Dual Space Coupling Model Guided Overlap-Free Scatterplot

Zeyu Li, Ruizhi Shi, Yan Liu, Shizhuo Long, Ziheng Guo, Shichao Jia, and Jiawan Zhang, Senior Member, IEEE

Abstract—The overdraw problem of scatterplots seriously interferes with the visual tasks. Existing methods, such as data sampling, node dispersion, subspace mapping, and visual abstraction, cannot guarantee the correspondence and consistency between the data points that reflect the intrinsic original data distribution and the corresponding visual units that reveal the presented data distribution, thus failing to obtain an overlap-free scatterplot with unbiased and lossless data distribution. A dual space coupling model is proposed in this paper to represent the complex bilateral relationship between data space and visual space theoretically and analytically. Under the guidance of the model, an overlap-free scatterplot method is developed through integration of the following: a geometry-based data transformation algorithm, namely DistributionTranscriptor; an efficient spatial mutual exclusion guided view transformation algorithm, namely PolarPacking; an overlap-free oriented visual encoding configuration model and a radius adjustment tool, namely \( f_{\text{draw}} \). Our method can ensure complete and accurate information transfer between the two spaces, maintaining consistency between the newly created scatterplot and the original data distribution on global and local features. Quantitative evaluation proves our remarkable progress on computational efficiency compared with the state-of-the-art methods. Three applications involving pattern enhancement, interaction improvement, and overdraw mitigation of trajectory visualization demonstrate the broad prospects of our method.

Index Terms—Scatterplot, overdraw, overlap-free, scalability, circle packing

1 INTRODUCTION

For 2D scatterplot visualization, maintaining high-quality data distribution while avoiding overdraw is still an unsolved problem.

Depending on the space in which the core operation is performed, existing solutions toward overdraw problem can be classified into three categories: data space, visual space, and hybrid methods. First, data space methods perform data transformation such as trimming, filtering, sampling, or aggregating operation, on the original data points to reduce the data volume. However, the asymmetrical correspondence between the data points and the visual units in visual space objectively introduces an endogenous contradiction between reducing overdraw and maintaining a lossless and unbiased data distribution. Second, visual space methods mainly focus on applying visual encoding adjustment and view transformation by elaborately configuring the size, position, transparency, or other visual channels of visual units. These methods can then be further classified into three sub-categories: node appearance adjustment [51] [27] whose strategy is to reduce the size and transparency of nodes; node dispersion [18] [20] [38] [47] which distributes nodes in an iteration process based on a physical or a mathematical optimization model; sub-space mapping [24] [15] [19] which injects the data nodes into a partition of visual space. However, adjusting the appearance of nodes cannot strictly avoid overlap, and the color blending caused by transparency leads to severe visual complexity. The two latter sub-categories may introduce serious distortions because they disregard the density preservation. Third, hybrid methods, such as bin aggregation [23] [1] [30] and contour map [14] [32] [28], relieve overdraw by replacing visual units that originally correspond to individual data points with visual objects, such as polygons and paths with higher level of visual abstraction. Abstraction leads to the loss of details.

The two major drawbacks of the existing methods lie in the asymmetrical correspondence between the data points and visual units and the inconsistency between the original data distribution and the distribution presented in the scatterplot. To obtain a complete solution
to scatterplot overlap problem, first, the data-visual space mapping should be unbiased or lossless. No data points can be discarded, and all data points should correspond to a unique visual unit. Second, principles and guidelines should also be carefully developed to ensure the strict overlap elimination and safe interaction\(^1\) to handle the scalability issue of data and support interactive exploration in visual space. Third, as the fundamental goal, the final scatterplot should accurately reflect the original data distribution. The existing overlap removal methods also pursue similar objectives but they always seriously sacrifice one of them, leading to the realization of some goals but accompanied by serious negative effects.

In this paper, after re-examining the overlap problem through a theoretical perspective, we regard it as an informal optimization problem under four formal criteria. In addition to emphasizing the safety of operations performed within a single space, a dual space coupling model is also proposed to represent the complex bilateral relationship between data and visual spaces analytically. Under the guidance of this model, we develop an overlap-free scatterplot visualization method comprising an unbiased and lossless geometry-based transcriptor that transcribes the data distribution into a set of discrete circles, an efficient circle packing algorithm that re-layouts these circles in visual space to ensure spatial mutual exclusion and reproduce the transcribed distribution to a tangible scatterplot, and a visual encoding configuration model to optimize the visual quality of the new scatterplot and ensure interaction safety. The proposed method can ensure complete and accurate information transfer between the data and visual spaces, maintaining consistency between the newly created scatterplot and the original data points on global and local features. Quantitative evaluations are conducted to compare with the state-of-the-art methods on time cost and five metrics that are designed for measuring the capability to preserve the features of original scatterplots. Three applications demonstrate the capability of our method to reveal the pattern hidden by overlap, improve the efficiency of interactive exploration, and mitigate the overlap problem in trajectory visualization.

The contributions of this paper can be summarized as follows:

- We propose a dual-space coupling model to represent the complex relationship and design considerations within and between the data and visual spaces theoretically and analytically. The model introduces a new design space for promising overlap removal algorithms and interaction paradigms.

- We propose an overlap-free scatterplot method which integrates \textit{DistributionTranslator} (a geometry-based data transformation algorithm), \textit{PolarPacking} (an efficient circle packing algorithm), and a visual encoding configuration model.

- We develop an easy-to-use radius adjustment tool \(f_{\text{radius}}\) on the basis of the configuration model to improve the visual quality of scatterplot and ensure interaction safety.

2 \textbf{RELATED WORK}

Under the topic of visual enhancement of scatterplots [46] [8], this paper focuses on methods to mitigate overlap.

2.1 Data Space Methods

Data space methods simply focus on data operation and completely ignore the stuff on visual units. Therefore, these methods simply pertain to the data transformation described in the classic visualization pipeline [7]. Data reduction and jitter are two typical data space methods.

Data reduction methods alleviate overlap by reducing data points, thus decreasing the visual units to be placed in visual space. Data sampling and aggregation are two commonly used data reduction strategies. Data sampling selects representative samples from the full set, while data aggregation aggregates subsets of the full set into newly created data points. However, they both suffer from inherent flaws on data loss, data bias, and visualization. The data loss is straightforward, while the data bias is caused by the unavoidable selection of data or/and goal. For example, data sampling methods have developed diverse sampling strategies but have all been designed for specific goals, such as maintaining relative density among regions [3] [13] [26] [4], emphasizing the spatial separation of samples [11] [50] [2], and preserving outliers [29] [12] [52]. No one-size-fits-all sampling strategy exists. Hence, the choice of a goal/strategy causes bias. Data reduction methods pose a fundamental conflict between avoiding overlap and preserving unbiased and lossless data distribution. Moreover, data reduction methods cannot eliminate overlaps because they completely ignore the size of visual units.

Typical jitter [48] alleviates overlap by randomly spreading data points in data space. However, jitter is unstable and cannot materially overcome overlap. Instead, jitter may lose meaningful data features or even cause more serious overlap.

2.2 Visual Space Methods

In contrast to data space methods, visual space methods focus entirely on visual units in visual space, reducing overlap by optimizing their appearance or position. These methods ensure one-to-one correspondence between data points and visual units, avoiding information loss at the data level. Therefore, these methods come down to visual encoding and view transformation operation following the classic visualization pipeline [7]. Appearance adjustment, node dispersion, and subspace mapping are three common visual space methods.

Appearance adjustment reduces overlap by decreasing the size and transparency of nodes [51] [27]. However, the adjustment usually binds with data, requiring time-consuming customization. Several recent semi-automatic techniques [34] [40] reduce the workload but are currently focused on single-class scatterplots. For multi-class scatterplots, transparency opens Pandora’s box of color blending, which markedly increases the visual complexity and significantly hinders visual tasks, such as class identification and class density comparison. Essentially, appearance adjustment cannot eliminate overlaps.

Node dispersion relieves overlap by spreading nodes from their original positions. Many dispersion strategies have been proposed for different layout goals in graph visualization. For example, VPSC [18], PFS [35], PFS’ [22], PRISM [20], and FTA [25] iteratively approach an ideal dispersion through force-directed or gradient descent techniques. These methods are good at preserving the orthogonal order of input nodes. GTree [38] declares a better dispersion on efficiency and shape preservation by growing a minimum spanning tree built on the Delaunay triangulation of nodes. However, GTree reduces space utilization. By contrast, RWordle-L [47] achieves a compact layout by placing nodes along a spiral curve under the constraint of mutual exclusion. Diamond [33] develops a stable layout process but sacrifices shape and density preservation. Overall, node dispersion methods usually suffer from three common problems. First, severe distortion frequently occurs, preventing basic visual tasks, such as cluster identification and trend analysis, because these methods do not consider density and shape preservation as mandatory constraints. Second, these methods cannot eliminate overlap due to early termination or falling into local optimum. Third, these methods are computationally inefficient; thus they are only applicable to small data sets. The three problems are exposed by the quantitative evaluation in Section 5.1.

Subspace mapping methods developed in recent years are the few ones that can eliminate overlap. Typically, these methods first divide visual space into a set of mutually-exclusive subspaces and then map each data point to a subspace by aligning the spatial proximity of two spaces. Subspaces can be generated by isometric grids (DGrid [24], IsoMatch [19], Oodanalyzer [9]), space-filling curves (HuGrid [15]), or space-filling treemaps (Nmap [17]). Compared with node dispersion methods, subspace mapping methods are generally faster and perform better in shape and density preservation. However, the loose coupling of space partition and data distribution leads to severe distortion on shape and density in regions with high density. The distortion can be observed in the quantitative evaluation in Section 5.2. By contrast, our method is even faster and maintains a high quality data distribution.

\(^1\) no overlap occurs during the interaction
2.3 Hybrid Methods

Hybrid methods perform data processing and conduct visual encoding or/and view transformation. Bin aggregation [23] [1] [30] and contour map [14] [32] [28] are two representative hybrid methods. In data space, bin aggregation performs data aggregation on data subsets divided by location, while contour map extracts a series of density stairs and corresponding boundaries according to density distribution. Information loss inevitably occurs. In visual space, the two methods discard circular nodes and encode extracted abstract information into visual units with higher level abstraction, such as polygons and paths. Consequently, the one-to-one correspondence between data points and visual units is destroyed. Our method is a hybrid method because it performs data and view transformations. Nevertheless, our method avoids the two aforementioned problems.

3 Theory Formulation

3.1 Dual Space Coupling Model

The overdraw problem substantially damages the effectiveness of scatterplot visualization which is reflected by severely hampered common visual tasks of scatterplots, such as cluster identification, shape exclusion performed in visual space even if nothing is performed in data space. The latter two criteria require the correspondence and consistency between data and visual spaces and are mandated.

Therefore, the goal of a desired scatterplot overdraw solution can be expressed as obtaining a high-quality reconstruction of the data distribution in visual space while ensuring mutual exclusion of data points (optional), mutual exclusion of visual units (mandatory), and data-visual space bijection. This goal can be formalized as:

\[
\text{argmax}(\text{similarity}(F_{V}(NS), F_{D}(DS))), \text{ s.t. } C_{1}, C_{2}, C_{3}, C_{4}
\]

Based on this formulation, we set up a theoretical framework to evaluate overlap removal methods. Table 1 presents a summary for comparison between our method and the existing methods. Most existing methods do not meet at least one criterion, which alleviate the overlap but are accompanied by serious negative effects. Then, we propose a dual space coupling model to represent the complex operations, transformations, and design considerations within and between data space and visual space. The model is conceptually illustrated in Fig. 2.

The proposed model analytically and theoretically clarifies the design and evaluation principles for feasible solutions. In addition to accommodating the existing techniques, our model introduces a new design space for promising overlap removal algorithms and interaction paradigms and further provides several guidelines for design practice. First, the mutual exclusion of data points cannot guarantee the mutual exclusion of visual units, and the latter can only be achieved by assigning a reasonable size and location for visual units. Furthermore, the latter should not presuppose the former. Second, pursuing the data distribution preservation of visual space is a good starting and driving force to achieve an ideal assignment under the premise of ensuring data lossless and unbiased. Third, visual quality, or visual prominence of visual units, is another important consideration, requiring a trade-off between large visual units and little disruption to distribution preservation.

A safe and fast interactive radius configuration tool is necessary to achieve a customized trade-off. The distribution consistency between data and visual spaces is not trivial, and we will discuss metrics of consistency in the next section.

3.2 Metrics

In this section, we present a measurement framework to measure the similarity in Formula 1 quantitatively. The framework is inspired by the visual task of scatterplots and the existing metrics [10] [24]. This framework comprises an overall metric and four sub-metrics. The overall metric presents a general perspective to measure the similarity of two scatterplots, while each sub-metric has a clear and independent semantics toward visual tasks. Specifically, displacement minimization and KNN preservation measure the local feature of scatterplots, which are closely related to the visual task of querying and inspecting similar data points. Density preservation and shape preservation focus on the global feature of scatterplots. The two metrics are designed for visual

### Table 1: Comparison with existing representative methods based on the four criteria mentioned in Section 3.1.

| Categories / Criteria | C1 | C2 | C3 | C4 |
|-----------------------|----|----|----|----|
| Data space methods    | ✓  | ✓  | ×  | ✓  |
|                       | ✓  | ✓  | ✓  | ✓  |
| Visual space methods  | ×  | ✓  | ✓  | ×  |
|                       | ×  | ✓  | ✓  | ✓  |
| Hybrid methods        | ×  | ×  | ×  | ✓  |

Table: Comparision with existing representative methods based on the four criteria mentioned in Section 3.1. ✓ means “close to be perfect” or “is very helpful” but cannot strictly meet the criterion.

---

**Fig. 2:** Conceptual illustration of our dual space coupling model.
tasks such as cluster identification, outlier detection, and trend analysis.

Given the original scatterplot \( S = \{ p_i \}_N \) and the re-located scatterplot \( S' = \{ p'_i \}_N \), where \( p_i \) and \( p'_i \) are a pair of corresponding data points in the two scatterplots and \( N \) denotes the number of data points. Each metric is briefly described below.

Displacement minimization is the same as [10]. We calculate the average of the Euclidean distance between all pairs of \( p_i \) and \( p'_i \) after scaling \( S \) and \( S' \) to the same size and aligning their centers. Then the relative displacement, that is, the ratio of the average to the width of the bounding box of \( S \), is taken as the final score. The score ranges in \([0, +\infty)\); a small score is superior.

KNN preservation is simply calculated by \( \frac{1}{N} \sum_{i=1}^{N} \left| \frac{\text{knn}(p_i)}{\text{knn}(p'_i)} - 1 \right| \), where \( \text{knn}(p_i) \) represents the k-nearest neighbors of point \( p_i \) in \( S \). The score ranges in \([0, 1] \). The KNN is effectively preserved when the score is close to 1.

Shape preservation extends from [47] by expanding the measured shape from the outermost contour of the scatterplot to a global scope. We first construct a standard shape in \( S \), and then observe whether the shape can be maintained in \( S' \). Specifically, as shown in the left, we first construct a group of concentric circles \( C_j \) in \( S \), and then find a group of points \( \{ p_i \}_k \) near each circle in \( S \). The \( k \) denotes the index of circles. Next, we compute the variance of the distance \( \Delta p_i \) between \( \{ p'_i \}_k \) and the corresponding center \( C_j \) for each circle. Obviously, a small average of the variances facilitates superior shape preservation. The score ranges in \([0, +\infty)\).

Density preservation is a new metric that measures the global density preservation by measuring the preservation of relative position of all points in an ordered sequence. Specifically, as shown in the left, for each point \( p_i \), we regard the reciprocal of its average distance from its KNN, that is \( \frac{1}{\text{avg dist}} \), as its local density. Then we calculate the quantile \( q(p_i) \) of the point \( p_i \) in the ordered sequence of all points sorted by the average distance. Therefore, the density preservation score is the average of the difference \( \Delta q \) between the quantiles of all paired points in \( S \) and \( S' \). The score ranges in \([0, 1] \). A score close to 0 is superior.

Overall similarity If \( S \) and \( S' \) look similar from multiple viewing angles, then they are similar overall. As shown in the left, we first project \( S \) and \( S' \) on the same set of directions, forming a series of paired ordered index sequences: \( \{ \text{seq}_j \} \) and \( \{ \text{seq}'_j \} \). These directions act as the viewing angles. The Kendall correlation coefficient [45] is then computed for each pair of the sequences as the similarity between \( S \) and \( S' \) on the corresponding angle. The final score is the average of all similarities and ranges from -1 to 1. A score close to 1 indicates a high overall similarity between \( S \) and \( S' \).

4 METHODS

On the basis of the dual-space model, we propose a method to build an overlap-free scatterplot and ensure its interaction safety. The core idea is to use a set of closely tangent circles (NS) in visual space to imitate the original data distribution in data space \( F_D(DS) \) in Formula 1. Some of these circles act as placeholders, filling the blank areas of \( F_D(DS) \). These circles are called dummy nodes and will not be rendered in visual space. The remaining circles, which are called data nodes, are in one-to-one correspondence with the data points. In addition to the radius \( r_{\text{pack}} \) used for packing, each data node has another radius \( r_{\text{draw}} \) used for rendering. \( r_{\text{pack}} \) and \( r_{\text{draw}} \) are designed to guarantee mutual exclusivity and visibility of nodes, respectively.

The core idea refers to the following three questions: (1) How to generate a set of circles that can record intact \( F_D(DS) \); (2) How to pack these circles sequentially in visual space to express the recorded \( F_V(DS) \) as observable \( F_V(\text{NS}) \); (3) How to configure \( r_{\text{draw}} \) to ensure no overlap occurs (safety) during rendering and interaction. Borrowing concepts from genetics, the first question aims to transcribe the original data distribution from data space to visual space; the second question aims to translate the transcribed distribution from an algebraic and intangible form into a visible and tangible form; the third question aims to modify the distribution in visual space. Fig.3 shows the pipeline of our method. A geometry-based data distribution transcription, an efficient spatial mutual exclusion guided view transformation, and an overlap-free oriented visual encoding configuration with an easy-to-use radius adjustment tool are the solutions to the three aforementioned problems. In the following, they will be introduced one by one.

4.1 Geometry-Based Data Distribution Transcription

The transcription must maintain the relative position of nodes and the relative density among regions. For the former, we simply inherit the original coordinates of the input data points; for the latter, we borrow the idea of frequency modulation (FM) halftoning, a technique widely used in traditional printing industry [6]. Fig.4 shows an example of FM halftoning. The example simulates continuous-tone imagery through the use of dots that are frequency varying in spacing, thus generating a gradient-like effect.

We use the similar idea to rebuild varying densities. First, We divide the space into square grids, and then fill each grid with circles. The grid containing \( \text{num} \) data points will be filled with \( \text{max}(k, \text{num}) \) circles. The packing radius \( r_{\text{pack}} \) of each circle is equal, given by line 6 of Algorithm 1. Herein \( \text{size} \) denotes the size of grids, and \( k \) is a threshold representing the minimum number of nodes to be packed in a grid. \( \text{size} \) and \( k \) are all the parameters of our method. \( k \) is set to prevent potential distortions in sparse regions. Extra \( (k - \text{num}) \) dummy nodes acting as placeholders are generated for the grid with less than \( k \) data points. The radius of these dummy nodes is also \( r_{\text{pack}} \). We assign an attribute called \( \text{density} \) to each data node. The attribute value is given by the ratio of the number of nodes \( \text{num} \) in the corresponding grid and the maximal \( \text{num} - \text{num}_{\text{max}} \). The default value of \( r_{\text{draw}} \) is \( r_{\text{pack}} \), that is, the minimum \( r_{\text{pack}} \), obtained in the case of the density is 1. The above settings globally ensure the relative density among regions.

We call a collection of points with the same coordinates thus causing overlaps in data space a singularity. As shown in the upper left corner of Fig.3, we spread each singularity into phyllotaxis to achieve the mutual exclusion of data points. Note that as stated in Section 3.1, the spread operation is not mandatory. It can be optionally performed before the gridding. However, the spread induces the appearance of singularities as circular agglomerate fogs in visual space which stand out from regular nodes, as if highlighting anomalies.

Algorithm 1 provides the specific steps of the transcription. The input of the algorithm is the original 2D dataset \( DS = \{ x, y \}_N \), where \( N \) is the number of data points. The output is \( NS = \{ (x, y, r_{\text{pack}}, r_{\text{draw}}, \text{density}) \}_N \), where \( N' \) is the number of nodes to be packed. Hence, \( N' - N \) is the number of generated dummy nodes. The red highlights the changes in \( NS \) compared with \( DS \).
The goal defined by Formula 1 can be achieved using $n^2$,$\Phi$ pack of nodes cannot guarantee the mutual exclusion of nodes. This section presents PolarPacking (Algorithm 2), an algorithm that achieves the required mutual exclusion while maintaining the relative position of nodes. The idea behind the algorithm is to reconstruct the distribution transcribed in $NS$ in polar coordinates.

PolarPacking is modified from CirclePacking [49]. CirclePacking is a layout technique that compactly packs circles without considering their relative positions. PolarPacking must maintain the relative position of the nodes while packing them compactly. As shown in the middle of Fig.3, we first calculate the distance and angle of each node relative to the center of the nodes to convert the relative position encoded in the Cartesian to the polar coordinate system. We then individually pack the nodes in ascending order by their distance. Each node is placed tangent to the outline shaped by the already packed nodes, following its tangent to the outline. During the process, the nodes in ascending order by their distance. Each node is placed Cartesian to the polar coordinate system. We then individually pack $NS$ in $\Omega$ and an interactive radius adjustment tool based on the configuration model and an interactive radius adjustment tool. In this section, we present a visual encoding configuration model and an interactive radius adjustment tool based on the model to solve the two problems by safely configuring $r_{draw}$ at a expense of distorting density distribution of extraordinary regions.

### 4.2 Spatial Mutual Exclusion Guided View Transformation

Directly rendering the $NS$ from the previous step in visual space is not feasible because the current coordinates and $r_{pack}$ of nodes cannot guarantee the mutual exclusion of nodes. This section presents PolarPacking (Algorithm 2), an algorithm that achieves the required mutual exclusion while maintaining the relative position of nodes. The idea behind the algorithm is to reconstruct the distribution transcribed in $NS$ in polar coordinates.

PolarPacking is modified from CirclePacking [49]. CirclePacking is a layout technique that compactly packs circles without considering their relative positions. PolarPacking must maintain the relative position of the nodes while packing them compactly. As shown in the middle of Fig.3, we first calculate the distance and angle of each node relative to the center of the nodes to convert the relative position encoded in the Cartesian to the polar coordinate system. We then individually pack the nodes in ascending order by their distance. Each node is placed tangent to the outline shaped by the already packed nodes, following its tangent to the outline. During the process, the nodes in ascending order by their distance. Each node is placed Cartesian to the polar coordinate system. We then individually pack the nodes in ascending order by their distance. Each node is placed tangent to the outline. During the process, the distance and angle play key roles in maintaining the relative position of nodes, and the $r_{pack}$ and tangency guarantee strict mutual exclusion. The detailed packing process is presented in Supplementary Material 1.

After filtering out dummy nodes, we get the output of PolarPacking $NS' = \{(x, y, r_{pack}, r_{draw}, density)\}_k$. The coordinates of nodes in $NS'$ are updated, while the value of other attributes is simply inherited from the previous $NS$. As this point, the goal defined by Formula 1 is achieved if a scatterplot is rendered using $NS'$.

### 4.3 Overlap-Free Oriented Visual Encoding Configuration

The goal defined by Formula 1 can be achieved using $NS'$, but the default $r_{draw}$, that is, $r_{pack}^1$, is inappropriate in some scenarios. For example, consider a high dynamic range dataset (HDR dataset) embedded with regions with extremely high density, which is common for large-scale datasets, its default $r_{pack}^1$ may seriously reduce the visual quality of the corresponding scatterplot. The first row of Fig.7 shows an example, where the “contrast” of the entire scatterplot is sharply reduced. The reason behind this is that the default $r_{pack}^1$ is usually very small for HDR datasets, leading to an extremely low fill rate of colored informative pixels in most regions. Moreover, outliers are likely to be ignored in this case. In this section, we present a visual encoding configuration model and an interactive radius adjustment tool based on the model to solve the two problems by safely configuring $r_{draw}$.

#### Algorithm 1: DistributionTranscriptor

```plaintext
input :
DS : \{(x, y)\}_N \in R^2_+      \quad \text{// original 2D data}
size \in R_+      \quad \text{// size of grids}
k \in N_+      \quad \text{// minimum number of node in a grid}
output :
NS : \{(x, y, r_{pack}, r_{draw}, density)\}_N' \in R^2_+      \quad \text{// node set}
1. DS \leftarrow \text{spreadSingualrities(DS)}
2. NS \leftarrow \emptyset
3. grids : \{(i, j, subDS)\}_{n \times n} \leftarrow \text{gridding(DS, size)}
4. for grid in grids do
5.   num \leftarrow \#grid.subDS
6.   r_{pack} \leftarrow \sqrt{\frac{\text{size}^2}{\pi \times \max(k, \text{num})}}
7.   density \leftarrow \frac{num}{\text{subDS}\times \text{size}} \in [0, 1]
8. for p in subDS do
9.   dataNode \leftarrow (p.x, p.y, r_{pack}, density)
10. NS \leftarrow NS \cup \{dataNode\}
11. end
12. while num < k do
13.   x \leftarrow \text{grid}.j \times \text{size} + \text{random}(0, \text{size})
14.   y \leftarrow \text{grid}.i \times \text{size} + \text{random}(0, \text{size})
15.   dummyNode \leftarrow (x, y, r_{pack})
16.   NS \leftarrow NS \cup \{dummyNode\}
17.   num \leftarrow num + 1
18. end
19. n,r_{draw}\leftarrow \text{min}(\{n,r_{pack}\}), n \in NS \quad \text{// set the default r_{draw}}
20. end
21. return NS
```

The configuration model is shown in Fig.5. The X-axis and Y-axis represent the density and r of nodes, respectively. The relationship between density and $r_{pack}$ is given by line 6 and 7 of Algorithm 1 is a smooth decreasing curve when num $\geq k$. The curve depicts the safe supremum of $r_{draw}$: $r_{pack}^1$ is the infimum of $r_{draw}$. The line $d = d_k$ and $r = r_{pack}^1$ divide the entire quadrant into four zones. $d_k$ is the density of the grid with k data points. Donate $r_{draw}$ and $r_{pack}$ are the specific $r_{draw}$ and $r_{pack}$ of regions with density d, then $r_{draw}^d$ satisfies $r_{pack}^1 \leq r_{draw}^d \leq r_{pack}^1$ in the safe zone; therefore, no overlap occurs. The case is the opposite in the unsafe zone. In the sparse zone (num $< k$), the $r_{pack}^d$ that acts as the supremum of $r_{draw}$ no longer exists. The restricted zone indicates that $r_{draw}^d$ should not be less than $r_{pack}^1$.

Based on the configuration model, we designed an interactive radius adjustment tool called $r_{draw}^d$. As shown in Fig.6, by moving the high-

![Fig. 3: Pipeline of our method. The three components transcribe, translate, and modify data distribution from data space to visual space. The algebraic data scheme below presents the evolution of the data flow. The changes between steps are highlighted in red.](image)

![Fig. 4: Example of FM halftoning.](image)
Algorithm 2: PolarPacking

\[
\begin{align*}
\text{input} & : & N_S &: \{(x, y, r_{pack}, r_{draw}, \text{density})\} & \forall n \in N \cup \{\text{center}\} & \forall n \in N \cup \{\text{center}\} \\
\text{output} & : & N_S' &: \{(x, y, r_{pack}, r_{draw}, \text{density})\} & \forall n \in N \cup \{\text{center}\}
\end{align*}
\]

1. \(N_S' \leftarrow \emptyset\)
2. \(\text{center} \leftarrow \text{center of } N_S\)
3. \(\text{for } n \in N_S \text{ do}\)
4. \(\text{if } n.d.is \leftarrow \text{distance}(n, \text{center}) \text{ then}\)
5. \(\text{if } n.d.angle \leftarrow \text{angleToXPositive}(n, \text{center}) \text{ then}\)
6. \(\text{end}\)
7. \(\text{ascendingSortByDistance}(N_S)\)
8. \(\text{frontChain} \leftarrow \text{initialize}(N_S[3])\)
9. \(\text{for } n \in N_S[3] \text{ do}\)
10. \(\text{subChain} \leftarrow \text{a subchain of length } 2 + \text{th centered at } n.d.angle\)
11. \(\text{the position that has the smallest angle difference to } n.d.angle \text{ and tangents to only two nodes on the subChain}\)
12. \(\text{updateFrontChain}(n)\)
13. \(N_S' \leftarrow N_S' \cup \{n\}\)
14. \(\text{end}\)
15. \(\text{filterOutDummyNodes}(N_S')\)
16. \(\text{return } N_S'\)

The second and third columns show the rendering results with the default \(r_{pack}\) and adjusted \(r_{draw}\), respectively. The visual quality of the scatterplot of HDR datasets (the first row) and the visual prominence of outliers (the second row) have been markedly improved.

---

5 Evaluation

Quantitative evaluation compares the performance of our method with state-of-the-art methods on time cost and the five metrics introduced in Section 3.2 using 50 real-world scatterplots with entirely different distributions. The effectiveness of our method is further demonstrated in qualitative evaluation by showing scatterplots of several representative datasets and the improvements we made in three applications.

5.1 Quantitative Evaluation

**Competing Algorithms, Datasets and Settings** Competing algorithms include node dispersion methods, namely PFS’ [22], PRISM [20], GTree [38], and RWordle-L [47], and subspace-mapping methods, namely HaGrid [15] and DGrid [24]. Related algorithms, such as VPSC [18] and Diamond [33], are disregarded due to their unacceptable time costs in practice.

We collected 50 real-world datasets from [53], [12], UCI data repository, network repository, and our previous visualization projects. The
Fig. 8: Twelve samples of the collected 50 real-world datasets.

datasets mainly involve the following four types: regular scatterplots with two semantic axes, projection results of high-dimensional data, coordinates from geographic space, and layout results of large-scale graphs. The number of data points ranges from 4177 to 928,991. Fifteen of them exceed 100k. The data distribution of these datasets are distinct and some are quite challenging, such as the datasets embedded with extremely high density regions and those with significant features, such as clusters, paths, and textures. Some datasets are shown in Fig.8.

In addition to the full datasets, we created a relatively small data collection, namely Sampled3k, due to the unbearable low computational efficiency of the competing node dispersion methods. This collection is built by randomly sampling 3000 points from each full dataset. The comparison with the node dispersion methods and sub-space mapping methods is performed on the Sampled3k and full datasets, respectively. All scatterplots are rendered in an 800 × 800 canvas. We take 2.4 as the size of nodes for the node dispersion methods because it strikes a balance between overdrawing mitigation and outlier observability.

The implementations of the node dispersion methods are all from [10], with default parameters. The implementation of HaGrid comes from [15]. We chose Hilbert curve as the space filling curve and set the depth level of the curve to \( l_{min} + 1 \) as suggested by the original paper [15]. DGrid is implemented by ourselves. Considering the volume of the collected datasets, the number of rows and columns are set to 2000 and the size of the convolution kernel is set to 31 × 31, which is a trade-off between computational efficiency and the visibility of local details. The \( k \) and size parameters of our method are fixed to 3 and 5, respectively. All the above algorithms are written in JavaScript. The experiment environment is Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz, 64G RAM. The metrics used include time cost and the five size parameters on the five metrics in our measurement framework. Fig.11 presents the results. The range of size, \( k \), and sampling rate is set to [1, 10], [1, 20], and [0, 1], respectively. By mapping these ranges linearly to the same length, the impact curves of all parameters corresponding to the same metric can be aligned in one subfigure. For each curve, the value of the corresponding metric increases along the X-axis, while the others are fixed at their defaults. Noticing the small fluctuation ranges of all metrics on the Y-axis and recalling their intrinsic ranges, we declare that the impact of the parameters on all metrics are controllable, general, and reassuring. In other words, our method is fairly robust. Furthermore, we notice that the parameter size has a larger impact than \( k \) and sampling rate, and all metrics get worse as it raises. This observation is reasonable because size, representing the size of grids, determines the global resolution of the captured structures hidden in the data distribution. A small size facilitates the detection and depiction of refined structures by our method. Accordingly, the parameter \( k \), representing the minimum number of nodes to be placed in a grid, acts as a local resolution and only shows a notable impact on Density Preservation and KNN Preservation that depict local features. Interestingly, similar to \( k \), sampling rate only affects Density Preservation and KNN Preservation; and the two metrics get worse as it raises.

5.2 Qualitative Evaluation

Compared with quantitative metrics, viewing scatterplots is an intuitive evaluation method. As shown in the first column of Fig.12, our method preserve global and local features of the original dataset, such as planes and textures. The subtle textures, equidistant intervals, and circular singularities shown in the last three columns prove the capability of our method to "reproduce" the intrinsic details covered by the tool. Fig.1 and Fig.13 show the comparison of our method with others. Fig.1 shows that, ostensibly, other methods can reduce the distortion of the density distribution and preserve the overall shape to some extent. However, in fact, their distortions still exist, hiding deceptively, and even new distortions arise. Specifically, neither the sampling nor qualitative evaluation.

The collected datasets, the implementation of our method and the five metrics, the detailed scores, and the scatterplots created by all algorithms are all available in GitHub\(^3\). The latter two are also presented in Supplementary Material 2. In addition, we implemented a demo\(^4\) to interact with the tool \( f_{\text{draw}} \) and visually inspect the created scatterplots.

Time Complexity The time complexity of DistributionTranscriptor is \( O(N^2) \), where \( N \) is the number of nodes to be packed. For PolarPackaging, the time complexity of sorting nodes is \( O(N^2\log N) \). However, to find the packing position of a given node, \( O(\sqrt{N}) \) time is spent to determine a truncated subChain, \( O(1) \) time to search the final position, and \( O(1) \) time to update the frontChain. Hence, the overall time complexity of PolarPackaging is \( O(N^2/\sqrt{N}) \). Additional details can be found in Supplementary Material 1.

Impact of Parameters To investigate the impact of parameters \( k \), size, and \( N \) (number of points, equivalent to sampling rate) on the time cost of PolarPackaging in practice, we first studied the relationship between \( N \) and time cost, and then the relationship between the parameters and \( N \). We built a total of 28 simulated datasets whose volume ranges from 5k to 5M. The coordinates and radius of each node are randomly sampled from a unit circle and an interval between 1 and 10, respectively. Fig.10 shows the \( N \)-Time cost curve. The figure reveals that PolarPackaging algorithm is fairly fast, only taking 10s to pack 1 million nodes. Then we selected five representative datasets in volume from the collected datasets, and plotted their k/size/sampling_rate-\( N \) curves (Fig.10(①②③), respectively. As the interested parameter changes, others are fixed to their defaults (3, 5, and 1 for \( k \), size, and sampling rate, respectively). As expected, \( N \) increases as \( k \) and sampling rate rise and size decreases. The parameter size shows a quadratic-like impact which is considerably larger than the linear-like impact of \( k \) and sampling rate.

In addition to time cost, we also investigate the impact of the parameters on the five metrics in our measurement framework. Fig.11 presents the results. The range of size, \( k \), and sampling rate is set to [1, 10], [1, 20], and [0, 1], respectively. By mapping these ranges linearly to the same length, the impact curves of all parameters corresponding to the same metric can be aligned in one subfigure. For each curve, the value of the corresponding metric increases along the X-axis, while the others are fixed at their defaults. Noticing the small fluctuation ranges of all metrics on the Y-axis and recalling their intrinsic ranges, we declare that the impact of the parameters on all metrics are controllable, general, and reassuring. In other words, our method is fairly robust. Furthermore, we notice that the parameter size has a larger impact than \( k \) and sampling rate, and all metrics get worse as it raises. This observation is reasonable because size, representing the size of grids, determines the global resolution of the captured structures hidden in the data distribution. A small size facilitates the detection and depiction of refined structures by our method. Accordingly, the parameter \( k \), representing the minimum number of nodes to be placed in a grid, acts as a local resolution and only shows a notable impact on Density Preservation and KNN Preservation that depict local features. Interestingly, similar to \( k \), sampling rate only affects Density Preservation and KNN Preservation; and the two metrics get worse as it raises.

\(^2\)https://observablehq.com/@fil/experimental-plot-beeswarm

\(^3\)https://github.com/diyike/scatterplotUnfold

\(^4\)https://diyike.github.io/scatterplotUnfold
the transparency adjustment could substantially prevent overlap, and the color blending caused by the latter prevents inspecting details by zooming (Fig.1(c)). HaGrid and DGrid locally damage the shape and density preservation. Fig.1 offers the evidence, in which HaGrid and DGrid cannot properly handle regions with extremely high density (Fig.1(b)). The solid blocks cover up all details, including relative density and textures. Moreover, the sharp and straight boundaries are artifacts. We call this issue crowding. It is caused by the failure of the two subspace mapping methods in allocating adequate subspaces for high density regions. By contrast, our method performs well in density and shape preservation (Fig.1(c)) and supports seamless zoom to view more details, such as outliers and singularities.

5.3 Applications

Three applications demonstrate the capabilities of our method in pattern enhancement, interaction improvement, and expandability.

Pattern enhancement in semantic space Fig.14(a) shows a science map of computer science created by [28] using 86k scientific literatures. This map acts as the original scatterplot. Fig.14(b) reveals the new scatterplot created by our method. The differences between ① and ⑥ and ② and ④ show the power of our method to enhance the visual prominence of potential clusters. Fig.14(c) shows that our radius adjustment tool $f_{\text{draw}}$ can safely change the intensity and scope of the enhancement without overlaps. These enhanced clusters, like landmarks in cities, quickly attract the visual attention of analysts and elicit their interest, serving as navigators. In Fig.14(d), we color each literature by its leading topic which is determined by a probability topic model. The consistency of the spatial distribution between clusters and the transparency adjustment could substantially prevent overlap, and the color blending caused by the latter prevents inspecting details by zooming (Fig.1(c)). HaGrid and DGrid locally damage the shape and density preservation. Fig.1 offers the evidence, in which HaGrid and DGrid cannot properly handle regions with extremely high density (Fig.1(b)). The solid blocks cover up all details, including relative density and textures. Moreover, the sharp and straight boundaries are artifacts. We call this issue crowding. It is caused by the failure of the two subspace mapping methods in allocating adequate subspaces for high density regions. By contrast, our method performs well in density and shape preservation (Fig.1(c)) and supports seamless zoom to view more details, such as outliers and singularities.

5.3 Applications

Three applications demonstrate the capabilities of our method in pattern enhancement, interaction improvement, and expandability.

Pattern enhancement in semantic space Fig.14(a) shows a science map of computer science created by [28] using 86k scientific literatures. This map acts as the original scatterplot. Fig.14(b) reveals the new scatterplot created by our method. The differences between ① and ⑥ and ② and ④ show the power of our method to enhance the visual prominence of potential clusters. Fig.14(c) shows that our radius adjustment tool $f_{\text{draw}}$ can safely change the intensity and scope of the enhancement without overlaps. These enhanced clusters, like landmarks in cities, quickly attract the visual attention of analysts and elicit their interest, serving as navigators. In Fig.14(d), we color each literature by its leading topic which is determined by a probability topic model. The consistency of the spatial distribution between clusters and
Fig. 14: Application of our method to enhance potential patterns in semantic space. Our method (d) enhances the visual prominence of the potential clusters in the original scatterplot (a) and (c). The enhanced clusters and the topics (encoded by colors) uncovered by a topic model complement each other, helping analysts gain a better understanding of the semantic space.

topics shown in Fig.14(c) and Fig.14(f) proves that these clusters can uncover potential topics hidden in the semantic space. More importantly, the semantics provided by the topic model and the spatial structure revealed by the clusters complement each other, jointly promoting the understanding of the semantic space. The reason is two-fold. (1) Clusters remedy the inadequate resolution of topics. As shown in the red box in Fig.14(f), the distinct sub-clusters indicate that the red topic can be further divided into sub-topics. (2) Topics help verify whether the clusters have specific, coherent, and understandable semantics. As shown in the green box in Fig.14(f), the chaotic distribution of topics implies vague semantics of the focused clusters. We emphasize that all the aforementioned benefits arise from the capability of our method to transfer the correct density distribution from data space to visual space.

Interaction improvement in semantic space In some scenarios, without overlaps, our visualization easily reveals the semantic structures (Fig.15(d)) and enables the analyst to grasp the insights hidden in the data quickly by free zoom and pan.

**Overdraw mitigation of trajectory visualization** Theoretically, our method can be extended to solve the overdraw of any 2D visualization that can be represented by 2D nodes, such as large-scale time series curves [36], parallel coordinate axis [41], trajectories [54], and scalar fields [42]. Here, we take the trajectory visualization as an example. Fig.16(a) presents an original visualization of vehicle trajectories near a four-lane intersection. The data is taken from the CVPR trajectory clustering dataset [37]. We sampled massive data points at equal intervals along each trajectory to form the input to our method. Fig.16(b) and Fig.16(c) show that both transparency adjustment and our method can reveal regions under great traffic pressure. As shown in Fig.16(d), two yellow anomalous trajectories on the right tend to be drowned in massive regular brown trajectories due to the overlap. Unfortunately, reducing transparency does not help, but only increases the likelihood of missing the two brown anomalous trajectories on the left (Fig.16(d)). By contrast, our method retains both kinds of anomaly (Fig.16(f)). Though transforming continuous trajectories into discrete points dramatically reduces informative colored pixels, leading to less “contrast” of our visualization, the potential of our method in mitigating overdraw of other data types has been successfully demonstrated.

6 CONCLUSION AND FUTURE WORK

In this paper, we contribute a dual space coupling model to represent the complex relationship within and between data space and visual space analytically to solve the scatterplot overdraw problem. The proposed model introduces a new design space for promising overlap removal algorithm and interaction paradigm. We also develop an overlap-free scatterplot visualization method on the basis of the model, which shows competitive advantages compared with the state-of-the-art methods.

The algorithms described in this paper are not perfect. The hard partition of space caused by gridding may result in observable regular boundaries, especially when the parameter size is large. A promising solution of this problem is to replace the regular grids with a semantic partition that follows the distribution features, such as superpixels. We leave this idea for future work. Another interesting idea is to extend our algorithms to solve the scalability issues of 3D visualization.

ACKNOWLEDGMENTS

The authors want to thank anonymous reviewers. This work was supported in part by a grant from National Natural Science Foundation of China (# 62172295).
[47] H. Strobelt, M. Spicker, A. Stoffel, D. Keim, and O. Deussen. Rolled-out wordles: A heuristic method for overlap removal of 2d data representatives. In Computer Graphics Forum, vol. 31, pp. 1135–1144. Wiley Online Library, 2012.

[48] M. Theus and S. Urbanek. Interactive graphics for data analysis: principles and examples. CRC Press, 2008.

[49] W. Wang, H. Wang, G. Dai, and H. Wang. Visualization of large hierarchical data by circle packing. In Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 517–520, 2006.

[50] L.-Y. Wei. Multi-class blue noise sampling. ACM Transactions on Graphics (TOG), 29(4):1–8, 2010.

[51] L. Wilkinson. The grammar of graphics. In Handbook of computational statistics, pp. 375–414. Springer, 2012.

[52] 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), pp. 57–68. IEEE, 2019.

[53] 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, 2020.

[54] H. Zhou, P. Xu, X. Yuan, and H. Qu. Edge bundling in information visualization. Tsinghua Science and Technology, 18(2):145–156, 2013.