PixelatedScatter: Arbitrary-level Visual Abstraction for Large-scale Multiclass Scatterplots

Ziheng Guo, Tianxiang Wei, Zeyu Li, Lianghao Zhang, Sisi Li, and Jiawan Zhang*, *Senior Member, IEEE*
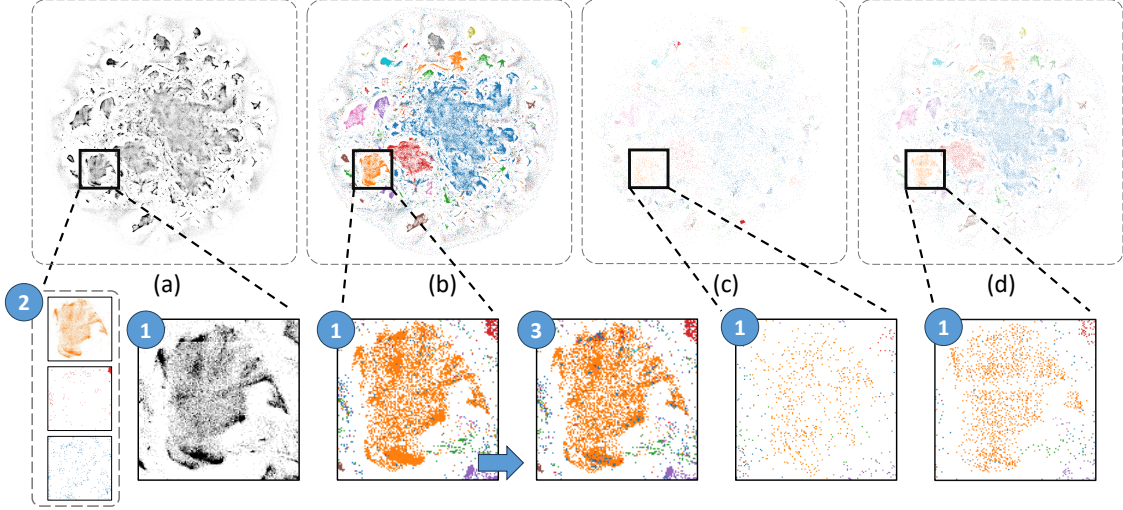
Fig. 1: The representations of different abstraction methods on a scatterplot containing 559,509 points and 10 classes. All images are rendered on an 800×800 canvas. (a) is the density greyscale map of the original scatterplot, where we manually adjusted the density-to-greyscale scale (density ranging from 0 to 20 points per pixel is linearly mapped to greyscale values from 0 to 1) to clearly distinguish low-density regions. (b), (c), and (d) represent our method, MDM (Voronoi + dot density) [23], and Recursive Subdivision Based Sampling [7], respectively. The results demonstrate that our method better preserves local details in medium-to-low density regions, such as texture, shape, and inter-class relative density (see (b), ①, compared to (c) and (d), ①). Our method also more effectively maintains density differences between regions (see global regions in (b), (c), and (d)). Furthermore, our method enhances outlier representation while preserving relative class density (see (b), ③, with reference to the class distributions in (a), ②).

Abstract—Overdraw is inevitable in large-scale scatterplots. Current scatterplot abstraction methods lose features in medium-to-low density regions. We propose a visual abstraction method designed to provide better feature preservation across arbitrary abstraction levels for large-scale scatterplots, particularly in medium-to-low density regions. The method consists of three closely interconnected steps: first, we partition the scatterplot into iso-density regions and equalize visual density; then, we allocate pixels for different classes within each region; finally, we reconstruct the data distribution based on pixels. User studies, quantitative and qualitative evaluations demonstrate that, compared to previous methods, our approach better preserves features and exhibits a special advantage when handling ultra-high dynamic range data distributions.

Index Terms—Scatterplot Abstraction, Overlap-free, Overdraw, Arbitrary Abstraction Level

✦

1 INTRODUCTION

Large-scale multiclass scatterplots often suffer from overdraw, where overlapping points in high density regions obscure the data distribution, leading to the loss of features such as data density and outliers. This severely interferes with various scatterplot tasks such as browsing and

- *Z. Guo, T. Wei, L. Zhang, S. Li, and J. Zhang(Corresponding Author) are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China. E-mail: {sygzh6, tianxiangwei, lianghaozhang}@tju.edu.cn; sisilee144144@gmail.com; jwzhang@tju.edu.cn.*
- *Z. Li is with the State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China. E-mail: lzyfun@cuc.edu.cn.*

exploring, cluster rationalization, and density judgments [39].

In recent years, researchers have proposed numerous methods to mitigate overdraw. Among these, visual abstraction methods have been widely adopted for large-scale scatterplot visualization because they clearly represent data features and offer rendering times independent of the dataset size. Such methods employ a simplified set of visual objects to convey regional features. Some of these methods are based on density estimation and use regional shapes for abstraction, such as continuous density maps [2] [15] [23] or contours [9] [30] [40]. However, these approaches can introduce visual chaos or synthesize new colors, and by obscuring the intra-region data distribution, they fail to provide a sufficiently detailed representation. Other methods rely on spatial partitioning, dividing the entire space into a series of patches, such as hexbins [27] [5] or grids [32] [19] [23], and express features on a per-patch basis. However, the data density in most large-scale multiclass scatterplots exhibits a high dynamic range (HDR), meaning there is a vast difference in density values between the densest and sparsest regions. When preserving global density differences,

existing methods severely compress the visual representation of vast medium-to-low density regions. This results in significant detail loss, manifesting as large blank or sparse regions. In Fig. 1(c), only a few regions are completely filled, while the vast majority of the remaining region, such as the region marked ① in (c), appears extremely sparse, losing the relative density (compare with ① in (a)) and blue and red outliers (compare with ② in (a)). Although Fig. 1(d) preserves the outliers, the regional density appears rather uniform, making it difficult to distinguish relatively dense regions from sparser ones.

To address above issue posed by HDR data, it is necessary to dynamically map data density to visual density, stretching the visual representation of medium-to-low density regions to make their internal details visible. However, the regional density within a patch is often inaccurate. For example, when a patch overlaps with both the dense edge of a cluster and an adjacent empty region, the average density will be a misleading 'medium' value that represents neither the dense nor the empty part accurately. Any visual enhancement based on this flawed regional density will inevitably distort the actual data density. Therefore, a prerequisite is to partition the space into what we call "iso-density regions"—spatially continuous areas of similar density—to ensure the average density is truly representative. Effectively identifying these regions in large-scale scatterplots poses three challenges. First, the regions should be of arbitrary shape to conform to the original data distribution contours. Second, their areas must be dynamically scalable. Finally, the adaptive partitioning process must be efficient for large-scale data. Grid partitioning with a fixed size fails to meet the second criterion, while previous adaptive methods often struggle with the first, typically producing regularly shaped partitions.

In this paper, we introduce PixelatedScatter, a visual abstraction method for large-scale multiclass scatterplots. Our method addresses all three challenges with a novel partitioning algorithm. It starts with a fast, grid-based partitioning (addressing the third challenge) and yields initial coarse partitions with arbitrary shapes (addressing the first challenge). Then, it adaptively iterates by reducing the grid cell size in denser areas to achieve finer partitions, thus satisfying the second challenge of scalable region sizes. Once this partitioning is established, we employ a density distribution equalization algorithm to adjust visual densities across regions. To preserve multiclass features based on the regional visual density, we then identify semantic outliers (points not belonging to the same class as their neighbors [46]) within each region and balance the preservation of relative class densities with the representation of outliers. This process determines the final pixel number for each class within each region. Finally, we place pixels as closely as possible to the original data distribution, and spread them out with minimum displacement to reconstruct the local data distribution. The final representation of pixels or pixel-like colored squares, which we term a pixelated representation, is a natural representation format for our grid-based approach. Moreover, since our partitioning process depends solely on the data distribution and not on the abstraction level (i.e., the representation level of visual units, where a higher resolution corresponds to a lower abstraction level), our method is inherently suitable for arbitrary levels of abstraction. We demonstrate the effectiveness of our method through a quantitative evaluation involving four metrics compared with related approaches, a user study, as well as two case studies. In summary, the contributions of our work are as follows:

- We propose a visual abstraction method for multiclass scatterplots that effectively preserves features across a high dynamic range of data densities, particularly suitable for large-scale datasets.

- We propose an efficient and accurate partitioning algorithm that rapidly partitions a scatterplot into arbitrarily shaped iso-density regions.

2 RELATED WORK

Researchers have proposed various strategies for visualizing scatterplots to mitigate overdraw, which can be broadly categorized into abstraction, appearance adjustment and node mapping.

2.1 Abstraction Methods

Abstraction methods typically simplify and optimize the visual representation of data via extracting features and re-encoding the abstracted features. These methods can be further divided into data space abstraction and visual space abstraction.

Data abstraction alleviates overdraw by reducing the number of points. As the reduction in the quantity of points usually leads to information loss, these methods generally do not aim to preserve all features but design sampling strategies for specific tasks to provide biased but clearer results. Typically, these methods focus on different goals such as relative densities preservation [12] [3] [6] [8], outlier preservation [31] [45] [8] and intra-region relation preservation. Methods biased on density is optimized to preserve the relative density of different regions. However, this focus on density might cause outlier loss. Conversely, outlier biased samplings excel at preserving outliers but may not accurately represent the overall density patterns. Wang et al. [7] introduced a non-uniform recursive sampling technique for multiclass scatterplots, aiming to preserve major outliers and relative class densities. This KD-tree-based partition, however, can introduce a bias by failing to preserve the spatial arrangement of points within a local region, which may distort the perceived shape of data clusters [46]. These methods inevitably introduce task-specific bias and cannot fully eliminate overdraw. It is noted that although work of Wang et al. is a sampling technique, it can be conceptually viewed as a form of visual abstraction. This is because their approach first computes colored grids for each leaf node to represent regional features, and only then selects a point of the same class from the grid as the sample.

Visual abstraction employs regional shapes to represent areal features, and usually shows clear results in preserving specific features, such as density maps [2] [15] [23] and contour lines [9] [30] [40]. Although this achieves a more explicit expression of density features and contours, outliers and sparse areas are often overlooked. Some methods incorporate outlier detection and explicitly highlight outliers [14] [33], while others adaptively segment densities to identify sparse areas [36]. In these high-level abstraction methods above, color blending and complex visual elements disrupt the perception of classes, making these methods unsuitable for multiclass scatterplots and incapable of revealing detailed structures.

In contrast to high-level abstraction, low-level abstraction methods assign a patch to each region, typically utilizing color distributions to reflect features. For example, hexbins [27] is used to preserve the regional density and class density. Multiclass Density Maps [23] render various multiclass density map representations based on 2D histograms.

2.2 Non-Abstraction Methods

In contrast to abstraction, non-abstraction methods reduce overlap by optimizing the appearance or position of points. These methods maintain the correspondence between data points and visual elements, but they cannot avoid the loss of information that accompanies the rendering of a large number of points on a canvas of limited size.

Appearance adjustment is a straightforward strategy to alleviate overdraw, such as node size reducing [44] [28], opacity decreasing [43] and shape alternative [25] [11]. Recent methods have automated or semi-automated these adjustments based on perceptual theories to enhance scatterplot visual quality. For multiclass scatterplots, many studies focus on color optimization to emphasize relationships across different classes [26] [42]. Some research proposes alternative plotting symbols to reduce overlap and enhance perceptual separation, as demonstrated by radial line symbols like the plus sign and the asterisk [41]. However, complex visual elements introduce new visual biases and significantly increase visual complexity. Since appearance properties lack the scalability to accommodate information (e.g., stacking few semi-transparent points may lead to a opaque area), these methods are typically suitable for small-scale scatterplots without significant overdraw issues.

Point mapping reduces overdraw by dispersing points while maintaining a one-to-one correspondence between data points and visual units. Ideally, they excel in feature preservation, especially local
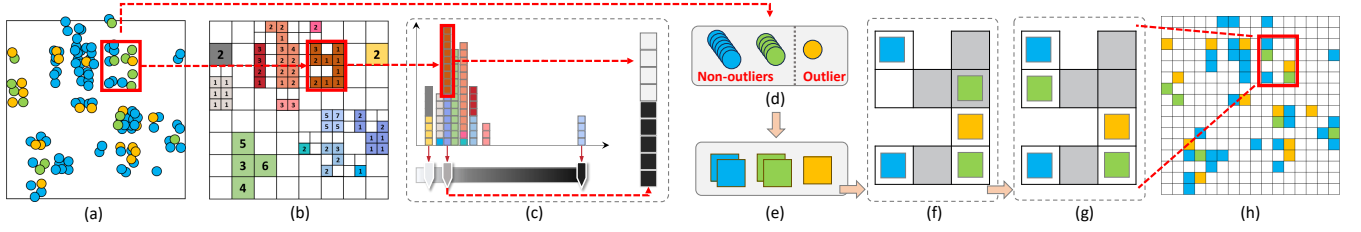
Fig. 2: Pipeline of our methods: (a) Original scatterplot; (b) Clusters after iso-density region partitioning, where each color represents a distinct cluster; (c) Splitting each cluster into density-consistent pixels to generate a data density distribution histogram, then equalizing the histogram and using the cumulative distribution function as the mapping to visual density, finally calculating the number of pixels that need coloring within each cluster; (d) Filtering outlier classes based on the class distribution within individual clusters; (e) Allocating the pixel number from (c) to obtain class pixel numbers within a cluster; (f) Constructing an initial pixel layout; (g) Dispersing pixels to produce a non-overlapping layout; (h) Final representation result.

structural features while screen resolution is sufficient to display discrete points. For instance, subspace mapping methods use equidistant grids (Dgrid [21], IsoMatch [16]), space-filling curves (HaGrid [10]), etc., to create mutually exclusive subspaces, then map data points into these subspaces to eliminate overlap. Node displacement methods [34] [20] [17] [35] [29] spread points of a certain size from their original positions, generally aiming to minimize dispersing area. However, there is an inherent conflict between node visibility (generally ensured by point radius) and displacement distortion, making point-to-point mapping methods prone to pixel overdraw.

The Point-to-Pixel Mapping approach [38] shifts the visual unit from points to pixels, aiming to better utilize screen space. For example, Gridfit [24] relocates unallocated data points along predefined curves to the nearest unoccupied pixels to fill space. This method achieves a balance between basic visibility requirements (avoiding pixel overdraw) and reducing distortion. However, displacement inevitably leads to the loss of local structures, and extensive filling can severely impair density and shape features.

3 METHODS

Our goal is to design a visual abstraction that enhances the visual representation of medium-to-low density regions while preserving the main features of multiclass scatterplots [7] [39] (relative data density, relative class density and outliers). We address this problem using three coherent steps: first, adaptively partitioning the scatterplot and nonlinearly mapping regional data density to visual density to enhance the visual density of medium-to-low density regions under the constraint of preserving relative data density; then, identifying outliers within regions to guide the preservation of relative class density and outlier class; finally, simulating the original data distribution by pixels. Fig. 2 illustrates the pipeline of our work.

3.1 Iso-density Region-based Density Equalization

Iso-density regions are a series of regions with internally similar density distributions that reflect different data patterns, as changes in underlying data patterns often lead to abrupt density shifts. We utilize this partitioning for the precise estimation of regional data density (see Fig. 2(b)). Then, in order to adapt to arbitrary data distributions, we propose to nonlinearly and monotonically map data density to equalized visual density, which is called regional density distribution equalization(see Fig. 2(d)). This is inspired by histogram equalization [18], as it can monotonically expand the distribution range of data, aligning with our goal. The fundamental basis of the above process is that an iso-density region almost has no multiple data patterns; as long as the mapping is monotonic, it preserves the relative density order between regions.

Iso-density Region Partition. The partition has three key characteristics: similar data densities within each region, continuous and irregular shapes of regions, and the ultra-fine granularity of the partitioning, which requires high time efficiency. To meet these characteristics, we employ a grid-based approach combined with a simple clustering strategy to rapidly produce a coarse partitioning result, followed by iterative refinement through sub-partitioning, where sub-partitioning
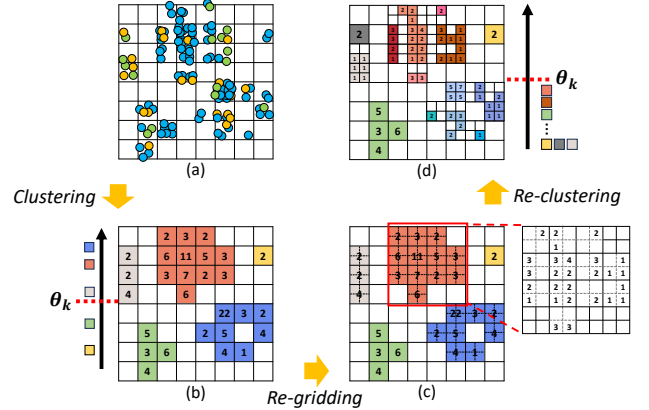

Fig. 3: Illustration of Iso-density Region Partition: (a) Gridding of the original data; (b) Clusters after initial clustering, with the kurtosis calculated for each cluster; (c) Re-gridding clusters with kurtosis larger than θ_k using a halved grid size; (d) The Final cluster partition result after re-clustering, where the kurtosis of each cluster is less than θ_k.

condition depends on the kurtosis of regional density distribution. The method is separated into three parts: gridding, clustering and iterative refinement.

Gridding partitions a scatterplot into square grids with a grid size. Due to the method involving multiple partitioning steps, the grid size varies with the granularity of partitioning. However, for any given scatterplot, we always begin with an initial gridding using a grid size of 1 pixel. This ensures that the result can always be precisely represented by canvas pixels, and is more conducive to understanding. In fact, the initial grid size represents the abstraction scale. In practice, we recommend adjusting the abstraction scale through canvas size; for example, setting the grid size to 2 pixels is equivalent to halving the canvas size, which is both safer and more flexible.

Clustering on grids involves two steps: firstly we count all grids containing points, referred to as non-empty grids, where each represents a cluster. Then, we traverse these non-empty grids, merging clusters within a 3×3 grid area centered on each non-empty grid. This guarantees each node in a cluster has another node within a distance of $2\sqrt{2} \times GS$, where GS represents the grid size.

Iterative refinement connects the results of gridding and clustering. The key idea is to sub-partition clusters adaptively through kurtosis. For every cluster, we count point numbers of all grids to get density distribution. Although the distribution is two-dimensional, we are not concerned with locations of valleys and peaks but rather the flatness of the distribution. So we directly calculate the one-dimensional kurtosis of distribution to estimate whether this cluster is uniform. If the kurtosis exceeds a fixed threshold θ_k, we half the grid size, then grid and cluster again, shown in (Algorithm 1). In our algorithm, we use a partitioning level L (relative to the initial gridding) to uniformly adjust multiple clusters to a specified level. Typically, L increases sequentially,

representing iterative refinement layer by layer. However, L can also decrease, indicating a re-gridding and clustering with a larger grid size. This iterative refinement process continues until every cluster kurtosis falls below θ_k, as shown in Fig. 3(d).

The whole partitioning is as follows: for a given scatterplot, we first perform initial gridding. Subsequently, we use an initial partitioning level L_{init} (default set to -1) to aggregate grids, which help preserve outliers in sparse region. Then, we apply iterative refinement to obtain the final iso-density region clusters.

Algorithm 1: Iso-density Region Partition

Input :
    $DS : \{(x, y)\}_N$ // Original 2D data
    θ_k // Kurtosis threshold
    L_{init} // Initial partition level
Output :
    $clusters_g : \{\{(x, y)\}_{n_i}\}_M, \sum n_i = N$ // Equal-density clusters

1 **function** Partition(DS, θ_k, L_{init})
2     $clusters_p \leftarrow \{DS\}$ // Clusters peaked need partition
3     $clusters_g \leftarrow \emptyset$ // Clusters gentle that is iso-density
4     $L \leftarrow L_{init}$ // Partition level
5   **while** $clusters_p$ not *Null* **do**
6         $clusters' \leftarrow$ SinglePartition($clusters_p, L$)
7         $clusters_p \leftarrow$ FilterKurtosis($clusters', \theta_k$)
8         $clusters_g \leftarrow clusters_g \cup (clusters' - clusters_p)$
9   L++
10   **end while**
11   **return** $clusters_g$
12 **end function**

13 **function** SinglePartition($clusters, L$)
14     $clusters' \leftarrow \emptyset$
15   **for** *cluster* in *clusters* **do**
16         $D \leftarrow$ Gridding($cluster, \frac{1}{2^L}$)   // Re-gridding with half grid size
17         $clusters' \leftarrow clusters' \cup$ Clustering(D)
18   **end for**
19   **return** $clusters'$
20 **end function**

Regional Density Distribution Equalization. First, for each iso-density region (hereafter "cluster"), we calculate its data density by dividing the number of points it contains by its area in pixels. Since these cluster regions can overlap, a single pixel location on the canvas might be associated with multiple clusters. We resolve such conflicts by assigning the pixel to the cluster with the smallest area-to-class-number ratio, a heuristic that prioritizes the preservation of more complex, multiclass features.

With the data density for each region established, we then construct a data density histogram. This histogram is built by considering the set of all regions; specifically, for each region, its area (in pixels) is added as a count to the histogram bin corresponding to that region's data density value. Therefore, the total population for the equalization is the sum of the areas of all regions. Next, we compute the Cumulative Distribution Function (CDF) from this histogram. The core of our mapping is this: the new visual density for a region is defined as its corresponding value in the CDF. Because the CDF is normalized between 0 and 1, this effectively assigns visual densities "as probabilities," stretching the density range to enhance contrast, as depicted by the greyscale mapping bar. Finally, the number of points to be rendered in each region is recalculated based on this new, equalized visual density. The resulting layout and its more uniform visual density histogram are shown in Fig. 4(b) and at the bottom of Fig. 4(c), respectively.

3.2 Outlier-guided Class Pixel Allocation

After partitioning a scatterplot into iso-density region clusters, a single cluster may contain points of different classes. If we allocate pixels
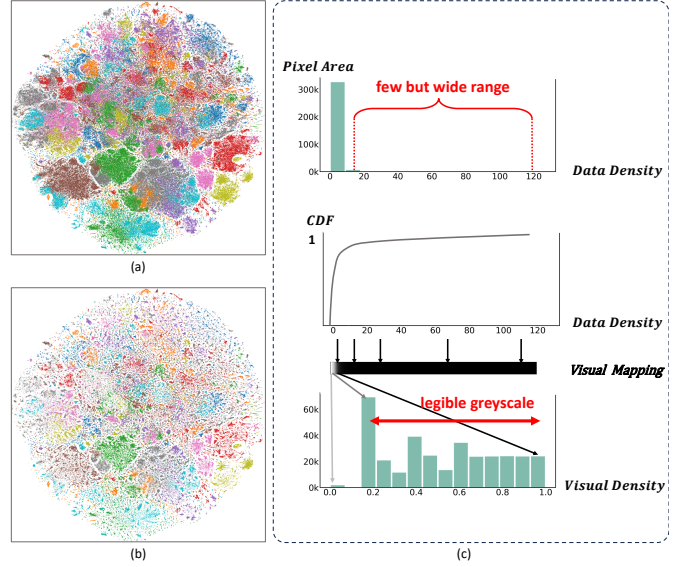

Fig. 4: Regional Density Distribution Equalization: (a) the original scatterplot; (b) effect after density equalization; (c) density equalization mapping process, where the top section shows the original data distribution, the middle section displays the cumulative distribution function after equalization, and the bottom section presents the distribution of visual density after mapping.

faithfully according to class proportions, outlier classes may not be preserved. Conversely, if we excessively enhance the representation of outlier classes, the relative class density may be compromised. Thus we defined the characteristic differences between outliers and non-outliers and proposed an outlier class filtering model. Based on this, two stages of formalized expression for class outlier emphasis are introduced to address the conflict between the outlier representation and relative class density preservation.

Outlier Classes Filtering. To accurately distinguish between outlier and non-outlier classes, we focus on two aspects: First, ensuring that the point number of each outlier class does not exceed a threshold $\tau_o \times N_{avg}$, where N_{avg} represents the average point number of classes. The introduction of N_{avg} is because the proportions of classes are related to the class number; the proportion of each class naturally decreases as the number increases. Second, the total proportion of non-outlier classes must be no less than a threshold τ_{ns}. These two thresholds ensure the minority of outliers while maintaining differences among non-outliers. These two thresholds are related, specifically:

For a n-classes cluster containing N_c points, the average point number of classes N_{avg} is $\frac{N_c}{n}$. The total number of outlier classes must not exceed $(n-1)\tau_o \frac{N_c}{n}$, reflecting an extreme case where the cluster contains only one non-outlier class. Consequently, the total number of non-outlier classes is no less than $(1 - \frac{n-1}{n}\tau_o)N_c$, which simplifies to the total proportion of non-outlier classes no less than $1 - \frac{n-1}{n}\tau_o$. This indicates that for a given total proportion of non-outlier classes threshold τ_{ns}, we can always compute a τ_o for any cluster with different classes numbers, ensuring the τ_{ns} constraint is satisfied:

$$\tau_o \leq (1 - \tau_{ns})\frac{n}{n-1}, \qquad (1)$$

where $0.5 \leq \tau_{ns} \leq 1$, because the area of non-outliers should not be less than that of outliers. In our method, we set the parameter τ_{ns} to 0.5 by default.

Outlier Emphasis Patterns. Through the aforementioned filtering, we distinguish between outlier and non-outlier classes in any cluster. Then we propose two stages of outlier emphasis to guide the allocation of pixels quantities:
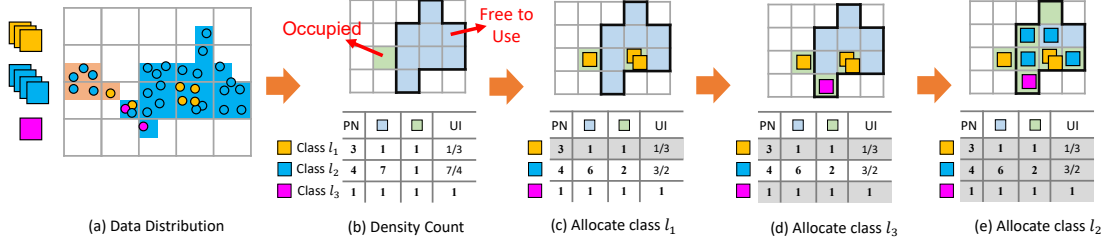
Fig. 5: Pipeline of Initial Layout Construction: (a) The initial state; (b) Identifying placeable (light blue) and unplaceable (light green) regions; (c, d, e) Iteratively placing the most urgent class after calculating urgent indexes (UI) to complete the initial layout.
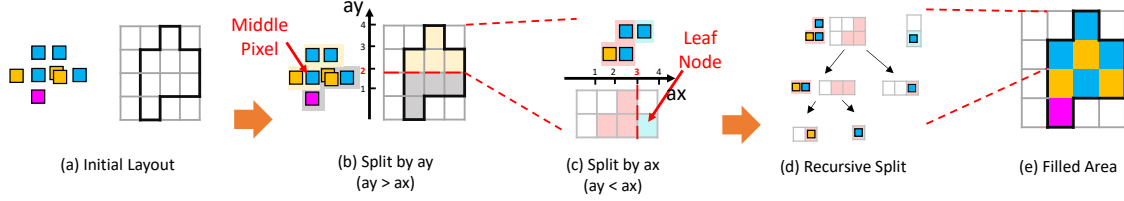

Fig. 6: Pipeline of KD-Tree Guided Pixel Dispersion: (a) The initial overlapping layout; (b) Splitting the region along its longer axis (the y-axis); (c, d) depict the sub-split within the yellow area, and such splits continue until each pixel is assigned to just one grid; (e) The non-overlapping pixel layout.

We assume that class i occupies p_i of point numbers in the region, with the total area being m pixels, so the ideal area with complete density preservation can be calculated as $A_i = p_i m$. The number of outlier class i is referred as ON_i.

ST1: We ensure the presence of pixels for semantic outliers via mandating the minimum pixel number more than one,

$$ON_i = \begin{cases} 1 & if \ A_i < 1 \\ [A_i] & else \end{cases} \qquad (2)$$

ST2: We ensure the non-inversion between outliers and non-outliers via mandating the area occupied by the most dense outliers not to exceed that of the least dense non-outliers, where h adjusts the degree of outlier emphasis.

$$ON_i = \begin{cases} 1 & if \ hA_i < 1 \\ [hA_i] & else \end{cases} \qquad (1 < h \leq h_{max}) \qquad (3)$$

The density reversal threshold h_{max} can be given by the following equation, where op and np are referred as outliers proportion and non-outliers proportion; j, k are class numbers of outliers and non-outliers:

$$\max op h_{max} = \frac{\min np}{\sum_1^k np_i}(1 - h_{max} \sum_1^j op_i) \qquad (4)$$

When $h = 1$, the outlier pixel numbers corresponds to that in **ST1**, indicating the retaining of outliers without emphasis. When $h = h_{max}$, the area occupied by the most dense outlier class equals that by the least dense non-outlier, reaching the threshold that maintains the non-reversal of density between outliers and non-outliers.

In ST1, nearly uniform density proportion preservation was applied to all classes. In ST2, density order preservation among classes was maintained, while relative proportion was preserved separately in outliers and non-outliers. Fig. 13 shows the results for h values of 2 and 10.

3.3 Pixel-based Data Distribution Reconstruction

Our goal is to reconstruct the data distribution by arranging multiclass pixels within cluster regions according to the original data distribution. To achieve this goal, we design a two-stage pixel layout method: First, we build an initial pixel layout based on the original data distribution. Subsequently, we propose an overlap-free layout method based on first stage. We use a median-split kd-tree to recursively partition the space and disperse overlapping pixels from the initial layout. Fig. 5 and 6 illustrate the pipeline of these two steps respectively.

Initial Layout Construction. We first construct an initial overlapping layout, which is strategically designed to minimize displacement costs in stage two by reducing initial overlaps. The core is a heuristic that prioritizes classes with fewer available grids for placement. For instance, a class with many potential locations can be deferred, as it will likely still have sufficient space after other classes are placed, like the blue class in Fig. 5(a). We formalize this by calculating an 'urgent index' (UI) for each class (placeable grids / class size). Pixels are then allocated to all placeable grids. If there are still unallocated pixels, they are placed cyclically according to density from highest to lowest. Fig. 5 illustrates the above process.

KD-Tree Guided Pixel Dispersion. After obtaining the initial pixel layout, we recursively disperse each pixel to a unique grid using a median-split kd-tree [22, 38], as shown in Fig. 6. A kd-tree is a space-partitioning data structure that organizes points in a multi-dimensional space; for our 2D pixel layout, it builds a binary tree by recursively splitting the set of pixels along alternating axes (x and y). Each split includes two key steps: deciding whether to split along the x-axis or y-axis and determining the position of split line. The first step based on the bounding rectangle of the region to be split. If the rectangle's (x-axis) width is greater than the (y-axis) height, then split along the x-axis; otherwise, split along the y-axis. This step helps maintain the balance of the binary tree. For the second step, we take the median x-coordinate (or y-coordinate) of the pixels as the split line.

4 EVALUATION

4.1 Quantitative Evaluation

This section presents the quantitative comparison involving four metrics at different levels of abstraction, qualitative comparison, and a user study conducted to determine whether the results are perceptible to users. Experiments and rendering were performed on an environment with i9-13900HX 2.2GHz processor, RTX 4060 Laptop GPU, and 48GB RAM. All code was implemented in JavaScript and uesd WebGL for fast browser-side rendering.

Datasets For evaluating our method, we utilized 20 real-world multiclass datasets from the UCI data repository [1] and Nomic Atlas data [13], with the number of data points ranging from 13K to 3.6M and the number of classes from 3 to 128. Among them, 11 datasets contain over 400K points, and 4 are at the million scale.

Comparison Methods We implemented pixel versions of all the following abstract layout methods: Recursive Subdivision based Sampling(bbreviated as RS* Sampling) [7], Multiclass Density Maps [23] and Color Weaving [32]. For broader evaluation, the state-of-the-art point mapping method DGrid [22] was also included, although such
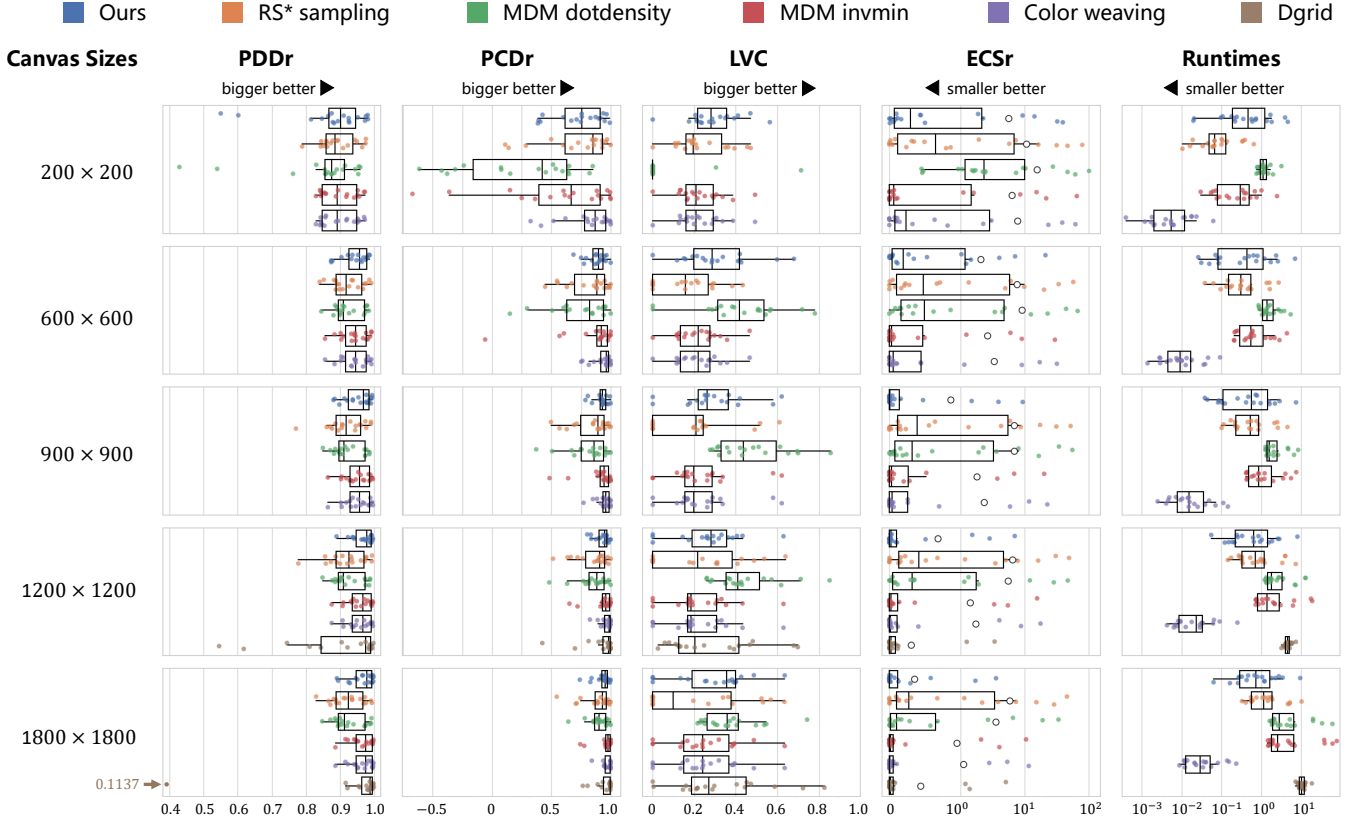
Fig. 7: Box plots were used to visualize the results for the four metrics and layout times across 5 canvas sizes. For PDDr, PCDr, and LVC, higher values indicate better performance; for ECSr, lower values are preferred.

methods are not suitable for arbitrary canvas sizes. Generally speaking, these methods are grid-based; however we set the grid size to a single pixel. The RS* Sampling parameters are set to $\lambda = 0.02$ and $\tau = 0.02$, following the default settings in the paper. MDM has two sets of parameters: the first uses *Voronoi* and *Dotdensity* to preserve regional density (with $rebin_{size}$ set to 2000 and $rescale$ set to $sqrt$ to enhance the representation of low-density areas); the second uses *Invmin* to highlight outliers. Color Weaving employs random weaving to preserve class density. For DGrid, the gaussian kernel size M was fixed at 31, as a practical trade-off between performance and acceptable runtime. Our method uses fixed parameters threshold $\theta_k = 10$ and visual emphasis ratio $h = 10$, while the initial level L_{init} was automatically determined as described in the parameter analysis.

Although Pixel-Relaxed Scatter Plots [38] is also a pixel-based technique, it was primarily designed to optimize the use of the screen space rather than preserve the data distributions of scatterplots. Specifically, the method generates a solidly tiled region that completely disrupts the original distribution's shape; and its color encoding is designed for single class. These issues render the method unsuitable as a competing approach. Concrete results are provided in supplementary material.

Metrics Since our method focuses on preserving density information and outliers, the metrics Perceived Data Densities Ratio (PDDr), Perceived Class Densities Ratio (PCDr), and Erased Class Sample ratio (ECSr), summarized by Wang et al. [7], align with our objectives and can help evaluation. Additionally, considering human perception of density is nonlinear, especially in large-scale scatterplots, there are many regions with similar data densities, or extremely high (or low) densities, where the human eye cannot clearly discern the visual density of these regions. Thus, greater visual contrast aids in aligning perception with data density. We introduce a new metric, Legible Visual Contrast (LVC), to measure the clarity of the layout's visual representation. We provide formal definitions as follows:

Given a pixelated map as $V : \mathbb{R}^2 \rightarrow \mathbb{R}$, indicating that each grid can be covered by a single label or be empty. A region Ω_i's visual density

$D_v(\Omega_i)$ is defined as the ratio of covered grids and all grid count.

Legible Visual Contrast (**LVC**) measures the overall visual clarity of a layout, quantified as the average perceivable contrast. Perceivability implies that if the visual density of two regions both does not exceed the visibility visual density threshold θ_{grey}, then their contribution to the contrast is disregarded. That's because sparse distributed pixel areas are challenging to observe with clear edges, making it visually difficult to distinguish such areas from other similarly sparse regions. Drawing upon well-established principles for calculating contrast in image processing [37], we define the local contrast C_{ij} between two regions Ω_i, Ω_j as the normalized difference in visual density:

$$C_{ij} = \begin{cases} 0 & if \ \max(D_v(\Omega_i), D_v(\Omega_j)) \leq \theta_{grey} \\ \frac{|D_v(\Omega_i) - \overline{D}_v(\Omega_i, \Omega_j)|}{\overline{D}_v(\Omega_i, \Omega_j)} & else \end{cases} \quad (5)$$

where $\overline{D}_v(\Omega_i, \Omega_j) = \frac{D_v(\Omega_i) + D_v(\Omega_j)}{2}$ is the average visual density of the two regions. Given this, LVC is then calculated as the mean of local contrasts across all neighboring region pairs:

$$LVC = \frac{\sum_i \sum_{j \in neighbors} C_{ij}}{RN} \quad (6)$$

where RN is the total number of region pairs that meet θ_{grey} condition. The range of LVC is [0,1], where values closer to 1 indicate better overall visual clarity of the layout.

Results All abstraction methods were calculated on canvas sizes of 200, 600, 900, 1200, and 1800. Because the canvas size required by DGrid depends on point numbers, it was evaluated only at 1200 (excluding dataset *Tweets from MUSC*) and 1800 (all data). To ensure fair comparison across canvases, the sliding-window size was scaled proportionally with the canvas (to 20, 60, 90, 120, and 180 pixels, respectively), so the analysis window always covered the same relative spatial region. We fix LVC window size at 10 and θ_{grey} at 0.3, because of the constancy of the human perceptual. Evaluation results are shown
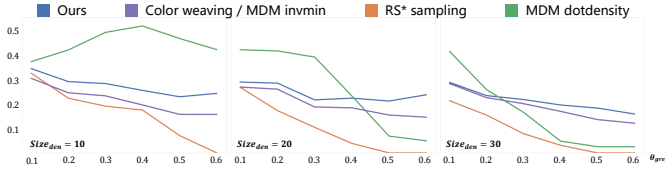
Fig. 8: The results of average LVCs on the 900×900 canvas size. Our method achieves better performance and stability when parameters change.

in Fig. 7; for ECSr, where the medians of different methods are close, we additionally annotate the mean with black-edged circles. At almost every canvas size our method consistently leads in both PDDr and ECSr. For PDDr it always attains the highest median with smaller extrema, demonstrating the stability of our region-density preservation strategy. For ECSr most methods score well because many datasets have few classes, masking differences. However, our method achieves the smallest minima and means at every canvas size, even outperforming the outlier-oriented MDM Invmin, proving its comprehensive ability to retain outliers. In PCDr our method ranks second, close to the best. Note that due to our explicit calculation of density and class preservation, there are minor rounding errors in our method, which may cause us to perform slightly worse on sparse, small-scale datasets. However, feature preservation in such datasets is relatively straightforward, and all methods score comparably high. These errors are negligible on large datasets, where our method achieves higher PCDr.

Fig. 8 shows the LVC trend on a 900×900 canvas under different window sizes $Size_{den}$ and θ_{grey}. High sensitivity to $Size_{den}$ suggests that contrast is scale-dependent, while a strong dependence on θ_{grey} indicates that the metric is artificially inflated because most regions have very low grey levels. Our method is the most stable and well-performed, indicating the reliable visual density enhancement. RS* Sampling lacks pronounced density differences and maintains similarly low density globally. MDM Dotdensity produces many nearly blank regions and a few dense ones, yielding strong contrast only between them. The contrasts achieved by MDM Invmin and Color Weaving are highly dependent to datasets and largely unrelated to canvas size.

Our method is also the most adaptive to canvas size: as the canvas shrinks its performance decays slowly and stably across all metrics. DGrid requires many empty grids to convey regional information, degrading severely on smaller canvases. Among density-based methods, RS* Sampling performs relatively better on small canvases because pixel-based density estimation becomes more uniform. However its metrics improve slowly with larger canvases, with ECSr and PDDr plateauing beyond 600. MDM Dotdensity behaves similarly; additionally, its large blank regions yield poor ECSr and PCDr. Other simple strategies (MDM Invmin and Color Weaving) deteriorate rapidly on small canvases: MDM Invmin over-emphasises outliers on 200 canvas size, sharply reducing PCDr, while Color Weaving's random sampling causes a marked drop in ECSr at the same size.

In terms of runtime, although not the fastest, our method processes most datasets within one second and returns results within ten seconds even for million-point, hundred-class datasets. Interestingly, whereas other methods slow down markedly with increasing canvas, ours shows little correlation with canvas size: at 1800 it is already the second fastest. This aligns with our design, which depends on data distributions rather than manual abstraction levels.

Time Complexity Considering the time complexity of iso-density region partition: assuming the total number of points in the dataset is N and the initial number of grids is G, the overall complexity is $dO(N) + dO(G)$, where d is the number of partitions. Since d has an upper bound, making the final time complexity $O(N) + O(G)$. Considering the time complexity of the subsequent layout process: the outlier separation of all clusters has a time complexity of $O(G)$. Pixel allocation consists of two steps: the initial pixel assignment with a time complexity of $O(G)$, and spatial partitioning and remapping using a kd-tree with a time complexity of $O(G \log G)$. Consequently, the overall time complexity of our method is $O(G \log G) + O(N)$.
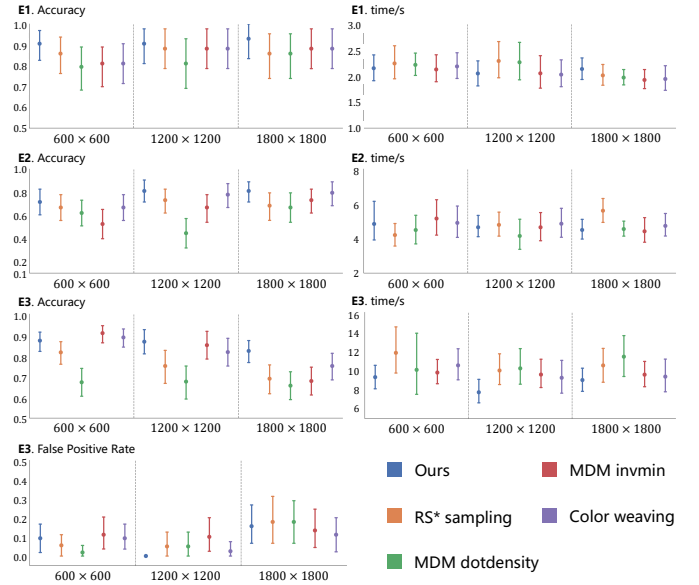

Fig. 9: Average accurary and complete time of **E1**.relative regional density perception and **E2**.relative class density perception. Average accurary, complete time and False Positive Rate(FPR) of **E3**.semantic outlier perception.

4.2 User Study

Although quantitative metrics provide insights, how well PDDr, PCDr, and ECSr reflect pixel-level human perception remains unclear. We therefore designed a user study to directly verify the perception of density and outlier information.

Experimental design. The study comprised three tasks: **E1** relative regional density perception, **E2** relative class density perception, and **E3** semantic outlier perception. For **E1** and **E2** we adopted the protocol of Yuan et al [46]. Specifically, **E1** asked which contained more points for two randomly selected non-empty regions; **E2** asked which class was more numerous within a given region. For **E3** we redesigned the task: for each scatterplot two classes were randomly selected; for each class we sampled five regions, each time choosing with equal probability either a random region containing the category or one without it. Participants need to identify which of five regions contained the target class. Effectiveness was measured by accuracy and false-positive rate. Because large scatterplots contain fine structures, the region size was set to *canvas size*/10, consistent with the quantitative evaluation, and experiments were conducted at canvas sizes of 600, 1200, and 1800. To maintain perceptual continuity within a given canvas size, we partitioned the three experiments into nine sets by canvas size and randomized question orders within each set to mitigate learning effects. Further details appear in the supplementary material.

Datasets. We selected seven of the previous twenty datasets: *Forest covertype*, *Hathi trust library*, *Dc census citizens*, *Cup98LRN*, *MoCap*, *CS rankings*, and *Daily sports*. Each of them contains 100 thousands of points, ensuring discernible differences among regions at the region size chosen before. These datasets vary widely: *Cup98LRN* and *CS rankings* exhibit complex inter-class relationships, whereas *Hathi trust library* has regions with diverse densities.

Participants. Twenty undergraduate or graduate students were recruited, eight with research experience in visualization or computer graphics. No participant reported color blindness or color weakness.

Apparatus. The study was conducted offline via a web interface on monitors with a resolution of 1920×1080. Before each session, a 20-minute briefing ensured that participants fully understood the background, procedure, and precautions.

Results. Fig. 9 reports accuracy and completion time for all tasks (**E3** additionally shows FPR). For relative regional density perception(**E1**), our method leads at every canvas size, attaining almost 0.9 accuracy; MDM Invmin and Color Weaving perform notice-
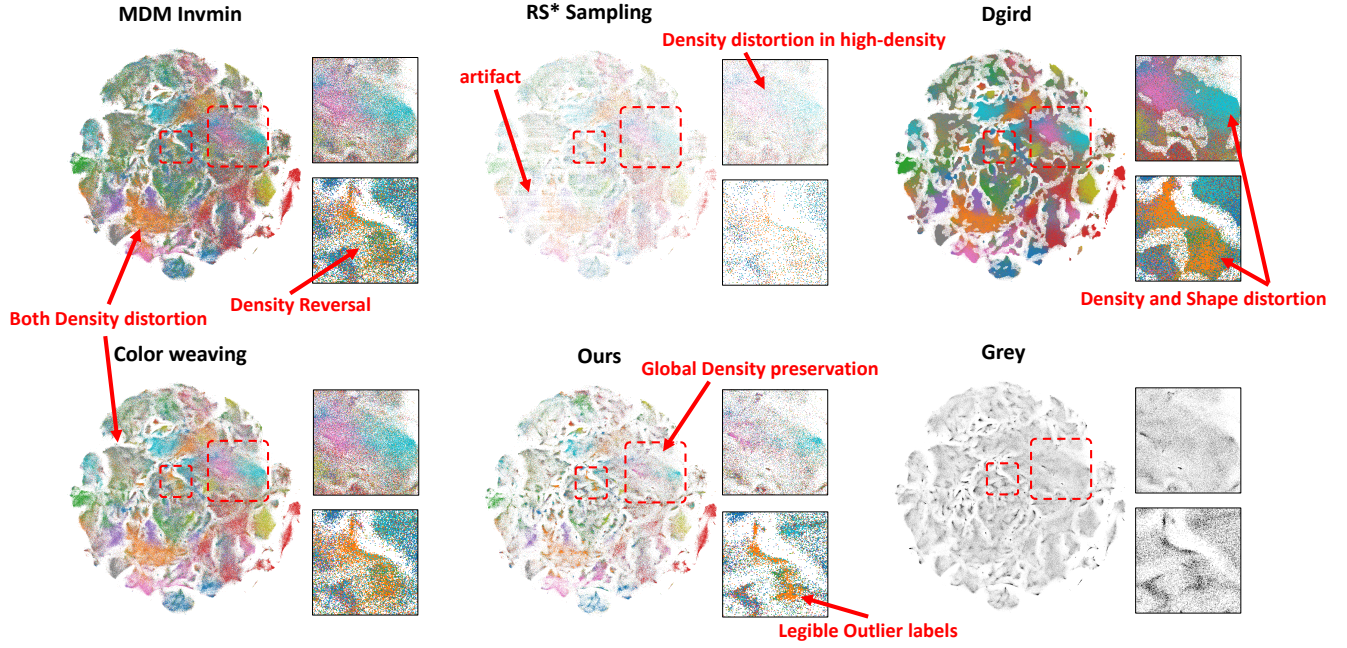
Fig. 10: Detailed comparison results on *DBLP samples*. The first five images show results from different methods, followed by a grayscale density map. This map linearly converts density values in the range [0, 7] to grayscale [0, 1], as this range captures over 99.5% of the data points.

ably worse on the small canvas. Friedman tests indicate statistically significant differences in accuracy among these methods ($p = 0.0362, 0.0625, 0.1070$ for canvas sizes of 600×600, 1200×1200, and 1800×1800 respectively; all subsequent p-values follow this order). For relative class density perception(**E2**), our method and Color Weaving share the top rank, whereas the outlier-oriented strategy of MDM Invmin more readily inverts class density on small canvases ($p = 0.0252, 0.0153, 0.0112$). Recognition times in **E1** and **E2** are short and similar across methods (for **E1**, $p = 0.6268, 0.7603, 0.2674$; for **E2**, $p = 0.4830, 0.6151, 0.3386$).

For semantic outlier perception(**E3**), on the 600 canvas size MDM Invmin and Color Weaving facilitate outlier perception due to more colored pixels; our method performs closely. As canvas size grows, outlier perception declines for all methods because the absolute number of colored pixels per region increases, making outliers harder to spot. Because of the outlier-enhancement strategy, our method amplifies outlier visibility, significantly outperforming others and declining more slowly in terms of accuracy ($p = 0.0117, 0.0245, 0.0382$). The strategy also yields easily recognizable outlier class clusters, giving our method the shortest completion times on all canvases ($p = 0.0910, 0.1290, 0.0306$). FPR appears correlated with the confusion of class distributions: our method maintains a comparatively low FPR across all canvas sizes ($p = 0.1875, 0.1598, 0.2081$).

4.3 Qualitative Evaluation

To further explore the effectiveness of methods in terms of density and outlier preservation, we conducted a qualitative comparison on the large-scale multiclass dataset *DBLP samples*, which contains 820,000 points across 34 classes, shown in Fig. 10. All methods were rendered on a 1200×1200 canvas. To facilitate the comparison of global density preservation, we provide the greyscale density map.

Both MDM Invmin and Color Weaving exhibited obvious density distortion because all pixels with points were rendered. This caused the visual density to be related to the area occupied by points rather than the data density. While MDM Invmin strategy can highlight outliers in most cases, it leads to an imbalance in inter-class density and suffers from density reversal, where low-density but uniformly distributed classes occupy more pixels than the density. In Dgrid, displacement disrupts local structures, with noticeable distortion particularly in high density areas, and large-scale scatterplots require additional canvas
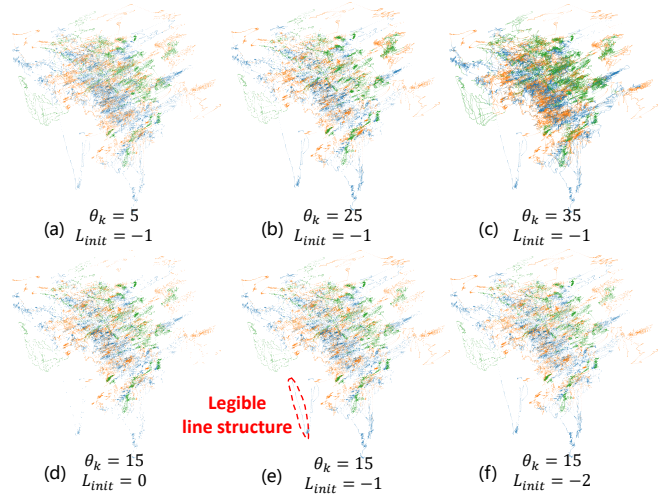

Fig. 11: Parameters analyze on *HT sensor*.

space. Our method also allows slight pixel displacement to maintain inter-class density, but our displacement is controlled and does not exceed the original region. Additionally, our nonlinear mapping of visual density enhances contrast for better density perception, and our outlier highlighting strategy safely accentuates outlier representation.

4.4 Parameter Analysis

The parameter θ_k influences the granularity of cluster partitioning, as shown in Fig. 11(a-c, e). For larger θ_k, the cluster partition is more lenient, even if there are some peaks within the region. When the parameter is smaller, we obtain more fine-grained clusters. Thus, smaller θ_k generally leads to higher contrast. However, too small θ_k can result in overly fine cluster partitions, causing loss of structural information and increased computational time. Typically, we set θ_k from 10 to 20, which maintains a good balance between density expression and computation time for most datasets. For sparser datasets or larger canvases, the value should be adjusted downward accordingly.

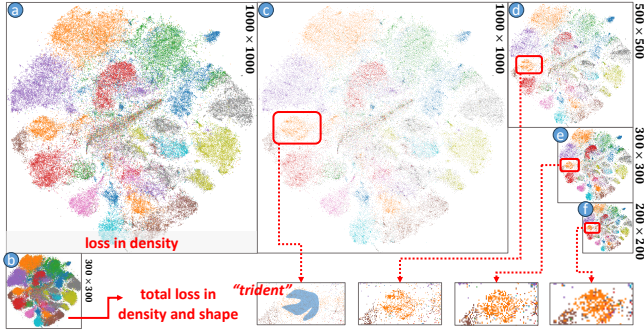The parameter L_{init} helps retain structural information in sparse

Fig. 12: High-definition display at any resolution on *CSranking*.

Fig. 13: Enhanced perception of semantic outliers on *Forest covertype*.

regions, as shown in Fig. 11(d-f). When the initial partitioning level is higher (i.e., L_{init} is set smaller), even low-density regions with dispersed points can be grouped into one cluster rather than a series of isolated clusters, helping in structural preservation for sparse regions. For example, Fig. 11(e) retains discrete line structures. However, too small L_{init} would reduce number of clusters in medium-density regions, which could cause lower contrast. We automate this process based on an empirical baseline of $L_{init} = -1$ for a 1000×1000 canvas. This value is then decreased by 1 for each doubling of the canvas size and increased by 1 for each halving, with the final result rounded to the nearest integer.

5 APPLICATION

5.1 High-Definition Display at Any Resolution

In scenarios such as printing or covers, there is always a need to provide results at a fixed resolution. Sometimes the resolution may even be extremely low, when in scenarios with network congestion or when thumbnails are required. Our method can display as much information as possible at any resolution. Fig. 12(c-f) shows the results of our method on the *CS ranking* dataset, with the resolution on the right. For comparison, (a)(b) show the effects of the original scatterplot layout. The most distinct feature of this dataset includes several clusters representing specific disciplines: peripheral clusters are more separated, whereas those near the center are smaller and less distinct. At a normal resolution, the original plot loses density information within the clusters, shown in (a). Our method preserves the density within the clusters, shown in (c). When the resolution reduces as in (b), the original boundaries of the clusters overlap and become indistinguishable. Additionally, colors blend into unrecognizable new colors. Yet, our method maintains the visibility of clusters at the same or lower resolution, and even the internal structure of the clusters is preserved. The orange cluster on the left in (c-f) demonstrates this capability, where we preserve its "*trident*" shaped structure. Class density and outliers are also maintained, with no color blending.

5.2 Enhanced Perception of Semantic Outliers

Forest covertype dataset [4] contains over 581,000 instances. The scatterplot is generated with the x-axis representing *Elevation* and the y-axis representing *Distance to Hydrology*, with forest cover type as the point class, as shown in Fig. 13(a). The distribution of forest cover seems to be significantly related to elevation. However, the overlap of points makes it difficult to identify sparse cover types, while the loss of density complicates the perception of coverage rates.

Through our method, the scatterplot is re-laid out as shown in (b). First, we obtain a clear perception of density: krummholz is primarily distributed at 3400m high, and forest cover is very sparse at this elevation. As the distance to hydrology increases, the forest rapidly diminishes, with the first significant change occurring at about 350m, where the forest cover is nearly halved; by 650m, it further decreases to just a few sparse trees. To better perceive outlier classes, we set $h = 2$, as shown in (d). The pink in the circled area, can be identified more clearly. Taking a step further, we set $h = 10$, which is nearly the maximum effect achievable in this dataset, as shown in (c). At this setting, the density of outlier classes is close to but does not exceed that of
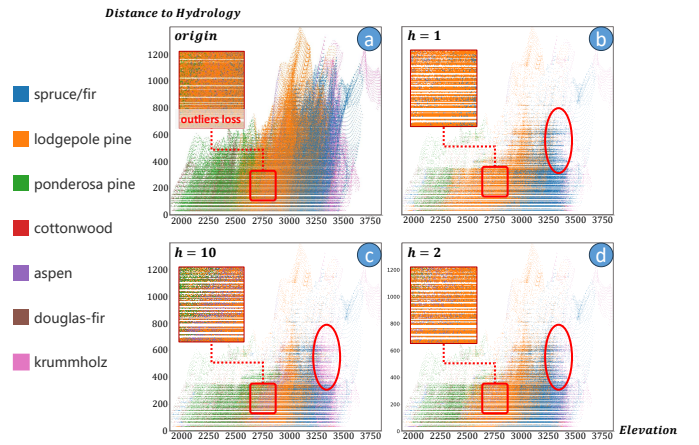
non-outlier classes, and the pink region within the ellipse can be clearly observed. We can observe the boundaries of semantic outlier classes in red box: ponderosa pine mainly spreads at elevations below 2300m, reaching up to about 2800m. Krumholz has a minor distribution from 2650m to 2850m in elevation, starting again in small amounts from 3250m up to the highest elevation.

6 DISCUSSION AND FUTURE WORK

Our method's effectiveness hinges on the perceptibility of its pixel-based output. While pixels offer maximum information granularity, we acknowledge that individual pixels, especially for outliers, can be difficult to discern. Our framework mitigates this through built-in tools like the Outlier Emphasis Pattern and an adjustable abstraction level (i.e., grid size). Beyond these static solutions, perceptibility could be further enhanced through interactive means like magnifying lenses or zooming, which we leave for future work. A related concern is that emphasizing outliers can potentially distort class proportions, creating visual chaos on small canvases. While our ST2 pattern helps prevent this, a more robust future refinement would be to avoid partitioning regions too small to faithfully represent class density.

Building on this work, we have several promising research directions. A seamless hybrid rendering model could be implemented to transition from the proposed pixelated abstraction (for global overview) to traditional point-based scatterplots (for local detail inspection) during zooming. Furthermore, the parallelizable nature of Iso-density Partitioning could be leveraged for significant performance acceleration or streaming data. Finally, the core principle of density equalization could be extended to other visualization variants, such as continuous-variable scatterplots or continuous density fields. For continuous-variable scatterplots, a direct approach is to apply density estimation within each region to generate a local probability density function. New points can then be sampled from this function and placed near original locations to maintain local structure. This sampling can also be tailored to up-sample from tails, thus enhancing outlier perception.

7 CONCLUSION

In this paper, we proposed PixelatedScatter, a visual abstraction method for large-scale multiclass scatterplots that effectively preserves features across a high dynamic range of data densities. Our method faithfully preserves the global relative regional density relations, enhances visual density contrast, and achieves a good balance between relative class density preservation and outlier highlighting. Moreover, our method can be suitable for arbitrary abstraction levels. We performed quantitative and qualitative evaluations, as well as a user study. Comparisons with state-of-the-art techniques demonstrate the advantages of our method.

REFERENCES

[1] A. Asuncion. Uci machine learning repository. Jan 2007. 5

[2] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, p. 1428–1435, Nov 2008. doi: 10.1109/tvcg.2008.119 1, 2

[3] E. Bertini and G. Santucci. Give chance a chance: Modeling density to enhance scatter plot quality through random data sampling. *Information Visualization*, p. 95–110, Jun 2006. doi: 10.1057/palgrave.ivs.9500122 2

[4] J. Blackard. Covertype. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C50K5N. 9

[5] D. B. Carr, R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield. Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82(398):424–436, 1987. doi: 10.2307/2289444 1

[6] H. Chen, W. Chen, H. Mei, Z. Liu, K. Zhou, W. Chen, W. Gu, and K.-L. Ma. Visual abstraction and exploration of multi-class scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, p. 1683–1692, Dec 2014. doi: 10.1109/tvcg.2014.2346594 2

[7] X. Chen, T. Ge, J. Zhang, B. Chen, C.-W. Fu, O. Deussen, and Y. Wang. A recursive subdivision technique for sampling multi-class scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, p. 729–738, Jan 2020. doi: 10.1109/tvcg.2019.2934541 1, 2, 3, 5, 6

[8] X. Chen, J. Zhang, C.-W. Fu, J.-D. Fekete, and Y. Wang. Pyramid-based scatterplots sampling for progressive and streaming data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):593–603, 2022. doi: 10.1109/TVCG.2021.3114880 2

[9] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, p. 1009–1016, Nov 2009. doi: 10.1109/tvcg.2009.122 1, 2

[10] R. Cutura, C. Morariu, Z. Cheng, Y. Wang, D. Weiskopf, and M. Sedlmair. Hagrid—gridify scatterplots with hilbert and gosper curves. In *Proceedings of the 14th International Symposium on Visual Information Communication and Interaction*, pp. 1–8, 2021. doi: 10.1145/3481549.3481569 3

[11] T. N. Dang, L. Wilkinson, and A. Anand. Stacking graphic elements to avoid over-plotting. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1044–1052, 2010. doi: 10.1109/TVCG.2010.197 2

[12] A. Dix and G. Ellis. by chance enhancing interaction with large data sets through statistical sampling. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, May 2002. doi: 10.1145/1556262.1556289 2

[13] Duderstadt, Brandon and Nomic AI. A Collection of Public Datasets on Nomic Atlas. https://atlas.nomic.ai/data/brandon, 2023. This citation refers to Nomic Ada Wiki, Online News, Tweets from Members of US Congress and X Community Notes. 5

[14] D. Feng, L. Kwock, Y. Lee, and R. Taylor. Matching visual saliency to confidence in plots of uncertain data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):980–989, 2010. doi: 10.1109/TVCG.2010.176 2

[15] D. Feng, L. Kwock, Y. Lee, and R. M. Taylor. Matching visual saliency to confidence in plots of uncertain data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):980–989, Nov 2010. doi: 10.1109/tvcg.2010.176 1, 2

[16] O. Fried, S. DiVerdi, M. Halber, E. Sizikova, and A. Finkelstein. Isomatch: Creating informative grid layouts. In *Computer graphics forum*, vol. 34, pp. 155–166. Wiley Online Library, 2015. doi: 10.1111/cgf.12549 3

[17] E. Gansner and Y. Hu. Efficient, proximity-preserving node overlap removal. *Journal of Graph Algorithms and Applications*, 14(1):53–74, 2010. doi: 10.7155/jgaa.00198 3

[18] R. C. Gonzales and P. Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987. 3

[19] H. Hagh-Shenas, S. Kim, V. Interrante, and C. Healey. Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. *IEEE transactions on visualization and computer graphics*, 13(6):1270–1277, 2007. doi: 10.1109/TVCG.2007.70623 1

[20] K. Hayashi, M. Inoue, T. Masuzawa, and H. Fujiwara. A layout adjustment problem for disjoint rectangles preserving orthogonal order. In *Graph Drawing: 6th International Symposium, GD'98 Montréal, Canada, August 13–15, 1998 Proceedings 6*, pp. 183–197. Springer, 1998. doi: 10.1007/3-540-37623-2_14 3

[21] G. M. Hilasaca, W. E. Marcílio-Jr, D. M. Eler, R. M. Martins, and F. V. Paulovich. Overlap removal of dimensionality reduction scatterplot layouts. *arXiv preprint arXiv:1903.06262*, 2019. doi: 10.1109/TVCG.2023.3309941 3

[22] G. M. Hilasaca, W. E. Marcílio-Jr, D. M. Eler, R. M. Martins, and F. V. Paulovich. A grid-based method for removing overlaps of dimensionality reduction scatterplot layouts. *IEEE Transactions on Visualization and Computer Graphics*, 30(8):5733–5749, 2024. doi: 10.1109/TVCG.2023.3309941 5

[23] J. Jo, F. Vernier, P. Dragicevic, and J.-D. Fekete. A declarative rendering model for multiclass density maps. *IEEE Transactions on Visualization and Computer Graphics*, p. 470–480, Jan 2019. doi: 10.1109/tvcg.2018.2865141 1, 2, 5

[24] D. A. Keim and A. Herrmann. *The gridfit algorithm: An efficient and effective approach to visualizing large amounts of spatial data*. IEEE, 1998. doi: 10.1109/VISUAL.1998.745301 3

[25] M. Krzywinski and B. Wong. Plotting symbols: choose distinct symbols that overlap without ambiguity and communicate relationships in data. *Nature methods*, 10(6):451–452, 2013. doi: 10.1038/nmeth.2490 2

[26] S. Lee, M. Sips, and H.-P. Seidel. Perceptually driven visibility optimization for categorical data visualization. *IEEE Transactions on visualization and computer graphics*, 19(10):1746–1757, 2012. doi: 10.1109/TVCG.2012.315 2

[27] B. Li, D. A. Griffith, and B. Becker. Spatially simplified scatterplots for large raster datasets. *Geo-spatial Information Science*, p. 81–93, Apr 2016. doi: 10.1080/10095020.2016.1179441 1, 2

[28] J. Li, J.-B. Martens, and J. J. van Wijk. A model of symbol size discrimination in scatterplots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Apr 2010. doi: 10.1145/1753326.1753714 2

[29] Z. Li, R. Shi, Y. Liu, S. Long, Z. Guo, S. Jia, and J. Zhang. Dual space coupling model guided overlap-free scatterplot. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):657–667, 2022. doi: 10.1109/TVCG.2022.3209459 3

[30] Z. Li, C. Zhang, S. Jia, and J. Zhang. Galex: Exploring the evolution and intersection of disciplines. *IEEE Transactions on Visualization and Computer Graphics*, p. 1–1, Jan 2019. doi: 10.1109/tvcg.2019.2934667 1, 2

[31] S. Liu, J. Xiao, J. Liu, X. Wang, J. Wu, and J. Zhu. Visual diagnosis of tree boosting methods. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):163–173, 2018. doi: 10.1109/TVCG.2017.2744378 2

[32] M. Luboschik, A. Radloff, and H. Schumann. A new weaving technique for handling overlapping regions. In *Proceedings of the International Conference on Advanced Visual Interfaces*, May 2010. doi: 10.1145/1842993.1842999 1, 5

[33] A. Mayorga and M. Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, p. 1526–1538, Sep 2013. doi: 10.1109/tvcg.2013.65 2

[34] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183–210, 1995. doi: 10.1006/jvlc.1995.1010 3

[35] L. Nachmanson, A. Nocaj, S. Bereg, L. Zhang, and A. Holroyd. Node overlap removal by growing a tree. In *Graph Drawing and Network Visualization: 24th International Symposium, GD 2016, Athens, Greece, September 19-21, 2016, Revised Selected Papers 24*, pp. 33–43. Springer, 2016. doi: 10.1007/978-3-319-50106-2_3 3

[36] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, Sep 2006. doi: 10.1109/tvcg.2006.170 2

[37] K. Panetta, C. Gao, and S. Agaian. No reference color image contrast and quality measures. *IEEE transactions on Consumer Electronics*, 59(3):643–651, 2013. doi: 10.1109/TCE.2013.6626251 6

[38] R. G. Raidou, M. E. Gröller, and M. Eisemann. Relaxing dense scatter plots with pixel-based mappings. *IEEE transactions on visualization and computer graphics*, 25(6):2205–2216, 2019. doi: 10.1109/TVCG.2019.2903956 3, 5, 6

[39] A. Sarikaya and M. Gleicher. Scatterplots: Tasks, data, and designs. *IEEE Transactions on Visualization and Computer Graphics*, p. 402–412, Jan 2018. doi: 10.1109/tvcg.2017.2744184 1, 3

[40] M. Tory, D. Sprague, F. Wu, W. Y. So, and T. Munzner. Spatialization design: Comparing points and landscapes. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1262–1269, Nov 2007. doi: 10.1109/tvcg.2007.70596 1, 2

[41] L. Tremmel. The visual separability of plotting symbols in scatterplots. *Journal of Computational and Graphical Statistics*, 4(2):101–112, 1995.

doi: 10.2307/1390760 2

[42] Y. Wang, X. Chen, T. Ge, C. Bao, M. Sedlmair, C.-W. Fu, O. Deussen, and B. Chen. Optimizing color assignment for perception of class separability in multiclass scatterplots. *IEEE transactions on visualization and computer graphics*, 25(1):820–829, 2018. doi: 10.1109/TVCG.2018.2864912 2

[43] L. Wilkinson. *The Grammar of Graphics*. Jan 1999. 2

[44] A. Woodruff, J. Landay, and M. Stonebraker. Constant density visualizations of non-uniform distributions of data. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, Nov 1998. doi: 10.1145/288392.288397 2

[45] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu. Interactive correction of mislabeled training data. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct 2019. doi: 10.1109/vast47406.2019. 8986943 2

[46] J. Yuan, S. Xiang, J. Xia, L. Yu, and S. Liu. Evaluation of sampling methods for scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1720–1730, 2021. doi: 10.1109/TVCG.2020.3030432 2, 7