Ng, Michael Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. *Adv Neural Inf Process Syst* 14 (2001), 849–856. https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf

[67] 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 *Proc. CHI*. ACM, New York, NY, 3659–3669. https://doi.org/10.1145/2858036.2858155

[68] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, et al. 2011. Scikit-learn: Machine Learning in Python. *J Mach Learn Res* 12 (2011), 2825–2830. https://doi.org/10.48550/arXiv.1201.0490

[69] Wei Peng, Matthew O Ward, and Elke A Rundensteiner. 2004. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proc. InfoVis*. IEEE, New York, NY, 89–96. https://doi.org/10.1109/INFVIS.2004.15

[70] Ghulam Jilani Quadri and Paul Rosen. 2021. Modeling the influence of visual density on cluster perception in scatterplots using topology. *IEEE Trans Vis Comput Graph* 27, 2 (2021), 1829–1839. https://doi.org/10.1109/TVCG.2020.3030365

[71] Ghulam Jilani Quadri and Paul Rosen. 2022. A survey of perception-based visualization studies by task. *IEEE Trans Vis Comput Graph* 28, 12 (2022), 5026–5048. https://doi.org/10.1109/TVCG.2021.3098240

[72] Siddhart Rajendran, John Maule, Anna Franklin, and Michael A Webster. 2021. Ensemble coding of color and luminance contrast. *Atten Percept Psychophys* 83 (2021), 911–924. https://doi.org/10.3758/s13414-020-02136-6

[73] Paulo E Rauber, Alexandre X Falcao, and Alexandru C Telea. 2018. Projections as visual aids for classification system design. *Inf Vis* 17, 4 (2018), 282–305. https://doi.org/10.1177/1473871617713337

[74] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. 2008. Incremental learning for robust visual tracking. *Int J Comput Vis* 77, 1-3 (2008), 125–141. https://doi.org/10.1007/s11263-007-0075-7

[75] Alper Sarikaya and Michael Gleicher. 2018. Scatterplots: Tasks, data, and designs. *IEEE Trans Vis Comput Graph* 24, 1 (2018), 402–412. https://doi.org/10.1109/TVCG.2017.2744184

[76] Michael Sedlmair and Michaël Aupetit. 2015. Data-driven evaluation of visual quality measures. *Comput Graph Forum* 34, 3 (2015), 201–210. https://doi.org/10.1111/cgf.12632

[77] Michael Sedlmair, Tamara Munzner, and Melanie Tory. 2013. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Trans Vis Comput Graph* 19, 12 (2013), 2634–2643. https://doi.org/10.1109/TVCG.2013.153

[78] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory. 2012. A taxonomy of visual cluster separation Factors. *Comput Graph Forum* 31, 3pt4 (2012), 1335–1344. https://doi.org/10.1111/j.1467-8659.2012.03125.x

[79] Mike Sips, Boris Neubert, John P Lewis, and Pat Hanrahan. 2009. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum* 28, 3 (2009), 831–838. https://doi.org/10.1111/j.1467-8659.2009.01467.x

[80] Stephen Smart and Danielle Albers Szafir. 2019. Measuring the separability of shape, size, and color in scatterplots. In *Proc. CHI*. ACM, New York, NY, Article 669, 14 pages. https://doi.org/10.1145/3290605.3300899

[81] Danielle Albers Szafir, Steve Haroz, Michael Gleicher, and Steven Franconeri. 2016. Four types of ensemble coding in data visualizations. *J Vis* 16, 5 (2016), 11–11. https://doi.org/10.1167/16.5.11

[82] Roberto Tamassia. 1987. On embedding a graph in the grid with the minimum number of bends. *SIAM J Comput* 16, 3 (1987), 421–444. https://doi.org/10.1137/0216030

[83] Andrada Tatu, Georgia Albuquerque, Martin Eisemann, Jorn Schneidewind, Holger Theisel, et al. 2009. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proc. VAST*. IEEE, New York, NY, 59–66. https://doi.org/10.1109/VAST.2009.5332628

[84] Andrada Tatu, Peter Bak, Enrico Bertini, Daniel Keim, and Joern Schneidewind. 2010. Visual quality metrics and human perception: An initial study on 2D projections of large multidimensional data. In *Proc. AVI*. ACM, New York, NY, 49–56. https://doi.org/10.1145/1842993.184300

[85] Valentin Todorov and Peter Filzmoser. 2010. An object-oriented framework for robust multivariate analysis. *J Stat Softw* 32 (2010), 1–47. https://doi.org/10.18637/jss.v032.i03

[86] Warren S. Torgerson. 1952. Multidimensional scaling: I. Theory and method. *Psychometrika* 17, 4 (1952), 401–419. https://doi.org/10.1007/BF02288916

[87] Chin Tseng, Ghulam Jilani Quadri, Zeyu Wang, and Danielle Albers Szafir. 2023. Measuring categorical perception in color-coded scatterplots. In *Proc. CHI*. ACM, New York, NY, USA, Article 824, 14 pages. https://doi.org/10.1145/3544548.3581416

[88] Edward R Tufte. 1985. The visual display of quantitative information. *J Healthc Qual* 7, 3 (1985), 15.

[89] John W Tukey and Paul A Tukey. 1985. Computer graphics and exploratory data analysis: An introduction. In *Proc. SIGGRAPH*, Vol. 85, 3. ACM, New York, NY, 773–785.

[90] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J Mach Learn Res* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[91] Kim J Vicente, Brian C Hayes, and Robert C Williges. 1987. Assaying and isolating individual differences in searching a hierarchical file system. *Hum Factors* 29, 3 (1987), 349–359. https://doi.org/10.1177/001872088702900308

[92] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, et al. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[93] Johan Wagemans, James H. Elder, Michael Kubovy, Stephen E. Palmer, Mary A. Peterson, et al. 2012. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization. *Psychol Bull* 138, 6 (2012), 1172–1217. https://doi.org/10.1037/a0029333

[94] Yunhai Wang, Kang Feng, Xiaowei Chu, Jian Zhang, Chi-Wing Fu, et al. 2017. A perception-driven approach to supervised dimensionality reduction for visualization. *IEEE Trans Vis Comput Graph* 24, 5 (2017), 1828–1840. https://doi.org/10.1109/TVCG.2017.2701829

[95] Yunhai Wang, Zeyu Wang, Tingting Liu, Michael Correll, Zhanglin Cheng, et al. 2019. Improving the robustness of scagnostics. *IEEE Trans Vis Comput Graph* 26, 1 (2019), 759–769. https://doi.org/10.1109/TVCG.2019.2934796

[96] Colin Ware. 2012. Information Visualization: Perception for Design, Waltham, MA.

[97] Leland Wilkinson, Anushka Anand, and Robert Grossman. 2005. Graph-theoretic scagnostics. In *Proc. InfoVis*. IEEE, New York, NY, 21–21. https://doi.org/10.1109/INFOVIS.2005.14

[98] Leland Wilkinson, Anushka Anand, and Robert Grossman. 2006. High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Trans Vis Comput Graph* 12, 6 (2006), 1363–1372. https://doi.org/10.1109/TVCG.2006.94

[99] Cindy Xiong, Cristina R Ceja, Casimir JH Ludwig, and Steven Franconeri. 2019. Biased average position estimates in line and bar graphs: Underestimation, overestimation, and perceptual pull. *IEEE Trans Vis Comput Graph* 26, 1 (2019), 301–310. https://doi.org/10.1109/TVCG.2019.2934400

[100] Cindy Xiong, Joel Shapiro, Jessica Hullman, and Steven Franconeri. 2019. Illusion of causality in visualized data. *IEEE Trans Vis Comput Graph* 26, 1 (2019), 853–862. https://doi.org/10.1109/TVCG.2019.2934399

[101] Cindy Xiong, Chase Stokes, Yea-Seul Kim, and Steven Franconeri. 2022. Seeing what you believe or believing what you see? Belief biases correlation estimation. *IEEE Trans Vis Comput Graph* 29, 1 (2022), 493–503. https://doi.org/10.1109/TVCG.2022.3209405

[102] Qing-Song Xu and Yi-Zeng Liang. 2001. Monte Carlo cross validation. *Chemom Intell Lab Syst* 56, 1 (2001), 1–11. https://doi.org/10.1016/S0169-7439(00)00122-2

[103] Fumeng Yang, James Tompkin, Lane Harrison, and David H Laidlaw. 2022. Visual cue effects on a classification accuracy estimation task in immersive scatterplots. *IEEE Trans Vis Comput Graph* 29, 12 (2022), 4858–4873. https://doi.org/10.1109/TVCG.2022.3192364

[104] Zehua Zeng and Leilani Battle. 2023. A review and collation of graphical perception knowledge for visualization recommendation. In *Proc. CHI*. ACM, New York, NY, USA, Article 820, 16 pages. https://doi.org/10.1145/3544548.3581349

[105] Caroline Ziemkiewicz and Robert Kosara. 2009. Preconceptions and individual differences in understanding visual metaphors. *Comput Graph Forum* 28, 3 (2009), 911–918. https://doi.org/10.1111/j.1467-8659.2009.01442.x