Cluster-Aware Grid Layout

Yuxing Zhou, Weikai Yang, Jiashu Chen, Changjian Chen, Zhiyang Shen, Xiaonan Luo, Lingyun Yu, and Shixia Liu

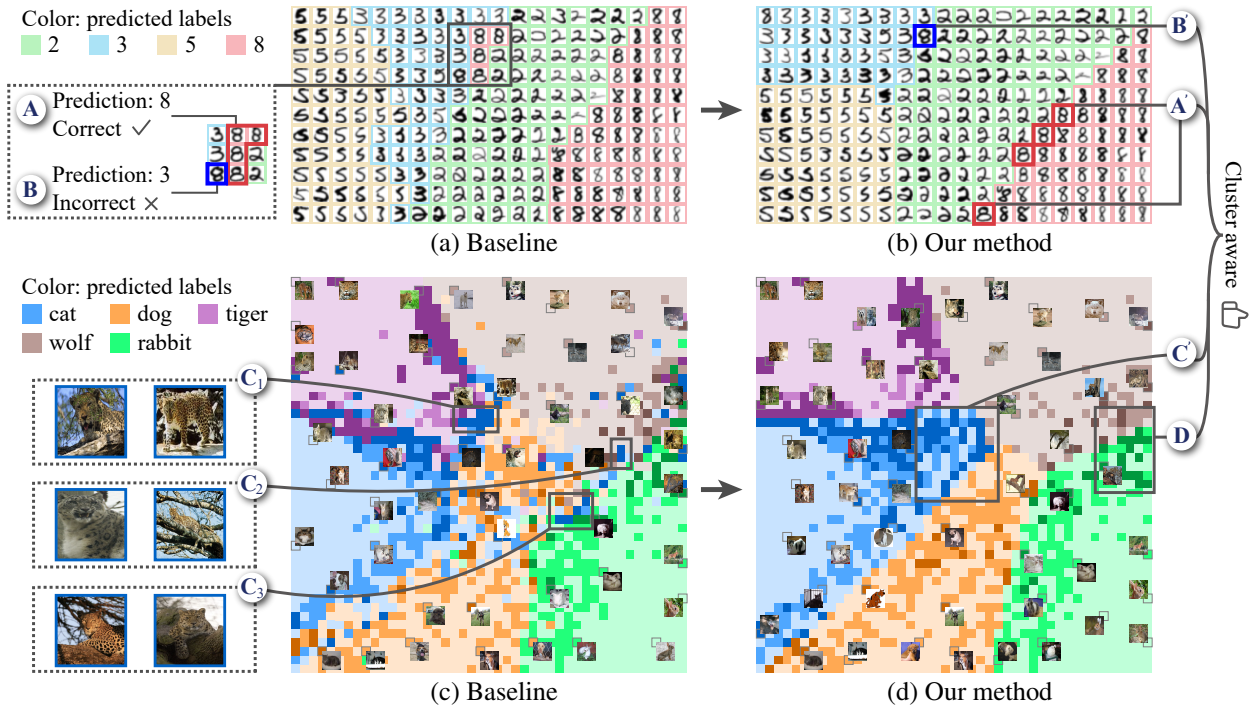


Fig. 1: Top: compared with the baseline (a) that only optimizes proximity, our method (b) places the samples that are predicted as "8" (A) in the cluster of "8." Bottom: compared with the baseline (c), our method (d) groups the dark blue cells C_1-C_3 in region C'.

Abstract— Grid visualizations are widely used in many applications to visually explain a set of data and their proximity relationships. However, existing layout methods face difficulties when dealing with the inherent cluster structures within the data. To address this issue, we propose a cluster-aware grid layout method that aims to better preserve cluster structures by simultaneously considering proximity, compactness, and convexity in the optimization process. Our method utilizes a hybrid optimization strategy that consists of two phases. The global phase aims to balance proximity and compactness within each cluster, while the local phase ensures the convexity of cluster shapes. We evaluate the proposed grid layout method through a series of quantitative experiments and two use cases, demonstrating its effectiveness in preserving cluster structures and facilitating analysis tasks.

Index Terms— Grid layout, similarity, convexity, compactness, optimization

1 Introduction

Grid visualizations are widely used to visually analyze data collections due to their high space efficiency [16]. Over two hundred CVPR 2022 papers utilize grid-like visualizations to compare and analyze model outputs. With these visualizations, computer vision researchers hope to perceive the samples in one cluster (*e.g.*, the samples with the same predictions) as a whole, which makes it easier to diagnose potential causes of low-performance models. If such cluster structures are not well perceived, accurate analysis and diagnosis will be hindered. For example, in the baseline method that only considers proximity (Fig. 1(a)), the samples of "8" in A and B are placed far away from the cluster of "8." This arrangement may lead users to draw the wrong conclusion that the model predicts these samples to be closely related to "3." However, the samples in A are similar to other samples of "8" and are predicted as "8." By preserving cluster structures, these samples are merged into the cluster of "8," which reduces false inferences. In addition, the sample that is misclassified as "3" (B) remains in the cluster of "3" (Fig. 1B′), which can be identified.

Several grid layout methods have been developed to improve readability by preserving the proximity relationships between data samples [17, 40, 50]. Despite their benefits, these methods struggle to maintain cluster structures within the data. For example, the samples in Fig. 1A are placed far away from their corresponding cluster. According to the Gestalt principles of perceptual grouping, preserving cluster structures requires not only the preservation of *proximity* relationships but also the *compactness* and *convexity* of each cluster shape [24,45,51].

- *Y. Zhou, W. Yang, J. Chen, Z. Shen, and S. Liu are with the School of Software, BNRist, Tsinghua University. Y. Zhou and W. Yang are joint first authors. S. Liu is the corresponding author. E-mail: {{yx-zhou19, yangwk21, cjs22, shenzhiy21}@mails., shixia@}tsinghua.edu.cn.*
- *C. Chen is with Kuaishou Technology. E-mail: chenchangjian@kuaishou.com*
- *X. Luo is with Guilin University of Electronic Technology. E-mail: luoxn@guet.edu.cn.*
- *L. Yu is with Xi'an Jiaotong-Liverpool University. E-mail: Lingyun.Yu@xjtlu.edu.cn.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

To develop a layout method that considers all three principles simultaneously, it is crucial to quantify them. Proximity is usually measured by the similarity preservation between samples and has been well studied by existing grid layout methods. The compactness is usually measured by the deviation of the grid positions from their corresponding cluster centers [45]. However, there is currently no widely accepted measure for quantifying shape convexity that aligns well with people's perception. To address this issue, we conducted a user study with 54 participants to evaluate which convexity measures are more consistent with people's perception. We found that although no single measure matched all participants' perception, two representative measures were preferred by two distinct groups of participants. However, these two measures conflict with each other to some extent. This requires our method to support different convexity measures to meet diverse user preferences.

After quantifying proximity, compactness, and convexity, we develop a cluster-aware grid layout method that balances the three measures. However, achieving this balance is challenging, especially when attempting to consider all three measures simultaneously during the layout process. Upon analyzing these measures, we discovered that proximity and compactness are affected by all grid cells, while convexity is sensitive to boundary cells between different clusters. Based on this finding, our layout method employs a global–local strategy to simplify the optimization process. Accordingly, the layout method consists of two phases: global assignment and local adjustment. The global assignment phase aims to generate a layout that balances proximity and compactness. This is formulated as a multi-task linear assignment problem and solved by an accelerated Jonker-Volgenant algorithm [9]. The local adjustment phase attempts to swap boundary cells between different clusters to improve convexity without apparently compromising the proximity and compactness achieved in the global assignment. Quantitative experiments demonstrate that our layout method achieves experimentally optimal balances among proximity, compactness, and convexity. We also present two use cases to exemplify the usage of our method.

The main contributions of our work include:

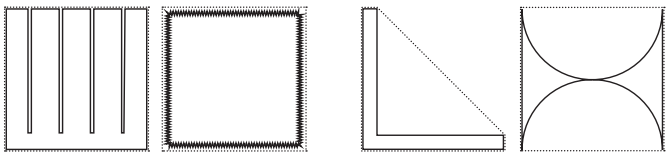
- study results on which convexity measures are more consistent with human perception.
- a grid layout method that achieves experimentally optimal balance among proximity, compactness, and convexity.
- an open-source implementation of the proposed grid layout method that enables easy plug-in of different convexity measures, which is available at https://github.com/thu-vis/Cluster-Aware-Grid-Layout.

2 Related Work

2.1 Convexity Measures

Mathematically, a shape is convex if it completely contains the line segment connecting any two points within the shape [44]. Based on this definition, researchers have developed various convexity measures, which can be classified into two categories [42]: area-based measures and boundary-based measures.

Area-based measures rely on the area of the shape to determine their scores. A common measure for evaluating the convexity of a shape is the area ratio, which computes the ratio of its actual area to that of its convex hull [13, 49]. This measure was extended by using the largest convex polygon contained in the shape (convex skull) [54] or considering the ratio between the area of the convex skull and the area of the convex hull [5]. However, these measures are sensitive to long and thin protrusions or intrusions because such protrusions/intrusions will largely affect the shape of its convex hull/skull. To address this issue, Rosin and Mumford [43] improved this measure by calculating the discrepancy between the area of the shape and its maximally overlapping convex shape. In addition to considering the area discrepancy, researchers have proposed several measures based on probability. For example, Held and Abe [19] estimated the degree of convexity by computing the probability that the shape contains line segments connecting two randomly sampled points inside the shape. Rahtu *et al.* [42] proposed a faster computation method by verifying if the shape contains a specific point on the segment (e.g., the midpoint) instead of examining the entire segment. Recently, Žunić and Rosin [61] proposed a parametric measure based on the similarity between the shape and its convex hull. A larger parameter of the measure leads to a stronger penalty for the area discrepancy near the boundary.



(a) 0.95, 0.50 (b) 0.96, 0.31 (c) 0.32, 0.87 (d) 0.21, 0.78

Fig. 2: The comparison between area ratios (the first value) and perimeter ratios (the second value). In (a) and (b), the boundary length changes considerably while the area changes slightly. The low perimeter ratios indicate that the perimeter ratio is more sensitive to changes in the boundary length. In contrast, (c) and (d) show considerable changes in their area and slight changes in boundary length. The low area ratios indicate that the area ratio is more sensitive to changes in the area.

Boundary-based measures evaluate the convexity of a shape by analyzing the geometric properties of its boundary, such as perimeter and tangents. A basic measure is the perimeter ratio, which computes the ratio of the L^2 perimeter of the convex hull to that of the shape [39]. Žunić and Rosin [54] further proposed to use the minimum ratio of the L^2 perimeter of its bounding rectangles to the L^1 perimeter of the shape over all the rotations, which is more sensitive to changes in the boundary of a concave region. Another boundary-based measure is proposed based on the fact that a convex shape always lies entirely on one side of its tangent [12]. The convexity measure is then measured by the average ratio over all the dominant parts cut by the tangents.

Generally, area-based measures are sensitive to changes in the area of a shape, while boundary-based measures are sensitive to changes in the boundary of a shape. Fig. 2 compares the area ratio (area-based) and perimeter ratio (boundary-based) in four examples. The perimeter ratios of the two examples with considerable changes in their boundary length ((a) and (b)) are much lower. While the area ratios of the two examples with considerable changes in their area ((c) and (d)) are much lower. The selection of convexity measures will depend on the specific analysis tasks. If users want to detect large concavities in the area, they usually choose area-based measures, while boundary-based measures are preferred for detecting irregular boundaries. To determine which measures are better aligned with human perception in a grid visualization, we conducted a user study to identify the most appropriate measures. These measures are given priority in our layout method.

2.2 Grid Visualizations

Initial efforts on grid visualizations randomly assign data samples to the grid cells [34]. Despite its simplicity, this method has proven useful for visually analyzing various data, including images [8, 21, 29, 37, 47], textual data [15, 46], video data [6, 25], relational data [33, 36, 59], geometric data [10, 34], and geospatial data [57, 60]. Subsequently, many grid layout methods have been developed to facilitate the analysis of similar samples by preserving pairwise distances between them. These methods fall into two categories [18]: direct mapping methods and projection-based methods.

The methods in the first category directly map high-dimensional samples onto a two-dimensional grid [2, 53]. Quadrianto *et al.* [40] proposed a method to maximize the correlation between the pairwise distances in the high-dimensional space and the pairwise distances in the grid layout. Another method, the self-sorting map [50], randomly assigns samples to grid cells and then iteratively swaps them to improve the similarity between neighboring samples. Barthel and Hezel [1] further improved the proximity preservation in the self-sorting map by utilizing an adaptive method to calculate the neighborhood representation of each sample. However, these two methods use a brute-force search to find the best swap, which is time-consuming. To boost efficiency, Barthel *et al.* [2] identified the best swap by solving a small-scale linear assignment problem locally. Other methods aim to optimize additional measures, such as compactness [35, 45] and aesthetic cri-

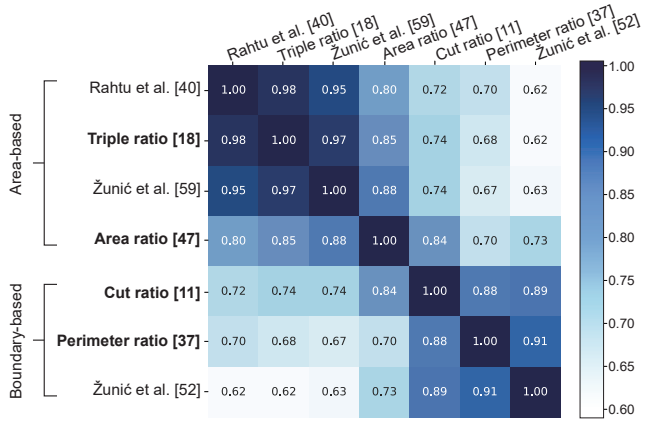


Fig. 3: Pearson correlations between seven convexity measures. The block-diagonal pattern highlights the presence of two different clusters.

teria [38, 48, 58]. To handle a large number of samples, Frey [16] generated a hierarchical grid layout by simultaneously optimizing the similarity between neighbors and the homogeneity within each node in the hierarchy. There are some treemap-based methods that visualize samples in a hierarchical grid format [3, 4]. However, these methods do not consider the proximity between samples within each cluster.

Projection-based methods first utilize a dimensional reduction technique to project the samples onto a 2D space without the grid constraints. Then the final layout is generated by moving samples from the projected positions to the grid cells [14, 17, 18, 20]. For example, IsoMatch [17] uses IsoMap to project images onto a 2D space. After building the bipartite graph between the projected positions and grid cells, the grid layout is obtained by finding the bipartite matching that minimizes the total distance moved. Since it is time-consuming to solve the bipartite matching problem, later studies attempt to accelerate the process of assigning projected samples into cells. Chen *et al.* [9] developed a *k*NN-based bipartite graph matching, which speeds up the algorithm by reducing the number of candidate grid cells for each sample. Additionally, a grid layout can be generated by removing the overlap between projected samples and then aligning them. DGrid [20] recursively bisects the projected samples until each partition contains exactly one sample. The partition result is then aligned with a regular grid layout. CorrelatedMultiples [31] uses a force-directed graph layout method to remove the overlap between samples and then aligns the samples horizontally and vertically to form a grid layout. This technique is also used to lay out a clustered graph in a grid format [23]. However, when the samples are unevenly projected on a 2D space, these alignment techniques can result in large movements from projected positions to grid cells. To address this issue, VRGids [18] employs the Voronoi relaxation to scatter the projected samples evenly on a bounded 2D space before assigning them to cells, which reduces the total movements.

Although the aforementioned methods have been proven effective in preserving the proximity relationships between data samples, they do not explicitly consider the cluster structures within the data. Consequently, they face difficulties when tasked with preserving such structures. Our layout method starts from a grid layout generated by any of these methods and then enhances the compactness and convexity of each cluster shape to preserve cluster structures. When handling a large number of samples, a flat grid layout encounters the scalability issue in displaying all the samples clearly in one view. Our method can either utilize the existing hierarchy in the input layout or build the hierarchy using sampling techniques [9]. At each level of the hierarchy, our method enhances compactness and convexity to preserve the cluster structures.

3 User Study on Convexity

The user study has two goals: first, to determine which measures closely align with human perception, and second, to explore whether variations in grid size and cluster number influence people's perception of convexity. To achieve these goals, three hypotheses have been formulated to guide the design of our user study:

H1: There exists a convexity measure that aligns with the perception of most people.

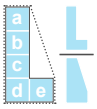
H2: The grid size influences people's perception of convexity.

H3: The cluster number influences people's perception of convexity.

3.1 User Study Design

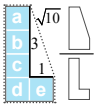
Selection of convexity measures. For the 10 convexity measures introduced in Sec. 2.1, we excluded 3 measures due to their high time complexity: two of them require computing convex skulls with a time complexity of $O(N^7)$ [5, 54], and one requires finding maximally overlapping convex shapes with a time complexity of $O(2^N)$ [43] (N is the number of vertices). Fig. 3 shows the Pearson correlations between the implemented convexity measures calculated on 9,689 different shapes. Further details of this experiment are summarized in supplemental material. The results in Fig. 3 indicate the seven measures are roughly classified into two clusters that correspond to the area-based measures and boundary-based measures, respectively. The correlations between four area-based measures are strong, and so are the correlations between three boundary-based measures. Due to the high correlations, we selected the representative measures rather than using all of them in the user study, thus making it easier for participants to compare multiple grid visualizations optimized for different measures.

Among the four area-based measures, *area ratio* [49] is selected first because it has a relatively weak correlation ($r < 0.9$) with the other three measures and cannot be represented by them. Of the remaining three highly-correlated area-based measures, we selected *triple ratio* [19] since it has stronger correlations with the other two measures (0.98 and 0.97). In addition, *triple ratio* is more efficient to be computed than Žunić's method [61], and also provides more precise measurements than Rahtu's method [42]. Among the three boundary-based measures, *perimeter ratio* [39] and Žunić's method [54] are both based on the perimeter and are highly correlated (0.91). We selected *perimeter ratio* because its calculation is more efficient ($O(N \log N)$) than the latter ($O(N^2)$). We also chose to include *cut ratio* [12] because it has a relatively weaker correlation ($r < 0.9$) with the other two measures. The definitions of the four selected measures are provided below:

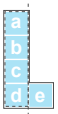


Area ratio (A) is defined as the ratio of the area of shape to the area of its convex hull. In this example, the length of each side of the squares is 1. Thus, the area of the shape is 5, and the area of the convex hull is $5+1.5=6.5$. The convexity score is calculated as $5/6.5 \approx 0.769$.

Triple ratio (T) is defined as the probability that, for a collinear triple (X,Y,Z), if both cells X and Z are inside the shape, then cell Y is also inside the shape. In this example, there are 8 collinear triples of which both endpoints are located inside the shape (*e.g.*, (a,b,c) and (a,b,e)). However, the interior cell of 2 triples ((a,f,e) and (b,f,e)) lies outside the shape. Thus, the convexity score is calculated as $(8-2)/8=0.75$.



Perimeter ratio (P) is defined as the ratio of the perimeter of the convex hull to the perimeter of the shape. In this example, the perimeter of the convex hull is $8+\sqrt{10}$, and the perimeter of the shape is $8+4=12$. The convexity score is calculated as $(8+\sqrt{10})/12 \approx 0.857$.



Cut ratio (C) is proposed based on the property that for a convex shape, all tangents to its boundary will not intersect its interior. It calculates, for each edge in the boundary, the ratio of the part cut by the tangent. The convexity measure is then defined as the average ratio over all the edges. In this example, the convexity score is calculated as $(1\times 8+0.8\times 3+0.4\times 1)/12=0.9$.

Participants. We recruited 54 participants (44 male and 10 female) in the experiment, including faculty members, researchers, developers, and graduate students with industry/research experience in visualization, computer graphics, computer vision, or mathematics. They have used grid layouts to explore data in their research/development. The

participants come from 3 countries and 9 institutions. The diversity in expertise, experience, and affiliations ensures that our recruited participants are representative. Among the 54 participants, all have knowledge of convexity, none reported color blindness or color weakness, and 41 of the participants are very familiar with grid visualizations. Upon completion, each participant was rewarded a $20 gift card.

Study procedure. At the beginning of the study, participants were presented with a tutorial video that introduced the definition of convex polygons and the user interface of the study system. After watching the video, the participants began a practice session with six trials. The answers and corresponding explanations were displayed after participants submitted their results. After completing the practice session and indicating their full understanding of the concept of convex polygons and the study interface, the participants proceeded to the formal study with 36 trials, in which answers were no longer displayed. Participants were instructed that they could take a brief break after completing every nine trials. Following the completion of all trials, they were asked to fill out a questionnaire that included personal information and a question asking them to explain how they compare the convexity of different grid visualizations. The entire process took approximately 40 minutes. The study received approval from the University Ethics Committee.

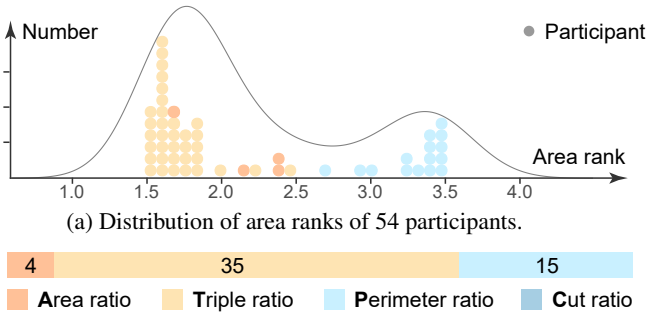
Trials and stimuli. To validate **H1**, in each formal trial, the participants were asked to rank four different grid visualizations, each optimized for one of the four selected convexity measures. This enabled the ranking results to reflect their preference for the convexity measures. Before the formal study, we provided six practice trials to ensure that the participants correctly understood the concept of convex polygons. The stimuli in each practice trial consisted of four grid visualizations arranged in descending order on all four convexity measures.

Conditions and design. To validate **H2** and **H3**, we manipulated two variables, the *grid size* and the *cluster number*, to control for their effects. In practical applications, the maximum grid size is typically restricted to 40x40 to ensure that each cell has enough space to display data clearly. Therefore, we chose three different grid sizes: 20x20, 30x30, and 40x40 in the experiment. We selected three cluster numbers of 3, 5, and 10 because analyzing a large number of clusters simultaneously can be challenging to the participants. To generate trials for each condition, we used four datasets, Animal [11], MNIST [28], CIFAR10 [27], and USPS [22], each of which contains 10 clusters. A full-factorial within-subjects design was used to evaluate the effects of the grid size and cluster number. As a result, for each participant, a total of 36 trials (3 grid sizes $\times$ 3 cluster numbers $\times$ 4 datasets) were evaluated in the formal study. The orders for the grid sizes and cluster numbers were counterbalanced using a Latin Square Design.

3.2 Result Analysis

H1: There exists a convexity measure that aligns with the perception of most people (partially confirmed).

We analyzed the ranking results to determine if there is a specific convexity measure that was preferred by the majority of the participants. We first processed the ranking result of each participant. If a participant chooses $=$ between two measures, their ranks will be the average of the ranks. For example, if a participant ranks the measures as $\mathbf{A} > \mathbf{T} > \mathbf{P} = \mathbf{C}$, their ranks will be 1, 2, 3.5, and 3.5. The ranking result of each participant is computed by averaging his/her ranks over all the trials. Next, we conducted Friedman tests to compare the ranks of different measures on all the participants. However, the tests showed no significant differences between the measures. Upon further examination of the ranking results, we observed that there was a large variance in the rankings for **T** and **P**: some participants ranked **T** highest and ranked **P** lowest, while others ranked **P** highest and ranked **T** lowest. This diversity led to the large variances and made the differences not significant. In addition, we found a strong correlation between two boundary-based measures **P** and **C** (0.955). This indicates that participants who preferred the boundary-based measure **P** also tended to prefer the boundary-based measure **C**. Similarly, a strong correlation was found between the two area-based measures **A** and **T** (0.926). We obtained the *boundary rank* for each participant by averaging the ranks of **P** and **C**, and the *area rank* by averaging the ranks of **A** and **T**. It is important to note that the sum of the boundary rank and area rank always adds up to five. Therefore, it is sufficient to analyze only one of them. Fig. 4(a) shows the distribution of area rank. There were two distinct groups of participants, where 39/54 (72.2%) of them preferred the area-based measures (area rank$<$ 2.5), while the remaining (15/54, 27.8%) preferred the boundary-based measures (area rank$>$ 2.5).



(a) Distribution of area ranks of 54 participants.

(b) Distribution of the most preferred measures of 54 participants.

Fig. 4: Examine the distribution of area ranks and most preferred measures of 54 participants. Among these participants, 39 preferred area-based measures (orange), while the other 15 preferred boundary-based measures (blue).

Next, we investigated the measures that were most preferred by each participant. Surprisingly, 35 of them ranked **T** highest, and 15 of them ranked **P** highest (Fig. 4(b)). The result indicates that most of the participants who preferred area-based measures ranked **T** higher than **A** (35/39, 89.7%), whereas all the participants who preferred boundary-based measures ranked **P** higher than **C** (15/15, 100.0%). We thus performed Friedman tests again to compare the ranks of different measures for the two groups of participants separately. The results showed that for participants who preferred area-based measures (Fig. 5 Left), the difference among the four measures was significant ($\chi^2(3) = 98.64,\ p < 0.001$). The corresponding effect size was 0.8612, which also indicated a great difference between different measures. The pairwise Wilcoxon signed-rank test results further indicated a strict preference order of $\mathbf{T} > \mathbf{A} > \mathbf{C} > \mathbf{P}$. For participants who preferred boundary-based measures (Fig. 5, Right), the difference among the four measures was also significant ($\chi^2(3) = 43.88,\ p < 0.001$), and the pairwise Wilcoxon test results showed a strict preference order of $\mathbf{P} > \mathbf{C} > \mathbf{A} > \mathbf{T}$. The corresponding effect size was 0.9751, which again indicated a great difference between different measures. It is important to note that this order is the exact reverse of the order found in participants who preferred area-based measures. Based on these findings, we concluded that no single measure aligns perfectly with

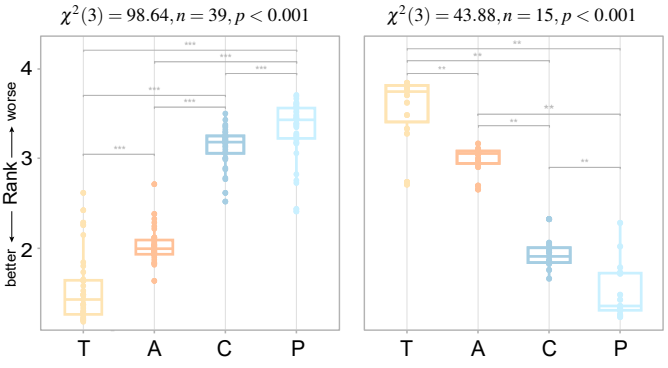
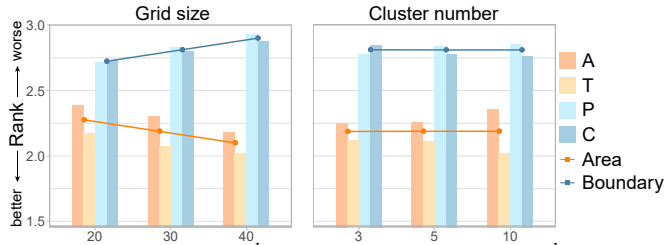


Fig. 5: Friedman tests and pairwise Wilcoxon signed-rank tests on four measures. Left: the participants who preferred area-based measures have a strict preference order of $\mathbf{T} > \mathbf{A} > \mathbf{C} > \mathbf{P}$; Right: the participants who preferred boundary-based measures have a strict preference order of $\mathbf{P} > \mathbf{C} > \mathbf{A} > \mathbf{T}$. Significance levels are denoted by asterisks: * indicates $p < 0.05$, ** indicates $p < 0.01$, and *** indicates $p < 0.001$.



contrast	statistics	n	p-value
30 - 20	473.5	54	0.021 *
40 - 30	509.0	54	0.045 *
40 - 20	212.0	54	8.43e-4 ***

contrast	statistics	n	p-value
5 - 3	774.0	54	0.607
10 - 5	751.5	54	0.753
10 - 3	751.5	54	0.572

Fig. 6: Compare the ranks of convexity measures for different grid sizes and different cluster numbers. The grid size influences people's perception of convexity, while the cluster number does not. The tables present the results of the pairwise Wilcoxon signed-rank test on the area rank. Significance levels are denoted by asterisks: * indicates $p < 0.05$,** indicates $p < 0.01$, and *** indicates $p < 0.001$.

the perception of all participants. However, measure **T** and measure **P** are the representative of area-based measures and boundary-based measures, respectively. For those who preferred area-based measures, measure **T** aligns best with their perception, and for those who preferred boundary-based measures, measure **P** aligns best with their perception. Therefore, **H1** is partially confirmed.

H2: The grid size influences people's perception of convexity (confirmed). Initially, we considered conducting a two-way ANOVA test to analyze the effects of the grid size and cluster number, given that there are two independent variables. The interaction effect between the two variables was not significant in the ANOVA test, suggesting that their effects can be analyzed separately. Additionally, our data violated the normality assumption. To analyze these two variables separately, we utilized the nonparametric Friedman test and pairwise Wilcoxon signed-rank tests, which do not depend on the normality assumption.

We first analyzed the effect of the grid size on the ranks of two representative measures, measure **T** and measure **P**, which respectively represent area-based and boundary-based measures. However, we did not observe any significant effect. After conducting a more thorough analysis, we noticed that participants who strongly preferred measure **T** and **P** consistently ranked these measures highest, and that varying the grid size had little effect on their ranking results. Therefore, we shifted our focus to analyzing the area rank (**A**+**T**)/2 rather than analyzing solely the rank of measure **T**, which was a more robust approach to the analysis. The effect of the grid size on the area rank becomes significant ($\chi^2(2) = 14.91, p < 0.001$). In addition, we observed that the area rank consistently decreased as the grid size increased, as illustrated in Fig. 6 left. The pairwise Wilcoxon test results further confirmed significant differences between the area rank for different grid sizes. Therefore, we concluded that as the grid size increases, the area-based measure aligns better with people's perception. **H2** is confirmed.

H3: The cluster number influences people's perception of convexity (rejected). Similarly, we conducted the Friedman test and pairwise Wilcoxon signed-rank tests to investigate the effect of the cluster number on the rank of measure **T**, the rank of measure **P**, and the area rank. There was no significant effect of the cluster number on any of these ranks. Additionally, the pairwise Wilcoxon test results showed no significant differences in area ranks between different grid sizes (Fig. 6 right). Based on these findings, we concluded that the cluster number does not influence the perception of convexity. **H3** is rejected.

3.3 Participant Feedback

We analyzed participants' feedback on how they compared the convexity of different grid visualizations, and we also conducted interviews with ten participants to gather further insight into their judgment-making process. Among the ten participants, seven preferred area-based measures, while the remaining three preferred boundary-based measures. Participants who favored area-based measures tended to overlook small zig-zags along a boundary and instead viewed the boundary as a line segment. One faculty member commented that the anti-aliasing effect in the human visual system could explain this judgment. According to this theory, these zig-zags become less noticeable with increasing grid sizes, which is consistent with our user study findings. Two faculty members with a research interest in computer graphics mentioned that they tended to compare the grid visualizations with the corresponding Voronoi diagrams when making judgments. They regarded a significant difference between the two as a sign of poor convexity. This explains why they do not like the layout optimized for boundary-based measures: a shape tends to become during the optimization of boundary-based measures, resulting in a larger difference compared to the corresponding Voronoi diagram. In contrast, the participants who preferred boundary-based measures usually paid more attention to the number of zig-zags on the boundary. They generally preferred a cluster shape with fewer edges along its boundary.

4 Cluster-Aware Grid Layout Method

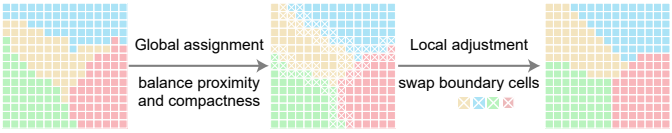


Fig. 7: Our layout pipeline: our method first balances proximity and compactness in the global assignment phase, and then improves the convexity by swapping the boundary cells in the local adjustment phase.

4.1 Design Principles

Our cluster-aware grid layout method aims to improve the analysis efficiency by enhancing the recognition of clusters [7,26]. This is well aligned with the Gestalt principles of perceptual grouping [32,45,51,55]. These principles investigate how certain elements tend to be perceived as a group. They can be classified into two categories: layout-irrelevant principles and layout-relevant principles. Layout-irrelevant principles describe how visual elements are grouped regardless of their spatial arrangement, which includes similarity, common fate, connectedness, and common region. In contrast, layout-relevant principles depend on the spatial relationships between elements, which include proximity, compactness, symmetry, and convexity. Our layout method is based on layout-relevant principles. Since the convexity principle overrules the symmetry principle [24], we employ the following three principles to develop a cluster-aware grid layout method:

Proximity: Similar samples should be placed close to each other.
Compactness: Samples in the same cluster should be placed in a compact form.
Convexity: Samples in the same cluster should form a convex shape.

4.2 Measuring Cluster Preservation

The key in developing our layout method is to quantify the three measures that correspond to the three selected Gestalt principles.

Proximity. To preserve proximity in a grid layout, the samples with higher similarity should have smaller Euclidean distances between their corresponding cell positions. Let $S = \{s_1, s_2, \ldots, s_n\}$ denote the samples and $\{g(s_1), g(s_2), \ldots, g(s_n)\}$ denote their corresponding cell positions. Given two samples s_i and s_j, their similarity is denoted as $c_{ij} \in [0,1]$, and their Euclidean distance on the grid is computed by $\|g(s_i) - g(s_j)\|$, where $\|\cdot\|$ is the Euclidean norm. The proximity of a grid layout is then determined by:

$$\text{Prox} = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(w\|g(s_i) - g(s_j)\| - (1 - c_{ij}) \right)^2, \qquad (1)$$

where w is a scaling factor to ensure that the first term ranges from 0 to 1. A smaller proximity value indicates better preservation of proximity.

Compactness. Proximity does not consider the cluster information and cannot guarantee the samples within the same cluster form a compact shape. To preserve compactness, samples should be placed closer to their corresponding cluster centers. Following Rottmann's

method [45], the compactness of the grid layout is measured by the total distance between the cell position of each sample and its corresponding cluster center:

$$\text{Comp} = \sum_{i=1}^{n} \|g(s_i) - \mu_i\|^2, \quad (2)$$

where μ_i is the corresponding cluster center of sample s_i. It is computed as the average cell position over the samples in the same cluster as s_i. A smaller compactness value indicates a more compact layout.

Convexity. As described in Sec. 3.1, there are four representative convexity measures, *area ratio*, *triple ratio*, *perimeter ratio*, and *cut ratio*. The quantifying method of each measure for a polygon is also introduced in this section. Once a user selects one of the aforementioned measures, the convexity of a grid layout is calculated as the average convexity score over all the cluster shapes in the grid layout.

4.3 Layout Algorithm

Optimization strategy. Given the large search space spanning over proximity, compactness, and convexity, it is impractical to achieve a perfect balance among the three measures. Our analysis of the calculation of these three measures reveals that proximity and compactness are affected by all grid cells, whereas convexity is sensitive to the boundary cells between different clusters. Based on this finding, we employ a global-local strategy to simplify the optimization process. As shown in Fig. 7, the global assignment achieves a good balance between proximity and compactness. The local adjustment swaps the boundary cells between clusters to improve convexity without apparently affecting the achieved proximity and compactness. The effectiveness of the global-local strategy is demonstrated in the ablation study (Table 1).

Global assignment. The goal of this phase is to generate a global layout that simultaneously optimizes proximity and compactness. Instead of directly optimizing proximity defined in Eq. (1), we take a grid layout generated by a proximity-preserving method as input and then minimize the distance between the input layout and our layout. This offers the flexibility to integrate any existing proximity-preserving method into our layout pipeline. Let $V = \{v_1, v_2, \ldots, v_n\}$ denote the grid cells. Without loss of generality, we set v_i as the input cell position of s_i. Our layout can be viewed as a new assignment from the samples S to the grid cells V, which is denoted by a binary matrix $\boldsymbol{\delta} = \{\delta_{ij}\}_{1 \le i,j \le n}$. Here, $\delta_{ij} = 1$ indicates that s_i is assigned to v_j (*i.e.*, $g(s_i) = v_j$), and otherwise, $\delta_{ij} = 0$. The proximity of the layout is measured by its distance to the input layout:

$$\text{Prox}(\boldsymbol{\delta}) = \sum_{i=1}^{n} \|g(s_i) - v_i\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \|v_j - v_i\|^2 \delta_{ij}. \quad (3)$$

With the notation $\boldsymbol{\delta}$, the compactness can be rewritten as:

$$\text{Comp}(\boldsymbol{\delta}) = \sum_{i=1}^{n} \|g(s_i) - \mu_i\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \|v_j - \mu_i\|^2 \delta_{ij}. \quad (4)$$

To simultaneously optimize proximity and compactness, the measures defined in Eqs. (3) and (4) are combined, and the global layout is achieved by minimizing

$$\begin{aligned} \underset{\boldsymbol{\delta}}{\text{minimize}} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\lambda \|v_j - v_i\|^2 + (1-\lambda) \|v_j - \mu_i\|^2 \right) \delta_{ij}, \\ \text{subject to} \quad & \sum_{i=1}^{n} \delta_{ij} = 1, \ \forall j \in \{1, 2, \ldots, n\}, \\ & \sum_{j=1}^{n} \delta_{ij} = 1, \ \forall i \in \{1, 2, \ldots, n\}, \\ & \delta_{ij} \in \{0, 1\}, \ \forall i, j, \end{aligned} \quad (5)$$

where $0 \le \lambda \le 1$ is the weight to balance proximity and compactness. The constraints ensure a one-to-one assignment from samples to grid cells. The optimization problem defined in Eq. (5) is a linear assignment problem and can be efficiently solved with an accelerated Jonker-Volgenant algorithm [9].

Determining weight λ that balances proximity and compactness is crucial for generating a global layout that achieves good results. Using a fixed value, such as $\lambda = 0.5$, may not be optimal for all cases, and manually tuning the parameter is labor-intensive and requires expertise. Thus, we employ the multi-task learning method proposed by Liu *et al.* [30] to determine λ. The key idea is to increase the weight of the task with poor performance so that it can be further improved. To assess the performance of each task, we compare the current proximity/compactness score with its optimal one. Specifically, we first obtain the layout with the optimal proximity $\boldsymbol{\delta}_p$, which is the input layout, and the layout with the optimal compactness $\boldsymbol{\delta}_c$, which is computed by optimizing compactness solely. The performance of optimizing proximity is determined by $\Delta_{\text{Prox}} = (\text{Prox}(\boldsymbol{\delta}) - \text{Prox}(\boldsymbol{\delta}_p))/(\text{Prox}(\boldsymbol{\delta}_c) - \text{Prox}(\boldsymbol{\delta}_p))$, and the performance of optimizing compactness is determined by $\Delta_{\text{Comp}} = (\text{Comp}(\boldsymbol{\delta}) - \text{Comp}(\boldsymbol{\delta}_c))/(\text{Comp}(\boldsymbol{\delta}_p) - \text{Comp}(\boldsymbol{\delta}_c))$. Once the performance of both tasks is determined, we calculate λ as $\Delta_{\text{Prox}}/(\Delta_{\text{Prox}} + \Delta_{\text{Comp}})$ and update the layout with the new weight. The above procedure is repeated until the layout converges.

Local adjustment. After obtaining the global layout, the local adjustment improves convexity by swapping the boundary cells between different clusters. In our implementation, a cell is considered to be a boundary cell if at least one of its neighboring cells within a 3x3 region belongs to a different cluster. We only swap boundary cells since they directly impact convexity. At each iteration, a boundary cell is randomly selected, and all possible swaps between the selected cell and other boundary cells are enumerated. The convexity is evaluated after each swap, and the optimal swap that increases convexity most is chosen. The iterative process stops when all the boundary cells are processed.

Supporting hierarchical grid layout. The two-phase layout method creates a flat grid layout that simultaneously optimizes proximity, compactness, and convexity. When handling a large number of samples, a hierarchical grid layout is necessary to support level-of-detail exploration. If the input is a hierarchical layout, our method enhances compactness and convexity at each level of the hierarchy while preserving proximity. Otherwise, our method creates a hierarchy using the sampling-based technique described in Chen *et al.*'s work [9]. It first samples a set of representative samples and creates the grid layout at the top level. The remaining samples are assigned to their closest representative sample. When the user selects a sub-region for exploration, the selected samples and some of the samples assigned to them are used to generate a grid layout in the same way. For both methods, we try to preserve the relative positions of the previously displayed samples for maintaining the mental map during exploration.

5 Evaluation

5.1 Quantitative Evaluation

5.1.1 Datasets and Experimental Settings

Datasets. We evaluated the effectiveness of our layout method on 11 datasets. Ten of them (Animals, Cifar10, Indian Food, Isolet, MNIST, Stanford Dogs, Texture, USPS, Weather, Wifi) are from Xia *et al.*'s work [56], while an additional dataset, OoD-Animals, is from OoD-Analyzer for data quality analysis [9]. There are eight image datasets, two textual datasets (Isolet and Texture), and one tabular dataset (Wifi). More details of these datasets are in supplemental material. For images, we used CLIP [41], a state-of-the-art pre-trained model to extract the feature vectors. For textual data and tabular data, we used the feature vectors provided by the dataset. The cluster labels of each dataset were set as the predictions, which were obtained using the k-NN classifier, where k was determined using cross-validation.

Experimental settings. The baseline is the state-of-the-art proximity-preserving grid layout method proposed by Chen *et al.* [9]. The method first projects samples into a 2D space using t-SNE, and then assigns the projected samples to cells by solving a linear assignment problem. Ours-G improves compactness only through the **g**lobal assignment, while Ours-L(T) and Ours-L(P) only use **l**ocal adjustment to improve **t**riple ratio (area-based) or **p**erimeter ratio (boundary-based). We chose these two measures because they were representative of area-based measures and boundary-based measures, respectively. We also evaluated the performance of different combination orders of the global phase and local phase, which resulted in four more methods: Ours-L(T)-G, Ours-L(P)-G, Ours-G-L(T), and Ours-G-L(P). In the following experiments, we generate the grid layouts using 3 different grid sizes: 20x20, 30x30, and 40x40, consistent with the grid sizes used in our user study.

Table 1: Comparison of six measures of all the methods. Baseline: OoDAnalyzer [9]; G: global; L: local; T: triple ratio; P: perimeter ratio.

Measure	Baseline	Ours-G	Ours-L(T)	Ours-L(P)	Ours-L(T)-G	Ours-L(P)-G	Ours-G-L(T)	Ours-G-L(P)
Proximity	**1.000**	0.996	0.998	0.992	0.997	0.996	0.996	0.994
Compactness	0.964	**0.970**	0.967	0.963	0.969	**0.970**	**0.970**	0.968
Area ratio	0.669	0.882	0.900	0.852	0.869	0.883	**0.913**	0.895
Triple ratio	0.936	0.991	0.995	0.954	0.989	0.991	**0.997**	0.978
Perimeter ratio	0.812	0.834	0.857	0.926	0.833	0.835	0.866	**0.935**
Cut ratio	0.802	0.870	0.872	0.913	0.866	0.872	0.890	**0.934**

Table 2: Comparison of six measures of the baseline (OoDAnalyzer [9]), Ours-T, and Ours-P with 3 different grid sizes.

Grid size	Proximity			Compactness			Area ratio			Triple ratio			Perimeter ratio			Cut ratio		
	Basel.	Ours-T	Ours-P	Basel.	Ours-T	Ours-P	Basel.	Ours-T	Ours-P	Basel.	Ours-T	Ours-P	Basel.	Ours-T	Ours-P	Basel.	Ours-T	Ours-P
20x20	**1.000**	0.996	0.993	0.965	**0.970**	0.968	0.750	**0.898**	0.896	0.951	**0.995**	0.976	0.843	0.875	**0.938**	0.839	0.895	**0.935**
30x30	**1.000**	0.996	0.994	0.964	**0.970**	0.969	0.664	**0.916**	0.897	0.938	**0.997**	0.979	0.816	0.865	**0.935**	0.801	0.891	**0.935**
40x40	**1.000**	0.995	0.993	0.962	**0.970**	0.969	0.591	**0.926**	0.893	0.919	**0.998**	0.978	0.775	0.858	**0.933**	0.766	0.885	**0.932**

Evaluation criteria. We used six measures to evaluate the quality of the grid layout: proximity, compactness, and four convexity measures, including triple ratio, area ratio, perimeter ratio, and cut ratio. The scores of the four convexity measures range from 0 to 1, and a higher score indicates better convexity, whereas the proximity and compactness scores (Eqs. (3) and (4)) range from 0 to infinity, and a smaller score indicates better proximity/compactness. To facilitate comparison, we apply the transformation $x \mapsto \exp(-x)$ to the proximity and compactness scores, such that they are normalized to a range of 0 to 1, and a higher score indicates better proximity/compactness.

5.1.2 Ablation Results

Our study was designed to examine the impacts of both the global assignment phase and the local adjustment phase. The effectiveness was demonstrated by comparing the associated methods using the six measures. The results presented in Table 1 were averaged over the 11 datasets and 3 grid sizes, and full results are available in supplemental material. All our methods preserved proximity well, with scores above 0.99. Thus, our analysis mainly focused on compactness and convexity. Compared with the baseline, Ours-G improved compactness from 0.964 to 0.970. Moreover, it also showed improvement on all four convexity measures, with more notable improvement on the area-based ones (area ratio and triple ratio). This is because the optimization of compactness leads to regular cluster shapes, which have higher scores in area-based measures. Regarding convexity measures, Ours-L(T) performed better than baseline/Ours-G/Ours-L(P) in terms of the area ratio and triple ratio, while Ours-L(P) surpassed baseline/Ours-G/Ours-L(T) with respect to the perimeter ratio and cut ratio. This indicates that optimizing an area-based convexity measure can lead to a large improvement on the other area-based measures due to their related optimization goals. The same applies to boundary-based measures.

Moreover, our ablation study explored the optimal order to combine the global assignment phase and the local adjustment phase. It was observed that the results of Ours-G, Ours-L(T)-G, and Ours-L(P)-G were quite similar. This indicates that the changes made in the global assignment phase have a greater influence than those in the local adjustment phase. Therefore, the local adjustment should be conducted after the global assignment. The comparison between Ours-G-L(T)/Ours-G-L(P) and Ours-G reveals that the local adjustment phase further improves convexity without apparently affecting the proximity and compactness achieved in the global assignment phase. Moreover, Ours-G-L(T) performed better than Ours-L(T) in terms of the area ratio and triple ratio, and Ours-G-L(P) performed better than Ours-L(P) regarding the perimeter ratio and cut ratio. This indicates that the global assignment phase provides a better initial layout for the local adjustment phase, leading to a larger improvement on convexity measures. Therefore, we choose Ours-G-L(T) and Ours-G-L(P) as the primary methods, which are abbreviated as **Ours-T** and **Ours-P** in the following experiments.

5.1.3 Comparison Results

Table 2 provides a detailed comparison between the baseline and Ours-T/Ours-P with 3 different grid sizes. The reported results were averaged over 11 datasets. In terms of proximity and compactness, there were no significant differences across different grid sizes. Ours-T achieved the highest compactness without affecting proximity too much. Compared to Ours-T, Ours-P achieved relatively lower scores in both proximity and compactness, but the differences were very small. Regarding convexity measures, Ours-T performed best in terms of area-based measures, while Ours-P performed best in terms of boundary-based measures. This indicates that our methods can fulfill the requirements of individuals who prefer either area-based or boundary-based measures. When comparing the convexity scores across different grid sizes, it is notable that while the convexity scores achieved by the baseline method decrease as the grid size increases, the scores of area-based measures achieved by Ours-T increase with larger grid sizes. After analyzing the calculation of area-based measures, it is discovered that the small zig-zags on the boundaries between different clusters have less impact

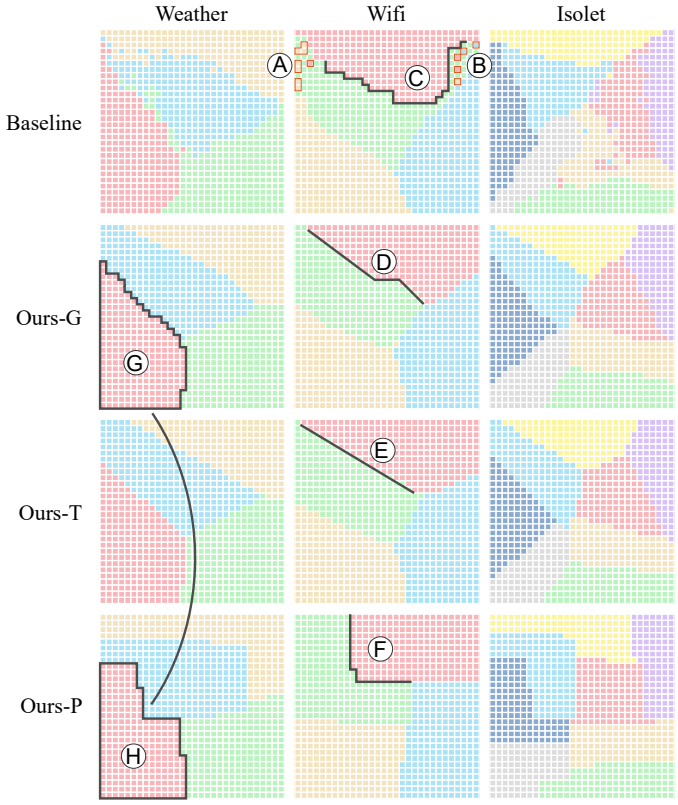


Fig. 8: Comparison of the layouts generated by the baseline (OoDAnalyzer [9]), Ours-G, Ours-T, and Ours-P. Using our methods, samples that fall into other clusters (A and B) are placed into their corresponding clusters, and irregular boundaries between different clusters (C) become regular (D, E, and F). Boundaries in Ours-G and Ours-T contain many slanted segments (D, E, and G), while boundaries in Ours-P mainly consist of horizontal and vertical segments (F and H).

on area-based measures as the grid size increases. As a result, Ours-T achieved higher area ratios and triple ratios with larger grid sizes. In contrast, the calculation of boundary-based convexity measures is not affected by the changes in grid size. Therefore, Ours-P did not achieve higher perimeter ratios or cut ratios with larger grid sizes.

Fig. 8 presents the layout results generated by the baseline, Ours-G, Ours-T, and Ours-P on six example datasets. The full results are summarized in supplemental material. In the layouts generated by the baseline, some samples fall into the clusters they do not belong to (Fig. 8A and B), and the boundaries between different clusters are irregular (Fig. 8C). After the global assignment phase, the samples within the same cluster are grouped together, and the boundaries become more regular (Fig. 8D). It is also noted that the results generated by Ours-G are similar to Ours-T, which is consistent with our findings that Ours-G achieves a more notable improvement on the area-based convexity measures in Sec. 5.1.2. Further comparison between Ours-G and Ours-T reveals that the boundaries generated by Ours-T are closer to line segments (Fig. 8E) than Ours-G, leading to higher area-based convexity scores. In contrast, the boundaries generated by Ours-P mainly consist of horizontal and vertical line segments (Fig. 8F), making the layout results dissimilar from the results generated by Ours-G and Ours-T. It is also observed that a shape ◺ (Fig. 8G) tends to be optimized towards ◣ (Fig. 8H), which has higher boundary-based convexity scores but lower area-based convexity scores.

5.1.4 Running Time

We evaluated the running time of our methods for 3 grid sizes on a desktop PC with an Intel i9-13900K CPU (5.0 GHz). The results were averaged over 11 datasets. As shown in Table 3(a), our methods generated a 30x30 grid layout in less than 1 second, and a 40x40 grid layout in around 3 seconds. Moreover, the results indicated that the global assignment phase consumed most of the time. Further analysis revealed that, on average, the optimal value of λ required solving the linear assignment problem approximately six times. This process can be accelerated if λ is fixed. A practical way to find an appropriate λ is to test different values on a set of representative datasets and choose the one that works well for most of them. As shown in Table 3(b), the running time of Ours-G approximately equals that of the baseline when λ is fixed, and the total time of our methods (Ours-T and Ours-P) is a little bit longer. For example, when generating a 40x40 grid layout, the baseline takes 0.319 seconds, and Ours-P and Ours-T take 0.961 and 1.513 seconds, respectively. Previous research has shown that a minimum size of 32x32 pixels is required to identify important objects in an image [52]. Taking a display with a resolution of 1920x1080 as an example, each cell in a 40x40 grid has a maximum width of only $1080/40 = 27$ pixels, which is below the minimum size. Thus, our methods closely approach the real-time requirements of most applications.

Table 3: Running time comparison (in seconds) of different methods to generate grid layouts with different sizes. Baseline: OoDAnalyzer [9].

(a) Adaptive λ in the global assignment phase.

Method	20x20	30x30	40x40
Ours-G	0.059	0.396	1.821
Ours-T	0.101	0.709	3.050
Ours-P	0.133	0.636	2.465

(b) Fixed λ in the global assignment phase.

Method	20x20	30x30	40x40
Baseline	0.013	0.081	0.319
Ours-G	0.009	0.064	0.291
Ours-T	0.055	0.377	1.513
Ours-P	0.092	0.309	0.961

(a) Baseline (b) Ours-T

Fig. 9: Compared with the baseline (a), the samples predicted as "4" (A and B) are merged into the cluster of "4" in our layout (b).

5.2 Use Cases

We present two use cases to showcase how our layout method facilitates 1) identifying misclassified samples; and 2) analyzing out-of-distribution (OoD) samples.

5.2.1 Identifying misclassified samples

This use case illustrates how our method aids in identifying misclassified samples in a classification task. We used the USPS dataset [41], which consists of 9,298 gray-scale images of handwritten digits from 0 to 9. As in the quantitative experiments, feature vectors were extracted using CLIP [41], and predictions were generated using a k-NN classifier (accuracy: 93.27%). Four classes, 1/4/7/9, were chosen for analysis because most of the misclassifications happened among the samples of these four classes. As the ground-truth labels are not available, users need to examine the samples in the grid layout to identify misclassified samples. To facilitate the identification, samples with the same predictions were treated as a single cluster, and we utilized both position and color to encode cluster structures.

Figs. 9(a) and (b) show a part of the grid layouts generated by the baseline and Ours-T, respectively. We chose Ours-T because the triple ratio is the most favored measure in our user study. In the baseline layout (Fig. 9(a)), the samples of "4" in Fig. 9A are placed far away from the cluster of "4." This arrangement may lead users to draw the wrong conclusion that the model predicts these samples to be not similar to "4." However, these samples are similar to other samples of "4" and are predicted as "4." By preserving cluster structures (Fig. 9(b)), these samples are merged into the cluster of "4" (Fig. 9A′), which reduces false inferences. Furthermore, misclassified samples, such as the samples in Fig. 9B, which are "7" but misclassified as "4," still remain in the cluster of "4" and can be easily identified in the cluster-aware layout.

5.2.2 Analyzing OoD samples

The second use case illustrates how our method facilitates the analysis of OoD samples, the test samples that are not well covered by training samples. Analyzing why OoD samples appear and adding corresponding samples to the training data can boost model performance [9]. A recent work, OoDAnalyzer [9], utilizes a grid visualization to help analyze OoD samples. We were interested in whether our cluster-aware grid layout could further improve analysis efficiency and help identify more OoD samples. Therefore, we invited one author of OoDAnalyzer (E1) to conduct the analysis on the OoD-Animals dataset, which was used in the case study of OoDAnalyzer. This dataset contains 7,270 training samples and 19,413 test samples of five categories: cat, dog, rabbit, tiger, and wolf. Following OoDAnalyzer, the color hue encodes the prediction. The color saturation encodes the OoD score, and a darker color indicates a larger OoD score that warrants examination. Similar to the first use case, samples with the same predictions were treated as a single cluster.

E1 first compared the grid layout used in OoDAnalyzer (Fig. 1(c)) and the corresponding grid layout generated by Ours-T (Fig. 1(d)). The OoD samples found in the previous work [9] can be easily identified in our layout. For example, in Fig. 1(c), some dark blue cells (cat) fell into different regions C_1–C_3. After examining the samples in these three regions one by one, E1 found that all these dark blue cells were the samples of "leopard." Since there was no category "leopard" in the training data, these samples were predicted as the most similar category "cat" (blue) but had high OoD scores. However, it was hard to analyze them in OoDAnalyzer because they were scattered into several categories. This arrangement may even lead users to draw the wrong

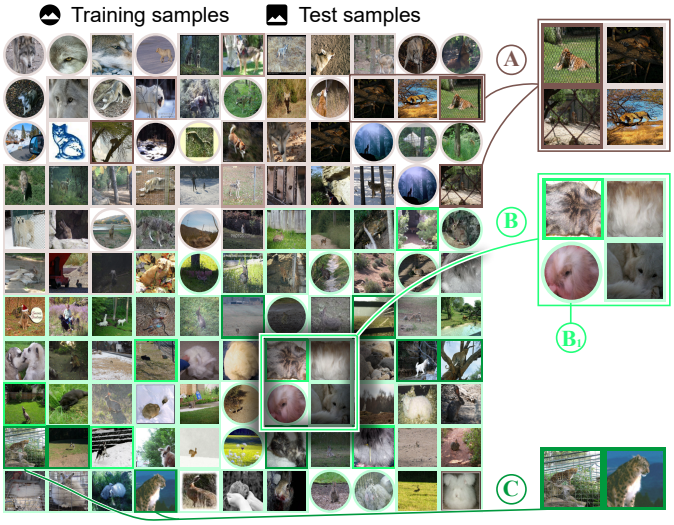


Fig. 10: After zooming into region Fig. 1D, more samples with high OoD scores are identified (A and C). Our layout also reveals that some samples with only part of an animal (B) are misclassified as "rabbit" due to a low-quality training sample with only the hair of a rabbit (B_1).

conclusion that the model predicts the samples in C_3 to be closely related to "rabbit." By preserving cluster structures (Fig. 1(d)), these OoD samples of "leopard" were grouped together in the cluster of "cat" (C'). This enables users to efficiently identify these OoD samples as a whole and take the associated actions.

Our cluster-aware layout also helped identify more OoD samples that were not identified in the previous work [9]. For example, the samples with high OoD scores in Fig. 1D were not identified before because they were not grouped together in OoDAnalyzer (Fig. 1(c)) and thus did not trigger examinations. E1 then zoomed into Fig. 1D to examine these samples (Fig. 10). Upon examination, E1 discovered that the samples with high OoD scores were "leopard" or "tiger" but predicted as "wolf" (Fig. 10A) or "rabbit" (Fig. 10C). Furthermore, during the examination, a new finding was discovered by E1. Many samples that only included part of an animal were predicted as "rabbit" (Fig. 10B). Upon further analysis, it was discovered that a low-quality training sample with only the hair of a rabbit (Fig. 10B_1) explained why the model tended to recognize images with hair as "rabbit."

5.3 Expert Feedback and Discussions

We interviewed three experts who are not co-authors of this work. E1 participated in the second use case, E2 is a senior computer vision researcher who often utilizes the grid layout to explore image datasets, and E3 is a mathematician with rich experience in convexity. Initially, we introduced both OoDAnalyzer and its enhanced version that integrates Ours-T as the layout method. Then the experts freely explored the OoD-Animals dataset using both of the systems and compared the layouts. We also collected their feedback during the exploration process. Each interview lasted 40-60 minutes. Our layout method received positive feedback from all the experts regarding its usefulness.

Enhancing cluster perception. All the experts agreed that our layout method enhanced the cluster perception and aided in efficiently identifying confused sample predictions. E2 commented that some light colors used in the original OoDAnalyzer system were hard to differentiate from each other, such as light brown and light purple. As a result, it took him more time to recognize the predictions when the samples with similar prediction colors were mixed together. Our method alleviated this by grouping samples with the same predictions together. E1 indicated that enhancing cluster perception helped him find more OoD samples. "As the samples with high OoD scores fall into different regions in the original system, I missed some of them in my previous analysis. The clear cluster structures in the new layout enable me to identify more OoD samples easily and prepare better training data."

Extensibility. The extensibility of our layout method comes from three sources. First, our method takes a grid layout as input, which offers the flexibility to integrate any existing layout method into our layout process. In addition to an input grid layout, E1 pointed out that our method could generate a grid layout based on a scatterplot by modifying the proximity calculation. Therefore, existing dimensional reduction techniques are readily integrated into our method for visualizing high-dimensional data. Second, our method supports different convexity measures to meet different analysis needs. For example, optimizing area-based measures usually makes fewer adjustments and hence better preserves both the achieved proximity and compactness. Optimizing boundary-based measures tends to result in cluster shapes that are close to the combinations of rectangles, making them suitable for smaller grid sizes. Users can either choose a convexity measure from our provided measures or even customize a measure that fits their goals. Third, the global-local strategy can be extended to optimize other measures, such as aesthetics and continuity. The global assignment can be used to optimize measures that are affected by all the grid cells, while the local adjustment can iteratively optimize measures that are only affected by specific cells. For example, E3 noted that when continuity is a concern, it is necessary to reject a swap operation that separates a cluster into two disconnected parts.

The experts have also suggested two interesting research topics, which provide insights for future studies.

Combining with other design variables. We have shown the effectiveness of our method in enhancing the perception of cluster structures by adjusting the positions of visual elements. E1 pointed out that in addition to the positions, there were also other design variables that could be used to enhance the cluster perception, such as color and shape. Following the Gestalt principle of similarity [55], we can encode samples in the same cluster using the same color or shape. This is already employed in our method. The principle of common region also suggests that we can add contours to group the samples in the same cluster. To further improve cluster perception, it is interesting to investigate how to combine these methods effectively. In addition, combining multiple methods together may cause cognitive overload. Therefore, it is worth studying how to balance the trade-off between enhancing cluster perception and avoiding cognitive overload.

Interactive editing. Although our method has provided a grid layout that well preserves cluster structures, E2 expressed his need to further adjust the grid layout based on his requirements. "Sometimes I would like to locally modify the boundary to make it clearer or force two similar samples to be placed adjacently." It is worth exploring user-friendly interactions that allow users to directly edit the layouts toward their desired effects. For example, at the sample level, we consider supporting users to move samples to the desired positions and select multiple samples to add must-link/cannot-link constraints among them. At the cluster level, users can sketch the desired cluster shapes or change the convexity calculation of certain clusters. At the global level, users have the flexibility to adjust λ, which balances the preservation of the original layout and the preservation of cluster structures. The larger the λ value, the better the original layout is preserved.

6 Conclusion

We present a cluster-aware grid layout method that enhances the perception of cluster structures by optimizing proximity, compactness, and convexity simultaneously. Starting from the input layout generated by any existing method, our method first optimizes proximity and compactness together in the global assignment phase. Then, a local adjustment phase swaps boundary cells between different clusters to improve convexity. To determine the convexity measure used in the local adjustment phase, we conducted a user study and identified two representative measures, triple ratio and perimeter ratio, to accommodate the diverse preferences of users. The quantitative evaluations demonstrate that our method achieves experimentally optimal balances among proximity, compactness, and convexity. Two use cases are also presented to demonstrate how our method can be practically useful in exploring image datasets and analyzing model predictions.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grants U21A20469, 61936002, the National Key R&D Program of China under Grant 2020YFB2104100, grants from the Institute Guo Qiang, THUIBCS, and BLBCI, and in part by Tsinghua-Kuaishou Institute of Future Media Data. The authors would like to thank Yifan Hu, Zhen Li, Zhaowei Wang, and Jun Yuan for their valuable contributions to the discussions, and Jiangning Zhu for his assistance in proofreading and voicing our video.

References

[1] K. U. Barthel and N. Hezel. Visually exploring millions of images using image maps and graphs. In *Big Data Analytics for Large-Scale Multimedia Search*, pp. 289–315. John Wiley & Sons, Ltd, 2019. doi: 10.1002/9781119376996.ch11 2

[2] K. U. Barthel, N. Hezel, K. Jung, and K. Schall. Improved evaluation and generation of grid layouts using distance preservation quality and linear assignment sorting. *Computer Graphics Forum*, 42(1):261–276, 2023. doi: 10.1111/cgf.14718 2

[3] B. B. Bederson. PhotoMesa: A zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, pp. 71–80. Orlando, 2001. doi: 10.1145/502348.502359 3

[4] D. Bertucci, M. M. Hamid, Y. Anand, A. Ruangrotsakun, D. Tabatabai, M. Perez, and M. Kahng. DendroMap: Visual exploration of large-scale image datasets for machine learning with treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):320–330, 2023. doi: 10.1109/TVCG.2022.3209425 3

[5] J. R. Bozeman and M. Pilling. The convexity ratio and applications. *Scientiae Mathematicae Japonicae*, 76(1):47–53, 2013. doi: 10.32219/isms.76.1_47 2, 3

[6] G. Y.-Y. Chan, L. G. Nonato, A. Chu, P. Raghavan, V. Aluru, and C. T. Silva. Motion browser: visualizing and understanding complex upper limb movement under obstetrical brachial plexus injuries. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):981–990, 2020. doi: 10.1109/TVCG.2019.2934280 2

[7] C. Chen, Z. Wang, J. Wu, X. Wang, L.-Z. Guo, Y.-F. Li, and S. Liu. Interactive graph construction for graph-based semi-supervised learning. *IEEE Transactions on Visualization and Computer Graphics*, 27(9):3701–3716, 2021. doi: 10.1109/TVCG.2021.3084694 5

[8] C. Chen, J. Wu, X. Wang, S. Xiang, S.-H. Zhang, Q. Tang, and S. Liu. Towards better caption supervision for object detection. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1941–1954, 2022. doi: 10.1109/TVCG.2021.3138933 2

[9] C. Chen, J. Yuan, Y. Lu, Y. Liu, H. Su, S. Yuan, and S. Liu. OoDAnalyzer: Interactive analysis of out-of-distribution samples. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3335–3349, 2021. doi: 10.1109/TVCG.2020.2973258 2, 3, 6, 7, 8, 9

[10] J. Choi, S.-E. Lee, Y. Lee, E. Cho, S. Chang, and W.-K. Jeong. DXplorer: a unified visualization framework for interactive dendritic spine analysis using 3d morphological features. *IEEE Transactions on Visualization and Computer Graphics*, 29(2):1424–1437, 2023. doi: 10.1109/TVCG.2021.3116656 2

[11] A. Corrado. Animals-10 dataset. https://www.kaggle.com/datasets/alessiocorrado99/animals10/, 2019. Last accessed 2023-7-1. 4

[12] M. P. Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016. 2, 3

[13] A. Efrat, Y. Hu, S. G. Kobourov, and S. Pupyrev. MapSets: Visualizing embedded and clustered graphs. In *Graph Drawing*, pp. 452–463. Würzburg, 2014. doi: 10.1007/978-3-662-45803-7_38 2

[14] D. Eppstein, M. van Kreveld, B. Speckmann, and F. Staals. Improved grid map layout by point set matching. *International Journal of Computational Geometry & Applications*, 25(02):101–122, 2015. doi: 10.1142/S0218195915500077 3

[15] C. Felix, A. V. Pandey, and E. Bertini. Texttile: An interactive visualization tool for seamless exploratory analysis of structured data and unstructured text. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):161–170, 2017. doi: 10.1109/TVCG.2016.2598447 2

[16] S. Frey. Optimizing grid layouts for level-of-detail exploration of large data collections. *Computer Graphics Forum*, 41(3):247–258, 2022. doi: 10.1111/cgf.14537 1, 3

[17] O. Fried, S. DiVerdi, M. Halber, E. Sizikova, and A. Finkelstein. Isomatch: Creating informative grid layouts. *Computer Graphics Forum*, 34(2):155–166, 2015. doi: 10.1111/cgf.12549 1, 3

[18] A. Halnaut, R. Giot, R. Bourqui, and D. Auber. VRGrid: Efficient transformation of 2d data into pixel grid layout. In *Proceedings of International Conference Information Visualisation*, pp. 11–20. Vienna, 2022. doi: 10.1109/iv56949.2022.00012 2, 3

[19] A. Held and K. Abe. On approximate convexity. *Pattern Recognition Letters*, 15(6):611–618, 1994. doi: 10.1016/0167-8655(94)90022-1 2, 3

[20] 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, 2023. doi: 10.48550/arXiv.1903.06262 3

[21] J. Huang, A. Mishra, B. C. Kwon, and C. Bryan. ConceptExplainer: Interactive explanation for deep neural networks from a concept perspective. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):831–841, 2023. doi: 10.1109/TVCG.2022.3209384 2

[22] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440 4

[23] T. Itoh, C. Muelder, K.-L. Ma, and J. Sese. A hybrid space-filling and force-directed layout method for visualizing multiple-category graphs. In *Proceedings of the IEEE Pacific Visualization Symposium*, pp. 121–128. Beijing, 2009. doi: 10.1109/PACIFICVIS.2009.4906846 3

[24] G. Kanizsa. Convexity and symmetry in figure-ground organization. *Vision and Artifact*, pp. 25–32, 1976. 1, 5

[25] D. Keefe, M. Ewert, W. Ribarsky, and R. Chang. Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1383–1390, 2009. doi: 10.1109/TVCG.2009.152 2

[26] R. Kehlbeck, J. Görtler, Y. Wang, and O. Deussen. SPEULER: Semantics-preserving euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):433–442, 2022. doi: 10.1109/TVCG.2021.3114834 5

[27] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009. 4

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791 4

[29] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):91–100, 2017. doi: 10.1109/tvcg.2016.2598831 2

[30] S. Liu, Y. Liang, and A. Gitter. Loss-balanced task weighting to reduce negative transfer in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence.*, pp. 9977–9978. Honolulu, 2019. doi: 10.1609/aaai.v33i01.33019977 6

[31] X. Liu, Y. Hu, S. North, and H.-W. Shen. CorrelatedMultiples: Spatially coherent small multiples with constrained multi-dimensional scaling. *Computer Graphics Forum*, 37(1):7–18, 2018. doi: 10.1111/cgf.12526 3

[32] M. Lu, S. Wang, J. Lanir, N. Fish, Y. Yue, D. Cohen-Or, and H. Huang. Winglets: Visualizing association with uncertainty in multi-class scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):770–779, 2020. doi: 10.1109/TVCG.2019.2934811 5

[33] T. Major and R. C. Basole. Graphicle: Exploring units, networks, and context in a blended visualization approach. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):576–585, 2019. doi: 10.1109/TVCG.2018.2865151 2

[34] J. Matejka, M. Glueck, E. Bradner, A. Hashemi, T. Grossman, and G. Fitzmaurice. Dream lens: Exploration and visualization of large-scale generative design datasets. In *Proceedings of the CHI conference on human factors in computing systems*, pp. 1–12. Montreal, 2018. doi: 10.1145/3173574.3173943 2

[35] W. Meulemans, J. Dykes, A. Slingsby, C. Turkay, and J. Wood. Small multiples with gaps. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):381–390, 2017. doi: 10.1109/TVCG.2016.2598542 2

[36] C. Muelder and K.-L. Ma. Rapid graph layout using space filling curves. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1301–1308, 2008. doi: 10.1109/TVCG.2008.158 2

[37] M. Oppermann and T. Munzner. VizSnippets: Compressing visualization bundles into representative previews for browsing visualization collections. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):747–757, 2022. doi: 10.1109/TVCG.2021.3114841 2

[38] X. Pan, F. Tang, W. Dong, C. Ma, Y. Meng, F. Huang, T.-Y. Lee, and C. Xu. Content-based visual summarization for image collections. *IEEE Transactions on Visualization and Computer Graphics*, 27(4):2298–2312, 2021. doi: 10.1109/TVCG.2019.2948611 3

[39] M. Peura, J. Iivarinen, et al. Efficiency of simple shape descriptors. In *Proceedings of International Workshop on Visual Form*, pp. 443–451. Capri, 1997. 2, 3

[40] N. Quadrianto, L. Song, and A. Smola. Kernelized sorting. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds., *Proceedings of Advances in Neural Information Processing Systems*, pp. 1–8. Vancouver, 2008. 1, 2

[41] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of International Conference on Machine Learning*, pp. 8748–8763. Virtual Event, 2021. 6, 8

[42] E. Rahtu, M. Salo, and J. Heikkila. A new convexity measure based on a probabilistic interpretation of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1501–1512, 2006. doi: 10.1109/TPAMI.2006.175 2, 3

[43] P. L. Rosin and C. L. Mumford. A symmetric convexity measure. *Computer Vision and Image Understanding*, 103(2):101–111, 2006. doi: 10.1016/j.cviu.2006.04.002 2, 3

[44] P. L. Rosin and J. Žunić. Probabilistic convexity measure. *IET Image Processing*, 1(2):182–188, 2007. doi: 10.1049/iet-ipr:20060185 2

[45] P. Rottmann, M. Wallinger, A. Bonerath, S. Gedicke, M. Nöllenburg, and J.-H. Haunert. MosaicSets: Embedding set systems into grid graphs. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):875–885, 2023. doi: 10.1109/TVCG.2022.3209485 1, 2, 5, 6

[46] R. Sevastjanova, E. Cakmak, S. Ravfogel, R. Cotterell, and M. El-Assady. Visual comparison of language model adaptation. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1178–1188, 2023. doi: 10.1109/TVCG.2022.3209458 2

[47] H. Song, J. Lee, T. J. Kim, K. H. Lee, B. Kim, and J. Seo. GazeDx: Interactive visual analytics framework for comparative gaze analysis with volumetric medical images. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):311–320, 2017. doi: 10.1109/TVCG.2016.2598796 2

[48] Y. Song, F. Tang, W. Dong, F. Huang, T.-Y. Lee, and C. Xu. Balance-aware grid collage for small image collections. *IEEE Transactions on Visualization and Computer Graphics*, 29(2):1330–1344, 2023. doi: 10.1109/TVCG.2021.3113031 3

[49] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014. 2, 3

[50] G. Strong and M. Gong. Self-sorting map: An efficient algorithm for presenting multimedia data in structured layouts. *IEEE Transactions on Multimedia*, 16(4):1045–1058, 2014. doi: 10.1109/TMM.2014.2306183 1, 2

[51] D. Todorovic. Gestalt principles. *Scholarpedia*, 3(12):5345, 2008. doi: 10.4249/scholarpedia.5345 1, 5

[52] A. Torralba. How many pixels make an image? *Visual neuroscience*, 26(1):123–131, 2009. doi: 10.1017/S0952523808080930 8

[53] Y. Tu, R. Qiu, Y.-S. Wang, P.-Y. Yen, and H.-W. Shen. PhraseMap: Attention-based keyphrases recommendation for information seeking. *IEEE Transactions on Visualization and Computer Graphics*, 2022. to be published. doi: 10.1109/TVCG.2022.3225114 2

[54] J. Žunić and P. L. Rosin. A new convexity measure for polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):923–934, 2004. doi: 10.1109/TPAMI.2004.19 2, 3

[55] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt. A century of gestalt psychology in visual perception: I. perceptual grouping and figure-ground organization. *Psychological bulletin*, 138(6):1172–1217, 2012. doi: 10.1037/a0029333 5, 9

[56] J. Xia, L. Huang, W. Lin, X. Zhao, J. Wu, Y. Chen, Y. Zhao, and W. Chen. Interactive visual cluster analysis by contrastive dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):734–744, 2023. doi: 10.1109/TVCG.2022.3209423 6

[57] C. Yang, Z. Zhang, Z. Fan, R. Jiang, Q. Chen, X. Song, and R. Shibasaki. EpiMob: Interactive visual analytics of citywide human mobility restrictions for epidemic control. *IEEE Transactions on Visualization and Computer Graphics*, 28(8):3586–3601, 2022. doi: 10.1109/TVCG.2022.3165385 2

[58] V. Yoghourdjian, T. Dwyer, G. Gange, S. Kieffer, K. Klein, and K. Marriott. High-quality ultra-compact grid layout of grouped networks. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):339–348, 2016. doi: 10.1109/TVCG.2015.2467251 3

[59] J. Yuan, M. Liu, F. Tian, and S. Liu. Visual analysis of neural architecture spaces for summarizing design principles. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):288–298, 2023. doi: 10.1109/TVCG.2022.3209404 2

[60] W. Zeng, C. Lin, J. Lin, J. Jiang, J. Xia, C. Turkay, and W. Chen. Revisiting the modifiable areal unit problem in deep traffic prediction with visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2), 2021. doi: 10.1109/TVCG.2020.3030410 2

[61] J. Žunić and P. L. Rosin. Measuring shapes with desired convex polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6):1394–1407, 2020. doi: 10.1109/TPAMI.2019.2898830 2, 3