Hierarchical Fuzzy-Cluster-Aware Grid Layout for Large-Scale Data

Yuxing Zhou, Changjian Chen, Zhiyang Shen, Jiangning Zhu, Jiashu Chen, Weikai Yang, Shixia Liu

***Abstract*—Fuzzy clusters, where ambiguous samples belong to multiple clusters, are common in real-world applications. Analyzing such ambiguous samples in large-scale datasets is crucial for practical applications, such as diagnosing machine learning models. A promising method to support such analysis is through hierarchical cluster-aware grid visualizations, which offer high space efficiency and clear cluster perception. However, existing cluster-aware grid layout methods cannot clarify ambiguity among fuzzy clusters, which limits their effectiveness in fuzzy cluster analysis. To tackle this issue, we introduce a hierarchical fuzzy-cluster-aware grid layout method that supports hierarchical exploration of large-scale datasets. Throughout the hierarchical exploration, it is crucial to facilitate fuzzy cluster analysis while maintaining visual continuity for users. To achieve this, we propose a two-step optimization strategy for enhancing cluster perception, clarifying ambiguity, and preserving stability during the exploration. The first step is to create cluster-aware partitions, where each partition corresponds to a cluster. This step focuses on enhancing cluster perception and maintaining the previous shapes and positions of clusters to preserve stability at the cluster level. The second step is to generate a grid layout for each partition. In addition to placing similar samples together, this step also places ambiguous samples near the boundaries to clarify ambiguity and reveal the root causes of their occurrences and maintains the relative positions of the samples in the same cluster to preserve stability at the sample level. Several quantitative experiments and a use case are conducted to demonstrate the effectiveness and usefulness of our method in analyzing large-scale datasets, especially in fuzzy cluster analysis.**

***Index Terms*—Fuzzy clusters, grid visualization, scalability, hierarchical exploration**

I. Introduction

Fuzzy clusters, where ambiguous samples belong to multiple clusters, are common in real-world applications. For example, an image in ImageNet [44] can belong to multiple classes because it may contain multiple objects. Hierarchical analysis of such ambiguous samples is beneficial in many big data analysis tasks. For example, it helps identify samples with unclear categories in a large-scale training dataset and promotes the understanding of the underlying reasons for their occurrences. This insight can then be used to improve the accuracy and robustness of machine learning models [22].

Many types of visualizations can support hierarchical analysis, such as tree diagrams, treemaps, and hierarchical grid visualizations [33], [59]. Among them, hierarchical grid visualizations are particularly well-suited for fuzzy cluster analysis due to their high space efficiency and proximity preservation [7]. Along this line, Zhou *et al.* [61] have made an initial effort. However, directly applying this method for fuzzy cluster analysis still presents two limitations. **First**, Zhou *et al.*'s method assumes a sample exclusively belongs to a single cluster. This results in some ambiguous samples being placed near the cluster centers with incorrect context of samples. Such misplacement hinders users from understanding the root causes of ambiguous samples. This issue is exacerbated during hierarchical exploration due to accumulated errors in sample positioning. **Second**, this method does not consider the preservation of the previous positions and shapes of clusters and cannot well preserve the previous positions of samples within each cluster during interactive exploration (*e.g.*, Fig. 1(a)). According to the work of Bridgeman and Tamassia [3] and our interviews with the experts, both aspects are essential to maintaining visual continuity for users.

To address these limitations, we develop a hierarchical fuzzy-cluster-aware grid layout method that supports hierarchical exploration and analysis of fuzzy clusters in large-scale datasets. Our method processes a large-scale dataset organized hierarchically, where initially, a set of samples from the first-level clusters is sampled and displayed in a grid layout. Users can then explore the displayed samples and select areas of interest for further exploration. To facilitate the exploration, the grid layout is designed to enhance cluster perception, clarify ambiguity, and preserve stability. However, it is challenging to balance these three aspects. Previous studies [4], [39] have shown that cluster-level optimization objectives (cluster perception and cluster-level stability) are less affected by sample-level optimization objectives (ambiguity and sample-level stability). This motivates us to use the divide-and-conquer paradigm, formulating a two-step optimization strategy. Specifically, the first step creates cluster-aware partitions, where each partition corresponds to a cluster. This step effectively enhances cluster perception and preserves the previous shapes and relative positions of clusters. The second step generates a grid layout for each partition. This step places similar samples together while placing ambiguous

This work was supported by the National Natural Science Foundation of China under grants U21A20469, 61936002, and 62402167, in part by Tsinghua-Kuaishou Institute of Future Media Data, the Hunan Natural Science Foundation under the grant 2025JJ60419, and the Science and Technology Innovation Program of Hunan Province under the grant 2023ZJ1080. (Corresponding author: Changjian Chen.)

Yuxing Zhou, Zhiyang Shen, Jiangning Zhu, Jiashu Chen, Shixia Liu are with the School of Software, BNRist, Tsinghua University, Beijing 100084, China (e-mail: jszhouyuxing@gmail.com, shenzhiy21@mails.tsinghua.edu.cn, zjn23@mails.tsinghua.edu.cn, cjs22@mails.tsinghua.edu.cn, shixia@tsinghua.edu.cn).

Changjian Chen is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan, 410082, China (e-mail: changjianchen@hnu.edu.cn).

Weikai Yang is with the Data Science and Analytics Thrust, Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong, 510730, China (e-mail: weikaiyang@hkust-gz.edu.cn).

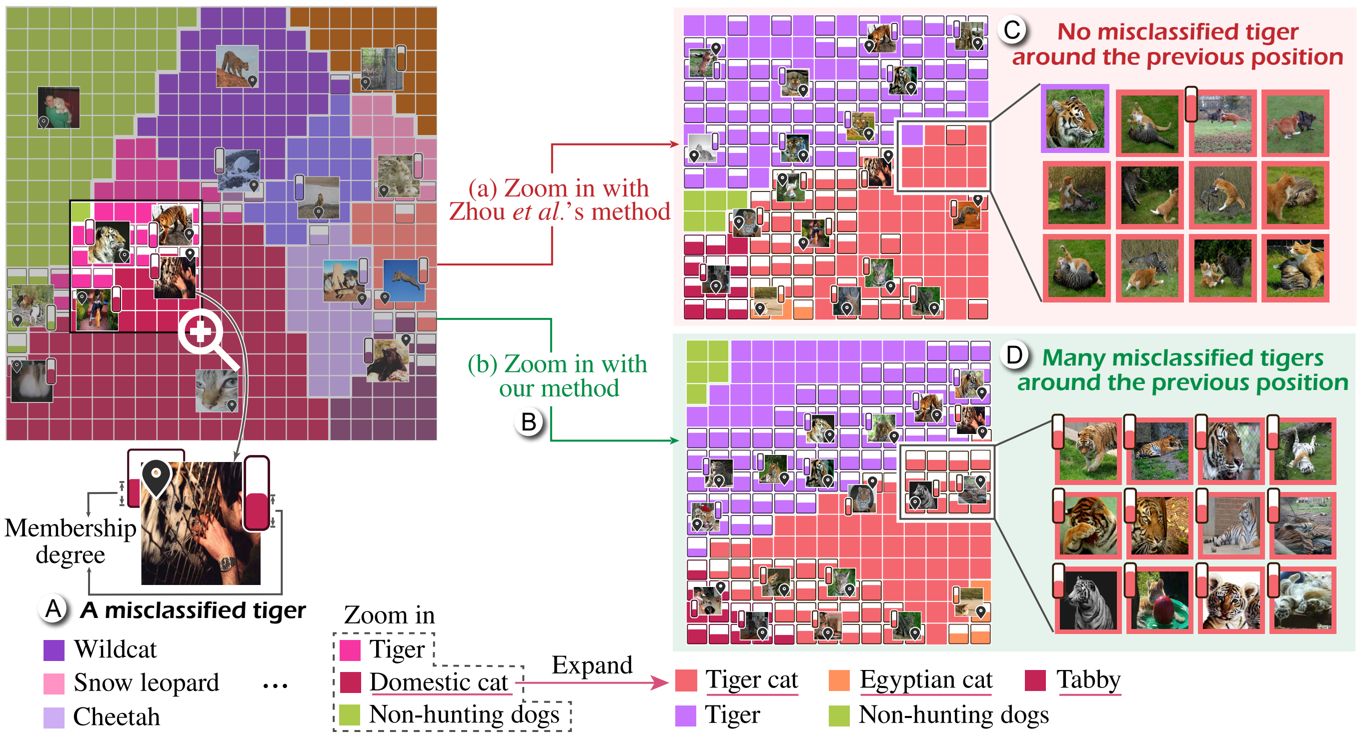

C No misclassified tiger around the previous position
(a) Zoom in with Zhou et al.'s method
(b) Zoom in with our method
B
D Many misclassified tigers around the previous position
Membership degree
A A misclassified tiger
Wildcat
Snow leopard
Cheetah
...
Zoom in
Tiger
Domestic cat
Non-hunting dogs
Expand
Tiger cat
Egyptian cat
Tabby
Tiger
Non-hunting dogs

Fig. 1. By checking the representative images, the user identifies a misclassified tiger and selects an area around it to zoom in for further analysis. (a) With the state-of-the-art hierarchical cluster-aware grid layout method, Zhou *et al.*'s method [61], the user finds no more misclassified tigers around the previous position. (b) With our method, the user identifies many misclassified tigers around the previous position, which saves him a lot of time.

samples close to the cluster boundaries and preserving the relative positions of samples. We conducted several experiments to demonstrate that our method is scalable to hundreds of thousands of samples while effectively balancing cluster perception enhancement, ambiguity clarification, and stability preservation. Additionally, we further illustrate the utility of our method through a use case that highlights its effectiveness in analyzing model predictions.

In summary, the core contributions of our work include:

- **A hierarchical fuzzy-cluster-aware grid layout method** that facilitates the exploration of large datasets at different levels of detail.
- **A two-step optimization strategy** that enhances cluster perception, clarifies ambiguity, and preserves stability.
- **An open-source implementation** of the hierarchical fuzzy-cluster-aware grid layout method for exploring large-scale datasets, which is available at https://osf.io/a8epu/?view_only=fac7bd5cbfc149fbb373df3e0eb5810f.

II. Related Work

Existing grid visualization methods can be classified into two categories: direct mapping and projection-based methods.

Direct mapping methods directly map samples from high-dimensional spaces to two-dimensional grid cells. One of the pioneering studies in this direction was proposed by Quadrianto *et al.* [40]. They formulated the mapping as a quadratic assignment problem to maximize the correlations between the pairwise distances in the high-dimensional spaces and the grid layout. Such a quadratic assignment formulation has also been employed by Rottmann *et al.* [43] for enhancing the compactness of samples within the same groups, and Yoghourdjian *et al.* [60] for maintaining the proximity of connected nodes in a network while preserving group containment within rectangular regions. In addition to solving a quadratic assignment problem, swapping strategies and neural networks are also utilized to maximize the similarities between neighboring samples. The Self-Sorting Map [49] utilizes the swapping strategies, which initially place samples randomly in a grid layout and then employ hierarchical swapping to improve the similarities between neighboring samples. The effectiveness of the Self-Sorting Map is further improved by Barthel *et al.* [1] with a new similarity measure that is better aligned with human perception, and by Song *et al.* [48] with a reinforcement learning mechanism. Meanwhile, the Self-Organization Map trains a neural network to place similar samples in neighboring grid cells [28]. It is widely used for data exploration, such as cluster analysis [18], [45], [47].

Although the existing direct mapping methods have achieved success in certain applications, such as image gallery generation [1], [40] and graph layout [42], [60], they are unstable and are ineffective in preserving proximity [15]. Therefore, projection-based methods are proposed.

Projection-based methods are designed to generate stable and proximity-preserving grid visualizations. To this end, they first project samples into a set of two-dimensional points with dimension reduction methods. Then, these two-dimensional points are assigned to grid cells. The existing methods primarily differ in the assignment step [12], [19]. Fried *et al.* [15] formulated the assignment as a weighted bipartite graph matching problem and solved it with the Hungarian algorithm. To accelerate the Hungarian algorithm, Chen *et al.* [7] developed a kNN-based approximation motivated by Hall's theorem to reduce the candidate grid cells for each sample. Instead of formulating the assignment as a weighted bipartite graph matching problem, CorrelatedMultiples [34] first employs a force-directed graph layout method to resolve sample overlaps. Then, the samples are perturbed locally to fit in a grid layout. All the aforementioned methods remove empty spaces in the grid layouts to display more samples. However, such a strategy may negatively affect the analysis on class separation and outlier [36]. To address this issue, DGrid [23] divided the space into grids such that one grid contains no more than one projected sample. The grids between samples are left empty to reveal the class separation and outliers.

Although these two categories of methods are effective, most of them suffer from the scalability issue when dealing with large-scale datasets. Therefore, several hierarchical grid layout methods have been proposed [2], [14], [61]. Among these methods, the most relevant ones are DendroMap [2] and Zhou *et al.*'s method [61]. DendroMap first clusters samples into a hierarchy. At each hierarchical level, an improved slice-dice treemap layout is utilized to place similar samples close to each other. Zhou *et al.*'s method first samples several samples to generate a cluster-aware grid visualization and assigns the remaining samples to the nearest displayed samples. If users select a set of displayed samples, these samples and some of their assigned samples are displayed for further analysis. Despite their effectiveness, they fail to accurately place ambiguous samples that can be easily mispredicted by the models near the cluster boundaries. Such misplacement poses a challenge for machine learning experts in analyzing these ambiguous samples, leading to an inaccurate understanding of model performance. In addition, neither of them preserves stability well during exploration. This often disrupts the visual continuity of exploration. To address these issues, we propose a scalable fuzzy-cluster-aware grid layout method that supports hierarchical exploration. During the exploration, a two-step optimization strategy is utilized to enhance cluster perception, clarify ambiguity, and preserve stability.

III. Design Goals

The development of the hierarchical fuzzy-cluster-aware grid layout method is based on collaboration with four machine learning experts and a literature review of existing grid visualizations. The machine learning experts include two Ph.D. students and two professors. None of them are the co-authors of this work. Two of them also have rich working experience in technology companies, ensuring the experts well represent both academia and industry. All of them frequently use data exploration toolkits (*e.g.*, FiftyOne [37]). In these toolkits, grid visualizations are key components. However, all these grid visualizations are less effective in analyzing fuzzy clusters, and most of them do not support the analysis of large-scale datasets. We have now summarized the specific disadvantages of these grid visualizations in the supplemental material. Therefore, they desire a tool to effectively analyze fuzzy clusters in large-scale datasets. To identify key design goals, we conducted four semi-structured interviews with the experts, each of which lasted 40-60 minutes. Based on the interviews and the literature review, we identified the following three design goals.

G1: Supporting cluster-aware hierarchical exploration. One of the most effective strategies for supporting the analysis of large-scale datasets is hierarchical exploration [5], [14], [31]. All the experts prefer this strategy because it is familiar to them. The need for hierarchical exploration is also consistent with the findings of previous research [2], [24]. In addition, enhancing cluster perception helps users perceive the samples in a cluster as a whole [26], [42], [51]. This prevents users from drawing wrong conclusions. Therefore, cluster perception should be enhanced during hierarchical exploration.

G2: Placing ambiguous samples near their cluster boundaries to facilitate fuzzy cluster analysis. Our experts also pointed out that to analyze fuzzy clusters, it is critical to place ambiguous samples in correct context. For example, E_2 said, "A correct context can help me correctly understand the root causes of ambiguous samples." However, all the experts have observed the misplacement of ambiguous samples in the existing grid layout methods. This hinders them from understanding the root causes of such ambiguous samples. For example, E_3 said, "I hope the ambiguous samples are placed near their cluster boundaries with correct context, enabling me to quickly identify the samples with which they are confused." While straightforward methods, such as displaying a list of ambiguous samples with their neighbors are available, these often require examining each sample and its neighbors

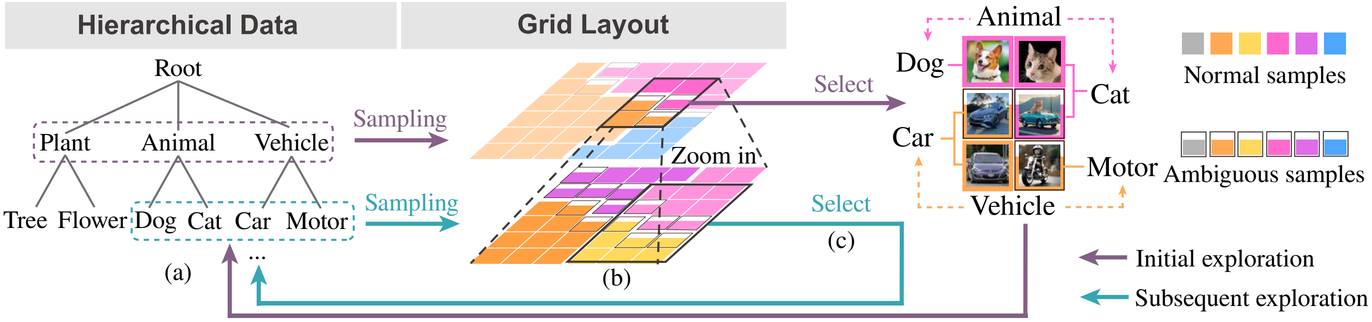


Fig. 2. Method overview: (a) given a large-scale dataset organized as a hierarchy, several samples are sampled for display; (b) the two-step strategy is utilized to generate the grid layout; (c) Users select samples of interest for further exploration.

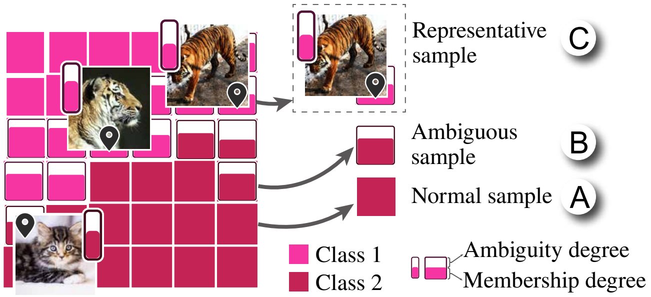


Fig. 3. Illustration of the visual encodings of the interface.

separately. This limits their utility in real-world tasks. For example, when analyzing two similar ambiguous samples with many overlapping neighbors in a grid visualization, users can examine their context simultaneously to quickly identify the root cause of their ambiguity. However, with a list, users have to examine the neighbors of each ambiguous sample individually, which largely increases the workload.

G3: Preserving stability during exploration. Preserving stability during exploration is important for preserving visual continuity for users [53]. According to the work of Bridgeman and Tamassia [3] and the interviews with the experts, we classified the stability into cluster-level (**G3-cluster**) and sample-level (**G3-sample**). For cluster-level stability, it is required to preserve the previous shapes and relative positions of clusters during exploration to facilitate easy tracking of clusters. For sample-level stability, the relative positions of samples should be preserved, which is essential for users to effectively track and analyze the samples of interest.

IV. Hierarchical Fuzzy-Cluster-Aware Grid Layout

A. Method Overview

Fig. 2 shows an overview of our hierarchical fuzzy-cluster-aware grid layout method. The input of our method is a large-scale dataset organized as a hierarchy. If such a hierarchy does not exist, it can be constructed using hierarchical clustering methods [38]. Initially, the clusters at the first hierarchical level are selected (*e.g.*, the clusters "plant," "animal," and "vehicle" in Fig. 2(a)), and several samples are sampled from these clusters for displaying. Then, a grid visualization is generated (Fig. 2(b)) using a two-step optimization strategy to enhance cluster perception (**G1**), clarify ambiguity (**G2**), and preserve stability (**G3**). Users can select several samples of interest in the grid layout (Fig. 2(c)). The clusters of the selected samples can be further expanded into sub-clusters by a simple tree-cut algorithm to avoid significantly imbalanced clusters. This tree-cut algorithm recursively expands the cluster with the most samples if the number of clusters is less than a threshold. The selected samples, along with more samples sampled from their neighbors in the selected/expanded clusters, are displayed in the grid visualization for further analysis. This selection strategy, supported by the tree-cut algorithm, selects a sufficient number of samples to display for each cluster, making the exploration effective even when cluster sizes and hierarchical depths vary.

Color palette. Assigning discriminative and harmonious colors to the grids of different clusters can enhance the perception of clusters [7]. However, generating a color scheme that assigns discriminative and harmonious colors to all clusters within the hierarchy is challenging or even impossible when the cluster number is larger than 26 [13], [52]. To address this color overloading issue, we use DynamicColor [8] to dynamically assign colors to the clusters as they are displayed, ensuring better discrimination and harmony. To ensure color continuity during exploration, DynamicColor utilizes the colors of parent clusters to guide the color assignment of their child clusters. Specifically, DynamicColor first determines a sphere in the color space for each parent cluster. Then, the colors of the child clusters are constrained to fall within the associated sphere. For better color continuity, it is essential to ensure discriminability between and within parent clusters. To ensure the discriminability between parent clusters, the distance between two spheres must exceed the radii of both spheres: $b_{ij} - r_i - r_j > \max(r_i, r_j)$. Here, r_i is the radius of the sphere of the i-th parent cluster, and b_{ij} is the distance between two spheres. To ensure the discriminability within the child clusters of a parent, the size of the spheres must increase with the number of child clusters. According to the experimental results in DynamicColor, the radius is proportional to the square root of the number of child clusters: $r_i/r_j = \sqrt{n_i}/\sqrt{n_j}$. n_i is the number of child clusters of the i-th parent cluster. Finally, the radii of the spheres are determined as the maximum radii that satisfy both $b_{ij} - r_i - r_j > \max(r_i, r_j)$ and $r_i/r_j = \sqrt{n_i}/\sqrt{n_j}$. With this strategy, the colors of child clusters within the same parent cluster are more similar to each other than to those of child clusters from different parent clusters. This helps users capture the correspondence between the colors of child clusters and their respective parent clusters, which effectively addresses the visual overloading issue.

Interface. To help users explore the grid visualizations, we develop a simple interface. As shown in Fig. 3, each square represents a sample. The filled squares denote normal samples (Fig. 3A), while the others represent ambiguous samples (Fig. 3B). The filling heights of ambiguous samples encode their membership degrees to their associated clusters. Accordingly, the height of the unfilled portion represents the ambiguity degree. In our implementation, samples with ambiguity degrees larger than 0.2 are treated as ambiguous samples. Users can adjust this threshold according to different applications. To give an overview of the data, the content of several representative samples is displayed (Fig. 3C), with each sample accompanied by a map pin (📍) indicating its grid cell. The representative samples are selected by prioritizing the most ambiguous ones while ensuring no overlapping between them.

B. Two-Step Optimization Strategy for Grid Layout

A straightforward way to generate fuzzy-cluster-aware and stable grid visualizations is to integrate ambiguity clarification and stability preservation into Zhou *et al.*'s cluster-aware method [61]. Specifically, taking a proximity-preserving grid layout as input, the samples are re-assigned to enhance cluster

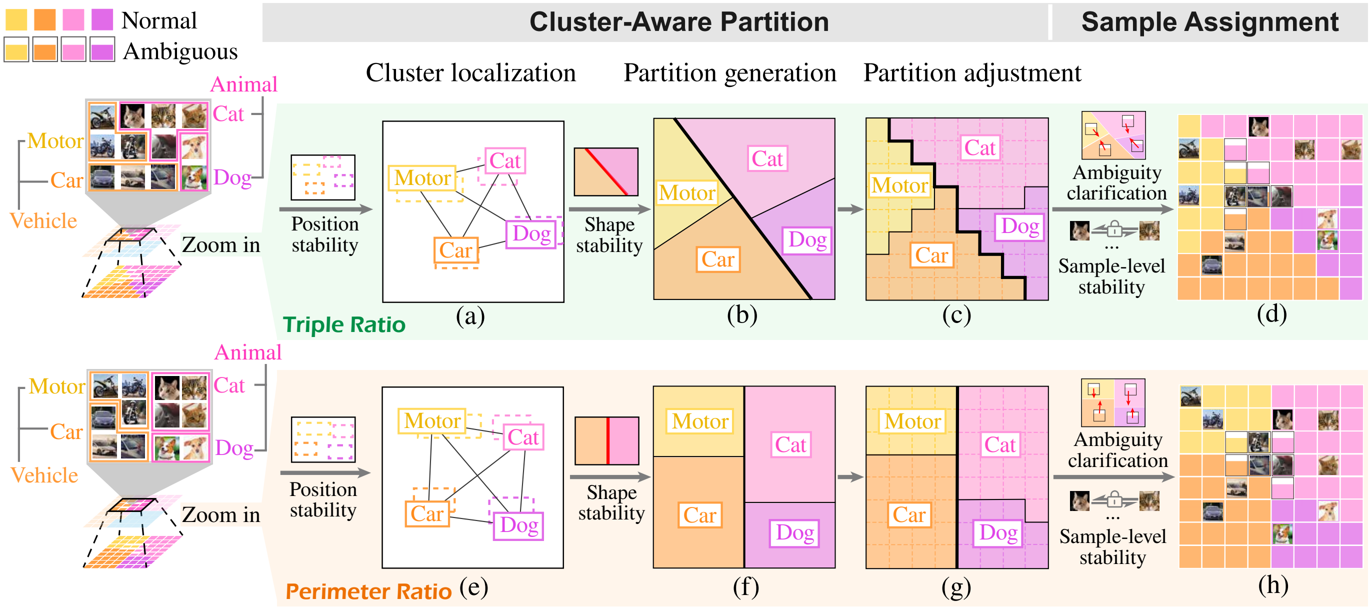


Fig. 4. Two-step optimization strategy: **cluster-aware partition** creates the partitions for each cluster to enhance cluster perception and preserve cluster-level stability; **sample assignment** places similar samples together for each cluster while clarifying ambiguity and preserving sample-level stability.

perception, clarify ambiguity, and preserve stability. However, this does not work for two reasons. First, for cluster-level stability, it is essential to preserve the previous positions and shapes of clusters. However, Zhou *et al.*'s method focuses on individual samples rather than considering each cluster as a whole, resulting in an inability to preserve cluster shapes and positions. Second, for the ambiguous clarification and sample-level stability, it is necessary to consider the relative positions between samples [10]. This complicates the grid layout problem into an NP-hard quadratic assignment problem (QAP) [29]. Although there are approximate algorithms such as the Fast Approximate QAP algorithm (FAQ) [55] with a time complexity of $O(N^3)$, they cannot ensure interactive rates for grid layout generation when dealing with thousands of samples. This constraint limits their utility for hierarchically exploring large-scale datasets [7].

The divide-and-conquer paradigm is an effective way to accelerate the grid layout generation by breaking it into manageable sub-problems. Previous studies [4], [39] point out that clusters are more readily perceived than individuals. It implies that cluster-level optimization objectives (cluster perception and cluster-level stability) are more important than sample-level optimization objectives (ambiguity and sample-level stability). Moreover, the sample-level optimization objectives are more sensitive to the sample positions within each cluster, which results in less influence on cluster-level optimization objectives. Motivated by these findings, we prioritize cluster-level optimization by dividing the problem at the cluster level. Subsequently, sample-level objectives are optimized for each cluster. Accordingly, a two-step strategy is developed, comprising **cluster-aware partition** and **sample assignment** (Fig. 4). Given a selected area in the grid layout, the cluster-aware partition step creates the partitions for each cluster to enhance cluster perception (**G1**) and preserve cluster-level stability (**G3-cluster**). The sample assignment step then assigns similar samples to neighboring grid cells in each partition while clarifying ambiguity (**G2**) and preserving sample-level stability (**G3-sample**). Since sample assignment in each cluster is independent of others, the QAP can be decomposed into multiple smaller QAPs and solved in parallel. This significantly reduces the time cost. Our experiments in Sec. V-A demonstrate that this two-step strategy can generate the grid layouts at interactive rates while enhancing cluster perception, clarifying ambiguity, and preserving stability well.

B1. Cluster-Aware Partition

The cluster-aware partition step creates partitions for each cluster while enhancing cluster perception and preserving cluster-level stability. Enhancing cluster perception involves the preservation of the cluster proximity, compactness, and convexity. Preserving cluster-level stability aims to preserve the relative positions of clusters and their shapes. To achieve these goals, we further divide this step into three sub-steps: **cluster localization**, **partition generation**, and **partition adjustment**. First, the cluster localization finds the cluster centers such that proximities between clusters are preserved, and the positions of clusters are preserved during the exploration (Figs. 4(a) and 4(e)). Then, partitions are generated around the cluster centers that ensure compactness and convexity while preserving the shapes of clusters (Figs. 4(b) and 4(f)). Finally, partition adjustment adjusts the partitions in accordance with the grid constraints to ensure all the partitions are filled with grids (Figs. 4(c) and 4(g)).

Cluster localization. Given M clusters, the cluster localization aims to find M cluster centers $C = \{c_1, c_2, \ldots, c_M\}$ that preserves proximities between clusters and maintains cluster position stability:

$$\min_{C} \sum_{j>i}^{M} \frac{1}{d_{ij}^2}(||c_i - c_j||_2 - d_{ij})^2 + \mu \sum_{i=1}^{|Z|} ||c_{z_i} - c'_{z_i}||_2^2. \quad (1)$$

The first term minimizes the difference between cluster distances in the grid layout and their ideal distances. The second term minimizes the difference between the current and previous positions of the cluster centers during exploration.

In the **first term**, following the Kamada–Kawai layout method [25], we use the L_2 loss to measure the difference between cluster distances in the grid layout and their ideal distances. d_{ij} is the ideal distance between the i-th and the j-th clusters. To ensure proximity (**G1**) and place more ambiguous samples near the boundaries (**G2**), clusters with high similarities, or those that are easily confused with each other, should be positioned closely together. As these two goals are not always positively related, d_{ij} is designed as the sum of two parts to encourage both goals, which is widely used for optimizing multiple objectives [50]. The first part of d_{ij} is the dissimilarity between the i-th and the j-th clusters, which is measured by their Euclidean distance in the high-dimensional space, following the work of Liu *et al.* [32]. The second part is the confusion between the i-th and the j-th clusters. Similar to the work of Gupta *et al.* [9], it is measured by the symmetric Kullback-Leibler divergence of the averaged membership degrees (*e.g.*, averaged prediction probability distributions) of the i-th and the j-th clusters. The two parts are normalized and averaged to obtain the ideal distances. In the **second term**, we use the L_2 loss to measure the difference between the current and previous positions of the cluster centers, following the work of Xu *et al.* [57]. As the displayed clusters are dynamically changed, we only apply this term to clusters that appear in both the current and previous grid layouts. $Z = \{z_1, z_2, ..., z_{|Z|}\}$ are the indices of these clusters. c'_{z_i} is the previous center position of cluster z_i. The weight μ controls the trade-off between the two terms and is determined with the multi-task learning method [8]. Its key idea is dynamically increasing the weights of the terms that are not well-optimized so that they can be further improved. To determine how well each term is optimized, we compare their current scores to their optimal ones. The optimal score for each term is obtained by optimizing this term only. The details of how this method adjusts the weights are given in the supplemental material. Eq. (1) is optimized with the force-directed method [16].

Partition generation. Based on the obtained cluster centers, we generate partitions that can accommodate all the samples in the associated clusters while ensuring cluster compactness and convexity. The previous user study [61] shows that no single convexity measure aligns with the perception of all people. However, two representative measures, the triple ratio and perimeter ratio measures, are preferred by the participants. As these two measures somewhat conflict with each other, we develop two partition generation methods for each of them (Figs. 4(b) and 4(f)).

- Triple Ratio. The triple ratio is defined as the probability that, given a collinear triple (v_i, v_j, v_k), if v_i and v_k are inside the partition, then v_j is also inside the partition. Under this convexity measure, we aim to generate partitions that maximize the triple ratio values while ensuring compactness and the accommodation of all samples in their associated partitions. This generation can be formulated as:

$$\begin{aligned} \min_{\Omega} \quad & \sum_{i=1}^{M} \int_{\Omega_i} ||x - c_i||^2 dx \\ \text{s.t.} \quad & |\Omega_i| = N_i, \ \forall i \in \{1, 2, \ldots, M\}. \end{aligned} \tag{2}$$

The objective function measures the compactness of the partitions in the 2D plane. The constraint ensures that each partition can accommodate all the samples in the associated cluster. According to the previous study [56], this formulation can also ensure convex partitions under the triple ratio measure. Ω_i, c_i, and N_i are the partition, the center, and the number of samples of the i-th cluster, respectively. Optimizing Eq. (2) is equivalent to generating a centroidal power diagram [56].

- Perimeter Ratio. The perimeter ratio is defined as the ratio of the perimeter of the partition's convex hull to its perimeter. Optimizing this measure results in partitions with horizontal or vertical boundaries in grid layouts. To generate such partitions, we enforce that each partition is a rectangle. With this constraint, we aim to ensure the compactness and the accommodation of all samples in their associated partitions. This is equivalent to ensuring the aspect ratios of the partitions are close to one and the areas of the partitions are proportional to the number of samples. To achieve this goal, we develop a greedy method with heuristic cuts, which is motivated by the treemap layout method [46]. Given a set of cluster centers, we first divide the layout space into two partitions with a horizontal or vertical cut. The position and direction of this cut are determined by ensuring that the aspect ratios of the partitions are close to one and each divided partition can accommodate all the samples of the clusters within it. Then, these two partitions are further divided in the same manner. This process is repeated until each partition contains one and only one cluster.

For both convexity measures, we utilize the initialization strategy to preserve shape stability. Specifically, we use the previous partitions to initialize the partition processes. If the selected clusters are expanded into sub-clusters, we further apply this method to generate partitions for the sub-clusters.

Partition adjustment. As the boundaries of the generated partitions may intersect with some grid cells, not all the associated samples can fit in these partitions as grid cells. Therefore, we develop a partition adjustment method that re-assigns these intersected grid cells to satisfy the grid constraint. The basic idea behind this method is that each intersected grid cell should be assigned to the closest partition in priority. Meanwhile, the number of grid cells in each partition should be equal to the number of samples from the associated cluster. This can be formulated as:

$$\begin{aligned} \max_{\pi} \quad & \sum_{i=1}^{N} \sum_{j=1}^{M} \pi_{ij} a_{ij} \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{N} \pi_{ij} = N_j, \ \forall j \in \{1, 2, \ldots, M\}, \\ & \textstyle\sum_{j=1}^{M} \pi_{ij} = 1, \ \forall i \in \{1, 2, \ldots, N\}, \\ & \pi_{ij} \in \{0, 1\}, \ \forall i, j, \\ & \text{Grid cells in each partition are connected.} \end{aligned} \tag{3}$$

N is the number of samples. a_{ij} is the area of the overlap between the i-th grid cell and the j-th partition. The first constraint ensures that the number of grid cells in each partition is equal to the number of samples from the associated cluster. The second and third constraints ensure that each grid cell is assigned to one and only one partition. The fourth constraint ensures that the grid cells in each partition constitute a connected region. $\pi_{ij} = 1$ indicates that the i-th grid cell is assigned to the j-th partition. It can be proved that optimizing Eq. (3) is NP-hard by reducing from the 3-SAT problem [27]. Thus, we develop a greedy algorithm for computational efficiency. For each partition, we assign cells in descending order of their overlap areas with this partition. A cell is not assigned to this partition if the connected region constraint is not satisfied. This process is repeated until all cells are assigned.

B2. Sample Assignment

After obtaining the partitions for all clusters, the sample assignment step (Figs. 4(d) and 4(h)) assigns similar samples to neighboring grid cells for each cluster parallelly. Meanwhile, it should place ambiguous samples close to the corresponding cluster boundaries and preserve sample-level stability during exploration. Accordingly, this can be formulated as a quadratic assignment problem. For the t-th cluster, let $S = \{s_1, s_2, \dots, s_n\}$ denote the samples of this cluster, and $V = \{v_1, v_2, \dots, v_n\}$ denote the grid cells in the associated partition. $n = N_t$ is the number of samples in this cluster. An assignment of the samples to cells δ=$\{\delta_{ij}\}_{1\leq i,j\leq n}$ is a binary matrix, in which $\delta_{ij} = 1$ indicates that the i-th sample is assigned to the j-th cell. The quadratic assignment problem for the t-th cluster is to minimize the following cost:

$$\begin{aligned}\min_{\delta} \quad & \sum_{i,j=1,i\neq j}^{n} \sum_{k,l=1}^{n} p_{ij} \log \frac{p_{ij}}{q_{kl}} \delta_{ik}\delta_{jl} + \omega_1 \sum_{i,j=1}^{n} \sum_{k=1}^{M} f_{ik} r_{jkt} \delta_{ij} \\ & + \omega_2 \sum_{i,j=1}^{|O|} \sum_{k,l=1}^{n} h((g'(s_{o_i}), g'(s_{o_j})), (v_k, v_l)) \delta_{o_i k}\delta_{o_j l} \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{n} \delta_{ij} = 1, \ \forall j \in \{1, 2, \dots, n\}, \\ & \textstyle\sum_{j=1}^{n} \delta_{ij} = 1, \ \forall i \in \{1, 2, \dots, n\}, \\ & \delta_{ij} \in \{0, 1\}, \ \forall i, j. \end{aligned} \tag{4}$$

The **first term** is to place similar samples close to each other. Following t-SNE [35], this term is defined as the Kullback-Leibler divergence between the distributions in the high-dimensional space and the two-dimensional grid layout. p_{ij} is the similarity of samples in the high-dimensional space, and q_{kl} is their similarity in the two-dimensional grid layout. We adopt Gaussian distribution for the calculation of p_{ij} and Student-t distribution for q_{kl}.

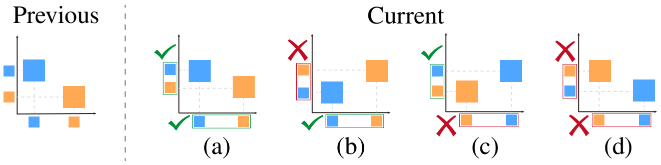


Fig. 5. Four examples with their orthogonal mental distance values: (a) zero, (b) one, (c) one, and (d) two.

The **second term** is to place ambiguous samples close to their corresponding boundaries. Following the work of Xu *et al.* [58], this term is defined as the total products between the membership degrees of ambiguous samples and their distances to the corresponding cluster boundaries. f_{ik} is the membership degree of the i-th sample to the k-th cluster. r_{jkt} is the distance of the j-th cell to the boundary between the k-th cluster and the cluster under processing (*i.e.*, the t-th cluster) if they are adjacent. If the two clusters are not adjacent, r_{jkt} is set to zero since there are no boundaries between them.

The **third term** is to preserve the relative positions of the samples during exploration, which is measured by the commonly used orthogonal mental distance [10]. If the orders of two cells in both the horizontal and vertical directions are not changed (*e.g.*, Fig. 5(a)), the orthogonal mental distance is zero. If the orders are changed in one direction, the orthogonal mental distance is one (*e.g.*, Figs. 5(b) and 5(c)). Otherwise, the distance is two (*e.g.*, Fig. 5(d)). Formally, for two pairs of cells $e = (v_1, v_2) = ((x_1, y_1), (x_2, y_2))$ and $e' = (v_1', v_2') = ((x_1', y_1'), (x_2', y_2'))$, the orthogonal mental distance is given by

$$\begin{aligned} h(e', e) = & |\epsilon(x_1' - x_2')| \cdot |(\epsilon(x_1' - x_2') - \epsilon(x_1 - x_2))|/2 \\ & + |\epsilon(y_1' - y_2')| \cdot |\epsilon(y_1' - y_2') - \epsilon(y_1 - y_2)|/2, \end{aligned} \tag{5}$$

where $\epsilon(x) \in \{-1, 0, 1\}$ is the sign function. $O = \{o_1, o_2, \dots, o_{|O|}\}$ are the set of indices of the samples that appear in the previous layout. $g'(s_i)$ are their corresponding cells in the previous layout. The constraints ensure each sample is assigned to a grid cell, and each grid cell is assigned a sample. The weights ω_1 and ω_2 control the trade-off between the three terms and are also determined by the multi-task learning method [8] used in the cluster localization sub-step. As optimizing Eq. (4) is NP-hard, we adopt an approximate algorithm, FAQ [55], to solve it efficiently.

V. Evaluation

To demonstrate the effectiveness of the proposed hierarchical fuzzy-cluster-aware grid layout method, we conducted two quantitative experiments and a use case on real-world datasets.

A. Quantitative Experiment

Datasets. We evaluated our method on three real-world datasets: CIFAR-100 [30], iNat2021-mini [54], and ImageNet-1k [44]. The number of classes and samples within these datasets are shown in Tab. I. The class numbers of these datasets range from 100 to $10,000$, and the sample numbers range from $60,000$ to $1,281,167$. We used three popular open-sourced model checkpoints [1][2][3] to extract features and predict classes for CIFAR-100, iNat2021-mini, and ImageNet-1k, respectively.

[1] https://huggingface.co/edadaltocg
[2] https://github.com/visipedia/newt
[3] https://github.com/rwightman/pytorch-image-models

TABLE I
THE NUMBERS OF CLASSES AND SAMPLES OF THE DATASETS.

Dataset	Class number	Sample number
CIFAR-100	100	$60,000$
iNat2021-mini	$10,000$	$500,000$
ImageNet-1k	$1,000$	$1,281,167$

Experimental settings. We compared our method with the state-of-the-art (SOTA) ones from the two categories of grid layout methods. The first baseline is Zhou *et al.*'s method, which is the SOTA projection-based grid layout method. To compare Zhou *et al.*'s method with ours, we simulated several zoom-in operations by randomly selecting areas in the grid layouts. The other two baselines are DendroMap [2] and LAS (linear assignment sorting) [1], the SOTA direct mapping grid layout methods. DendroMap [2] displays samples in a treemap. As only the treemap nodes can be selected for zooming in DendroMap, we simulated the zoom-in operations by randomly selecting treemap nodes instead of areas. To enable LAS to handle large-scale datasets, we enhance it by incorporating the same hierarchical exploration strategy used in our method, with the distinction that the grid visualizations at each level are generated by LAS. Exemplary zoom-in operations of these methods can be found in the supplemental material.

For Zhou *et al.*'s method, as it is able to optimize both the triple ratio and perimeter ratio convexity measures, we compared our method to it under both measures. DentroMap tends to generate grid layouts with horizontal or vertical boundaries. Therefore, we only compare our method to it with the perimeter ratio convexity measure, which also tends to generate such boundaries. LAS tends to generate grid layouts with slanted boundaries. Thus, we compare it to our method with the triple ratio convexity measure, which is more suited to layouts with slanted boundaries. To evaluate the effects of grid sizes, the comparison is conducted under three different grid sizes: 30×30, 40×40, and 50×50.

Evaluation measures. As enhancing cluster perception (*i.e.*, compactness, convexity, and proximity), clarifying ambiguity, and preserving stability are important for hierarchical fuzzy-cluster-aware grid visualizations [3], [61], we evaluated the results from these perspectives.

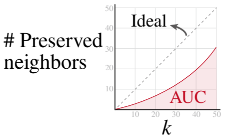
Compactness. According to Rottmann *et al.* [43], compactness is quantified by calculating the average squared distances between the grid cells of samples and their cluster centers:

$$\text{Comp.} = \sum_{i=1}^{N} \frac{||g(s_i) - c_i||^2}{N},$$

where $g(s_i)$ and c_i are the associated grid cell and cluster center of the i-th sample, respectively.

Convexity. The two representative convexity measures, the triple ratio and perimeter ratio, are used for evaluation [61]. They have been introduced in Sec. IV-B1. As these two measures conflict with each other to some extent, we evaluated the grid layout results under them separately.

Proximity. DendroMap [2] evaluates the proximity by the k-nearest neighbor (kNN) preservation plot. It shows how well the kNNs in the high-dimensional space are preserved in the grid layouts under different k values. Inspired by the AUROC [21], we utilized the areas under the curves (AUC) in the plots as the quantitative measure to ensure quantitative comparability and robustness to the choice of k. We limited the maximum value of k to 50 to balance effectiveness and computational efficiency.



Stability. For *cluster-level stability*, it is crucial to preserve the shapes and relative positions of clusters. Therefore, two measures are considered: cluster shape stability (Stab-shape) and cluster position stability (Stab-position). For the cluster shape stability, we utilized the Intersect over Union (IoU) between the previous and current shapes of clusters as the evaluation measure. We chose the IoU because it is widely used for measuring the similarity between two shapes [62]. As the positions and scales of the clusters change during zooming in, we rescaled the previous and current shapes and aligned their centers before calculating the IoU values. For the cluster position stability, as the absolute positions of clusters change during zooming in, we utilize the changes of relative positions to measure the position stability:

$$\text{Stab-position} = \sum_{i}^{|Z|} \sum_{j}^{|Z|} ||\overline{c_{z_i} c_{z_j}} - \overline{c'_{z_i} c'_{z_j}}||^2 \frac{N_{z_i} N_{z_j}}{N^2},$$

where c'_i and c_i are the previous and current centers of the i-th cluster, respectively. $\overline{c_i c_j}$ is a vector from c_i to c_j. $Z = \{z_1, z_2, ..., z_{|Z|}\}$ are the indices of the clusters that appear both in the previous and current grid layouts. N_i is the sample number in the i-th cluster, and N is the total sample number.

For *sample-level stability* (Stab-sample), we utilized the commonly-used orthogonal mental distance [10]. It measures how likely the orders of pairs of cells are changed. The formal definition is given in Sec. IV-B2.

Ambiguity. To accurately measure ambiguity preservation, two perspectives need to be considered. If two clusters are adjacent, we evaluated the ambiguity preservation by the products between the membership degrees of ambiguous samples and their squared distances to the corresponding cluster boundaries, the same as the sample assignment step in Sec. IV-B2. If two clusters are not adjacent, the number of ambiguous samples between them should be small. Therefore, we punish such results by using the products between the membership degrees of ambiguous samples and a large value. In our implementation, we set this large value for an ambiguous sample as the largest squared distance between cells in its associated cluster. We use the prediction probability distribution as the membership degrees of a sample to different clusters. The samples whose ambiguity degrees are larger than 0.2 are treated as ambiguous samples in our evaluation.

Comparison Results. The quantitative experiment results are shown in Tabs. II and III. The results are averaged over $1,350$ trials (150 simulated zoom-in operations $\times$ 3 grid sizes $\times$ 3 datasets). The detailed results are in supplemental material.

TABLE II
COMPARISON OF ZHOU *et al.*'S METHOD AND OUR METHOD. ↑ (↓) INDICATES THE HIGHER (LOWER) IS BETTER.

Conv. type	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
Triple ratio	Zhou *et al.*'s method	0.020	0.996	462.1	0.65	0.214	695.9	6.06
	Ours	**0.019**	**0.997**	506.0	**0.93**	**0.100**	**3.2**	**4.41**
Perimeter ratio	Zhou *et al.*'s method	0.025	0.918	442.0	0.55	0.229	703.1	7.54
	Ours	**0.022**	**0.965**	**501.8**	**0.90**	**0.115**	**3.7**	**4.56**

TABLE III
COMPARISON OF DENDROMAP AND OUR METHOD. ↑ (↓) INDICATES THE HIGHER (LOWER) IS BETTER.

Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
DendroMap	0.109	0.536	366.9	0.34	0.323	1560.8	11.34
Ours	**0.029**	**0.959**	**518.6**	**0.89**	**0.109**	**8.9**	**4.25**

TABLE IV
COMPARISON OF LAS AND OUR METHOD. ↑ (↓) INDICATES THE HIGHER (LOWER) IS BETTER.

Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
LAS	0.058	0.603	545.2	0.34	0.560	1452.8	57.87
Ours	**0.019**	**0.997**	506.0	**0.93**	**0.100**	**3.2**	**4.41**
Ours with proximity only	0.058	0.589	**545.8**	0.33	0.479	1433.4	57.42

Tab. II shows the quantitative comparison between Zhou *et al.*'s method and our method. To explain their difference more intuitively, we also visually compared their results on three datasets in Fig. 6. From Tab. II and Fig. 6, we identified the following observations.

Compactness. Regarding the compactness, our method is slightly better than that of Zhou *et al.*'s method. This indicates that our method does not sacrifice compactness preservation to achieve ambiguity clarification and stability preservation.

Convexity. For the triple ratio measure, the convexity scores of both methods are very close. Our method improves the convexity score by a margin for the perimeter ratio measure. This is because our method tends to generate clusters that are closer to rectangles than Zhou *et al.*'s method (*e.g.*, Figs. 6G and 6H), which has high perimeter ratios.

Proximity. Our method performs better than Zhou *et al.*'s method in terms of proximity measure. The main reason is that Zhou *et al.*'s method preserves proximity by first projecting samples to a set of two-dimensional points with dimension reduction methods. Then, these two-dimensional points are assigned to grid cells. Both the projection and assignment introduce distortions. Our method projects samples to grid layouts directly via the quadratic assignment, which reduces the distortions and thus improves the proximity.

Cluster-level stability. Compared with Zhou *et al.*'s method, our method performs better in terms of cluster-level stability measures. For example, in Fig. 6D, the purple cluster is in the top-left corner within the selected area. However, Zhou *et al.*'s method places it in the middle left after zooming (Fig. 6E). On the contrary, our method still places this cluster in the top-left corner (Fig. 6F). In addition, the shapes of clusters are better preserved by our method (*e.g.*, Fig. 6L) than Zhou *et al.*'s method (*e.g.*, Fig. 6K).

Sample-level stability. Regarding the sample-level stability measure, our method performs better than Zhou *et al.*'s method. To better visually compare the preservation of the sample-level stability, we selected one cluster (Fig. 6A) and followed the work of Han *et al.* [20] to re-assign different colors to cells according to their positions. Then, we apply the color scheme to the new grid layouts after zooming in. As shown in Figs. 6B and 6C, the colored grid layout generated by our method is more similar to the previous one than that generated by Zhou *et al.*'s method. It shows that the relative positions of samples are better preserved by our method.

Ambiguity. Our method also clarifies ambiguity better than Zhou *et al.*'s method. For example, the ambiguous samples scatter in the grid layout generated by Zhou *et al.*'s method (*e.g.*, Fig. 6I). However, our method places them closer to the boundaries (*e.g.*, Fig. 6J).

Tabs. III and IV show the comparison with DendroMap and LAS. Consistently, our method performs better than them in all measures except that our method performs worse than LAS in proximity. This is because LAS primarily focuses on optimizing proximity, whereas our method balances multiple objectives, including convexity, compactness, stability, and ambiguity. This multifaceted optimization sometimes results in sub-optimal proximity. To validate this, we conducted an additional comparison where our method focuses solely on optimizing proximity. As shown in Table IV, our method slightly performs better than LAS in the proximity measure, demonstrating the effectiveness of our optimization method.

Running Time. To demonstrate that our method can meet the interactive requirements, we conducted an experiment in a desktop PC equipped with an Intel i9-13900K CPU. In addition to the previously used grid sizes (30×30, 40×40, and 50×50), we also assessed the running time in generating larger grid sizes (60×60, 80×80, and 100×100) to provide a more comprehensive evaluation.

As shown in Tab. V, our method can generate a 50×50 grid layout around 1 second. If the number of samples is larger than 50×50, our method can build a hierarchy for them and ensure no more than 50×50 samples are displayed

TABLE V
THE RUNNING TIME (IN SECONDS) OF OUR METHOD UNDER DIFFERENT GRID SIZES.

Conv. type	Method	30 × 30	40 × 40	50 × 50	60 × 60	80 × 80	100 × 100
Triple ratio	Ours	0.51	0.82	1.38	1.75	3.92	8.08
Perimeter ratio	Ours	0.39	0.65	1.27	1.58	3.61	7.49

at each level. In this way, our method can support real-time interactive exploration of large-scale datasets.

B. Use Case

We conducted a use case to demonstrate how ambiguity clarification and stability preservation help analyze model performance. We invited E_1 to analyze a ViT-B model [11] trained on the ImageNet-1k dataset. The training accuracy was 85.62%. E_1 wondered why the ViT-B model could not fit the training data well. Therefore, he used our method to analyze the model predictions on the training data. The samples with the same predicted class were treated as a cluster, and colors were used to encode predicted classes. Since deep learning models often underestimate the ambiguity degrees of samples [17], E_1 set the ambiguity degree threshold as a relatively low value of 0.2. E_1 began his analysis on the sub-hierarchy "animal" because he was familiar with this sub-hierarchy.

Identifying confusion between clusters. In the grid layout, E_1 found many ambiguous samples (Fig. 7(a)). Seeking to understand the underlying causes, he selected an area with the most ambiguous samples (Fig. 7A) and zoomed in for closer inspection. Upon zooming in (Fig. 7(b)), only two clusters were left: "hunting dog" (brown) and "non-hunting dog" (yellow). E_1 observed that all ambiguous samples were near the boundaries, which helped him identify and analyze them easily. For instance, in Fig. 7D, ambiguous samples of hunting and non-hunting dogs are positioned closely. Examining their associated images (Figs. 7C and 7E), E_1 noted that these dogs, such as the Lhasa Apso (hunting dogs) and Shih-Tzu dogs (non-hunting dogs), have similar appearances characterized by dense, long hairs. These features made them challenging to distinguish. This explains why confusion arose in this area. Conversely, with Zhou *et al.*'s method, these non-hunting dogs, such as the Shih-Tzu dogs, were positioned away from the boundaries of their cluster (Fig. 7F). Moreover, many of their neighbors were Boxer dogs (*e.g.*, Fig. 7G). This would mislead users to conclude that Shih-Tzu and Boxer dogs are easily confused. However, this is not correct as Boxer dogs do not have long hairs, making them easily distinguishable from Shih-Tzu dogs.

Identifying misclassifications within clusters. E_1 also wondered about the confusion within the displayed clusters. Therefore, he selected an area without ambiguous samples (Fig. 7B) to zoom in and expanded the clusters. After zooming, E_1 examined the representative images and found an interesting one

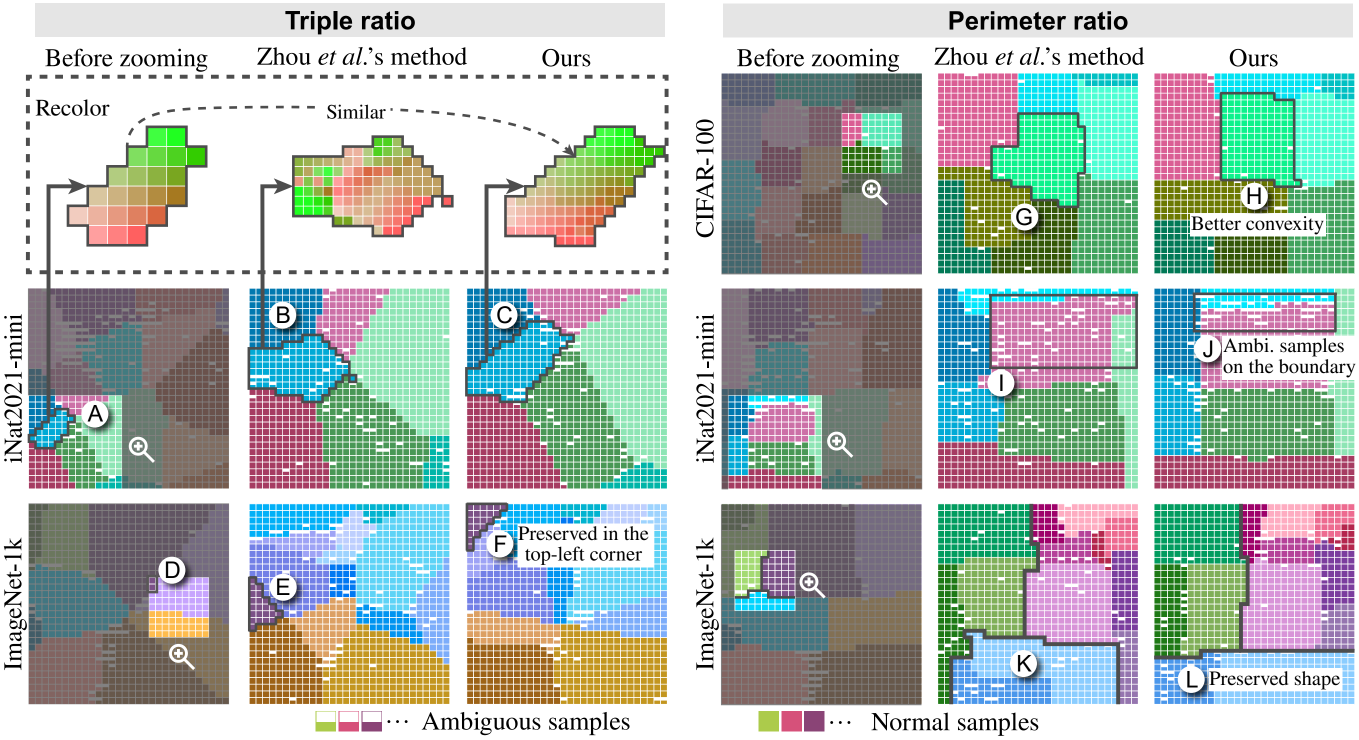


Fig. 6. Grid visualizations generated by Zhou *et al.*'s method and our method on different datasets.

(Fig. 1A). This is an image of a tiger, but it was classified as a domestic cat. Wondering why this image was misclassified, E_1 selected an area around this image to further zoom in (Fig. 1B) and expanded the cluster “domestic cat” into three sub-clusters. As the misclassified image was in the middle-left part of the selected area, E_1 first checked the middle-left part of the new grid visualization. He immediately identified more misclassified tigers (Fig. 1D). By checking their ground truth, E_1 found that these images were mislabeled as tiger cats because both of these two breeds had black stripes on their bodies. This led to the misclassifications. On the contrary, by checking the middle-left part of the grid layout generated by Zhou *et al.*’s method, only several correctly classified tiger cats were identified (Fig. 1C). This hindered the identification of the root causes of the misclassifications.

VI. Expert Feedback and Discussion

After the use case, we conducted interviews with four machine learning experts to collect feedback, including E_1 and three newly invited ones (E_5, E_6, and E_7). We introduced the use case to the new experts and allowed them to independently explore the tool before the interviews. The expert feedback on the usability of our method was generally positive. They also identified several limitations, which highlighted areas for further research and development.

A. Usability

Efficient exploration of large-scale datasets. All the experts appreciated the efficient exploration of large-scale datasets supported by our method. “It is impressive to explore hundreds of thousands of samples with this tool. Currently, to find samples of interest in large-scale datasets, I have to write one-off scripts to process the data. This is much more tedious and inefficient than using this tool.” E_5 commented. E_6 was impressed by the hierarchical exploration environment supported by our method. He said, “It enables me to have an overview of the data first. Then I can examine the areas of interest, such as those with many ambiguous samples, for more detailed analysis.”

Facilitating inferences on model performance. E_7 noted the clarification of ambiguous samples near the boundaries. He commented that this reduced false inferences on model performance. In his current practice, E_7 used t-SNE [35] and grid layouts [6], [7] to examine the model predictions on samples, especially ambiguous samples. However, he often found that many ambiguous samples were placed near the centers of their associated clusters. This hindered him from identifying which classes these ambiguous samples were confused with. By placing ambiguous samples near the boundaries, such classes can be quickly identified to facilitate the analysis of root causes of ambiguous samples (*e.g.*, Figs. 7C and 7E). E_6 also shared one experience: although some classes might have ambiguous samples but were not adjacent, users could analyze their ambiguous samples by filtering out other irrelevant classes.

Easy to track samples. The experts also acknowledged the capacity of our method to preserve stability. E_1, who had used Zhou *et al.*’s method, especially liked this stability preservation. When using Zhou *et al.*’s method, E_1 often found that the stability of samples is not preserved when zooming in. This usually led to additional efforts to find the samples of interest he identified before zooming in. By preserving stability, our method reduces these additional efforts. “For example, when I zoomed in the area with a misclassified tiger, I quickly identified many misclassified tigers with similar patterns in this area. This saved me a lot of time and efforts.” He said.

B. Limitations and Future Work

Generalization to multi-modal data. In the current implementation, our method primarily analyzes image data. However, multi-modal data, such as meteorological data, is more

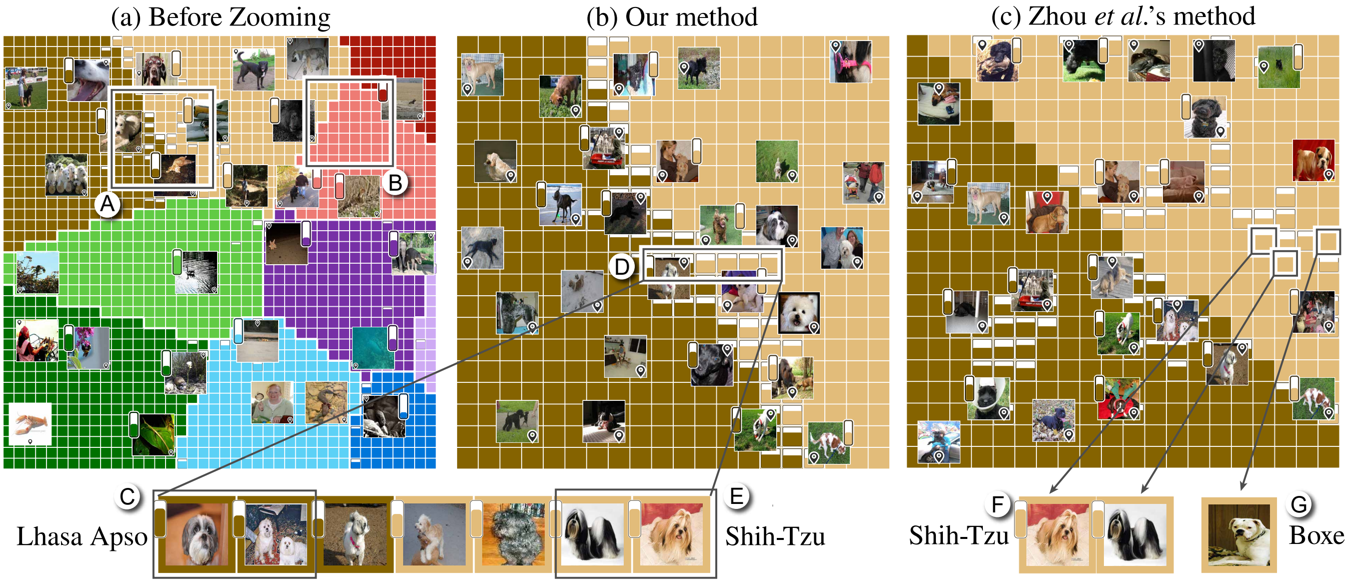


Fig. 7. The grid visualizations (a) before zooming; (b) generated by our method; (c) generated by Zhou *et al.*’s method.

commonly used in real-world applications. Effective analysis of such data requires not only examining their content but also understanding the relationships between different modalities. For example, understanding the relationships between precipitation data and cloud radar data can elucidate the mechanisms driving precipitation processes [41]. However, integrating different modalities into a single grid layout while maintaining their interrelationships poses significant challenges. For example, when projecting images and the labels of objects in images, directly using existing grid layout methods treats images and object labels equally. This causes the object names to collapse together because object labels are more similar to each other than the images. Moreover, since the images have fixed sizes while the labels vary in size, arranging them compactly on a 2D plane becomes an NP-hard packing problem, which is also non-trivial. Thus, investigating how to adapt our method to accommodate multi-modal data is a promising future research direction.

Formal user studies on usability. Although our method is developed in collaboration with machine learning experts, its applicability extends beyond the domain of machine learning because it is agnostic to the definition of clusters and ambiguous samples. For example, in graph analysis, communities within a graph can be treated as clusters, while nodes that belong to multiple communities are analogous to ambiguous samples. Our method can be applied to analyze both communities and ambiguous samples in such domains. However, in our current evaluation, we only invited experts from the field of machine learning to evaluate the effectiveness of our method. It may limit the validity of our findings as the machine learning experts do not fully represent the users from other domains. Hence, it would be valuable for future research to conduct user studies with a more diverse set of users from different domains for evaluating the effectiveness and efficiency of our method.

VII. Conclusion

In this paper, we present a hierarchical fuzzy-cluster-aware grid layout method for hierarchical exploration of large-scale datasets. The key feature of our method is a two-step optimization strategy to facilitate fuzzy cluster analysis while maintaining visual continuity for users during the exploration. This strategy consists of a cluster-aware partition method to enhance cluster perception while maintaining cluster-level stability, and a sample assignment method to maintain sample-level stability and place ambiguous samples near the boundaries. The effectiveness of our method is demonstrated by its better quantitative results compared with the baselines, its utility in model debugging showcased through the use case, and the positive feedback from machine learning experts.

References

[1] 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] 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] S. Bridgeman and R. Tamassia. Difference metrics for interactive orthogonal graph drawing algorithms. In *International Symposium on Graph Drawing*, pp. 57–71, 1998. doi: 10.1007/3-540-37623-2_5

[4] P. Cavanagh. Visual cognition. *Vision Research*, 51(13):1538–1551, 2011. doi: 10.1016/j.visres.2011.01.015

[5] C. Chen, J. Chen, W. Yang, H. Wang, J. Knittel, X. Zhao, S. Koch, T. Ertl, and S. Liu. Enhancing single-frame supervision for better temporal action localization. *IEEE Transactions on Visualization and Computer Graphics*, 30(6):2903–2915, 2024. doi: 10.1109/TVCG.2024.3388521

[6] 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

[7] 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

[8] J. Chen, W. Yang, Z. Jia, L. Xiao, and S. Liu. Dynamic color assignment for hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[9] M. Das Gupta, S. Srinivasa, M. Antony, et al. Kl divergence based agglomerative clustering for automated vitiligo grading. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2700–2709, 2015. doi: 10.1109/cvpr.2015.7298886

[10] S. Diehl and C. Görg. Graphs, they are changing. In *Graph Drawing*, pp. 23–31, 2002. doi: 10.1007/3-540-36151-0_3

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[12] 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

[13] H. Fang, S. Walton, E. Delahaye, J. Harris, D. A. Storchak, and M. Chen. Categorical colormap optimization with visualization case studies. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):871–880, 2017. doi: 10.1109/TVCG.2016.2599214

[14] 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

[15] 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

[16] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. doi: 10.1002/spe.4380211102

[17] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the International Conference on Machine Learning*, pp. 1321–1330, 2017.

[18] D. Guo. Flow mapping and multivariate visualization of large spatial interaction data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1041–1048, 2009.

[19] 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

[20] C. Han, J. Jo, A. Li, B. Lee, O. Deussen, and Y. Wang. SizePairs: Achieving stable and balanced temporal treemaps using hierarchical size-based pairing. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):193–202, 2022. doi: 10.1109/tvcg.2022.3209450

[21] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001. doi: 10.1023/A:1010920819831

[22] W. He, L. Zou, A. K. Shekar, L. Gou, and L. Ren. Where can we help? a visual analytics approach to diagnosing and improving semantic segmentation of movable objects. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1040–1050, 2021. doi: 10.1109/TVCG.2021.3114855

[23] G. M. Hilasaca, W. E. Marcílio-Jr, D. M. Eler, R. M. Martins, and F. V. Paulovich. A grid-based method for removing overlaps of dimensionality reduction scatterplot layouts. *IEEE Transactions on Visualization and Computer Graphics*, 2024. doi: 10.1109/tvcg.2023.3309941

[24] F. Hohman, H. Park, C. Robinson, and D. H. P. Chau. Summit: Scaling deep learning interpretability by visualizing activation and attribution

summarizations. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1096–1106, 2019. doi: 10.1109/tvcg.2019.2934659

[25] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15, 1989. doi: 10.1016/0020-0190(89)90102-6

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

[27] R. M. Karp. *Reducibility among combinatorial problems*. Springer, 2010. doi: 10.1007/978-3-540-68279-0_8

[28] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[29] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, pp. 53–76, 1957. doi: 10.2307/1907742

[30] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[31] D. Li, X. Guo, X. Shu, L. Xiao, L. Yu, and S. Liu. RouteFlow: Trajectory-aware animated transitions. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, 2025. doi: 10.1145/3706598.3714300

[32] S. Liu, C. Chen, Y. Lu, F. Ouyang, and B. Wang. An interactive method to improve crowdsourced annotations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):235–245, 2019. doi: 10.1109/tvcg.2018.2864843

[33] S. Liu, W. Yang, J. Wang, and J. Yuan. *Visualization for Artificial Intelligence*. Synthesis Lectures on Visualization. Springer Cham, 2025. doi: 10.1007/978-3-031-75340-4

[34] 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

[35] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[36] L. Micallef, G. Palmas, A. Oulasvirta, and T. Weinkauf. Towards perceptual optimization of the visual design of scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1588–1599, 2017. doi: 10.1109/tvcg.2017.2674978

[37] B. E. Moore and J. J. Corso. FiftyOne. *GitHub. Note: https://github.com/voxel51/fiftyone*, 2020.

[38] F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012. doi: 10.1002/widm.53

[39] S. E. Palmer. Common region: A new principle of perceptual grouping. *Cognitive Psychology*, 24(3):436–447, 1992. doi: 10.1016/0010-0285(92)90014-S

[40] N. Quadrianto, A. J. Smola, L. Song, and T. Tuytelaars. Kernelized sorting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1809–1821, 2010. doi: 10.1109/TPAMI.2009.184

[41] D. Rosenfeld, M. O. Andreae, A. Asmi, M. Chin, G. de Leeuw, D. P. Donovan, R. Kahn, S. Kinne, N. Kivekäs, M. Kulmala, et al. Global observations of aerosol-cloud-precipitation-climate interactions. *Reviews of Geophysics*, 52(4):750–808, 2014. doi: 10.1002/2013rg000441

[42] 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

[43] 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

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y

[45] D. Sacha, M. Kraus, J. Bernard, M. Behrisch, T. Schreck, Y. Asano, and D. A. Keim. SOMFlow: Guided exploratory cluster analysis with self-organizing maps and analytic provenance. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):120–130, 2017. doi: 10.1109/tvcg.2017.2744805

[46] W. Scheibel, D. Limberger, and J. Döllner. Survey of treemap layout algorithms. In *Proceedings of the International Symposium on Visual Information Communication and Interaction*, pp. 1–9, 2020. doi: 10.1145/3430036.3430041

[47] T. Schreck, J. Bernard, T. Von Landesberger, and J. Kohlhammer. Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization*, 8(1):14–29, 2009. doi: 10.1109/vast.2008.4677350

[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

[49] 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

[50] T. Tian, J. Zhu, and Y. Qiaoben. Max-margin majority voting for learning from crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10):2480–2494, 2019. doi: 10.1109/tpami.2018.2860987

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

[52] C. Tseng, G. J. Quadri, Z. Wang, and D. A. Szafir. Measuring categorical perception in color-coded scatterplots. In *proceedings of the CHI conference on human factors in computing systems*, pp. 1–14, 2023. doi: 10.1145/3544548.3581416

[53] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):1–10, 2016. doi: 10.1109/TVCG.2015.2468078

[54] G. Van Horn, E. Cole, S. Beery, K. Wilber, S. Belongie, and O. Mac Aodha. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12884–12893, 2021. doi: 10.1109/cvpr46437.2021.01269

[55] J. T. Vogelstein, J. M. Conroy, V. Lyzinski, L. J. Podrazik, S. G. Kratzer, E. T. Harley, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe. Fast approximate quadratic programming for graph matching. *PLOS One*, 10(4):e0121002, 2015. doi: 10.1371/journal.pone.0121002

[56] S.-Q. Xin, B. Lévy, Z. Chen, L. Chu, Y. Yu, C. Tu, and W. Wang. Centroidal power diagrams with capacity constraints: Computation, applications, and extension. *ACM Transactions on Graphics*, 35(6):1–12, 2016. doi: 10.1145/2980179.2982428

[57] K. S. Xu, M. Kliger, and A. O. Hero. A regularized graph layout framework for dynamic network visualization. *Data Mining and Knowledge Discovery*, 27:84–116, 2013. doi: 10.1007/s10618-012-0286-6

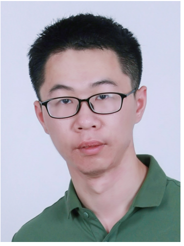
[58] Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, and R. Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pp. 11525–11536, 2021.

[59] W. Yang, M. Liu, Z. Wang, and S. Liu. Foundation models meet visualizations: Challenges and opportunities. *Computational Visual Media*, 10(3):399–424, 2024. doi: 10.1007/s41095-023-0393-x

[60] 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

[61] Y. Zhou, W. Yang, J. Chen, C. Chen, Z. Shen, X. Luo, L. Yu, and S. Liu. Cluster-aware grid layout. *IEEE Transactions on Visualization and Computer Graphics*, 2023. doi: 10.1109/tvcg.2023.3326934

[62] J. Žunić and P. L. Rosin. Measuring shapes with desired convex polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6):1394–1407, 2019. doi: 10.1109/tpami.2019.2898830



Yuxing Zhou is a second-year master's student at the School of Software, Tsinghua University. His research interest is layout algorithms. He received a B.S. degree from Tsinghua University.

Changjian Chen is an assistant professor at Hunan University. He received a Ph.D. from Tsinghua University and a B.S. from University of Science and Technology of China. His research interests focus on visual analytics and machine learning, especially visual analysis methods to improve training data quality.

Zhiyang Shen is an undergraduate student at the School of Software at Tsinghua University. His research interests include explainable machine learning and visualization.

Jiangning Zhu is a second-year Ph.D. student at the School of Software, Tsinghua University. His research interest is explainable artificial intelligence. He received a B.S. degree from Tsinghua University.

Jiashu Chen is a second-year master's student at the School of Software, Tsinghua University. His research interests lie in perception and machine learning. He received a B.S. degree from Tsinghua University.

Weikai Yang is an assistant professor in Hong Kong University of Science and Technology (Guangzhou). His research interests lie in visual analytics, machine learning, and data quality improvement. He received a B.S. and a Ph.D from Tsinghua University.

Shixia Liu is a professor at Tsinghua University. Her research interests include explainable artificial intelligence, visual analytics for big data. She worked as a research staff member at IBM China Research Lab and a lead researcher at Microsoft Research Asia. She received a B.S. and M.S. from Harbin Institute of Technology, a Ph.D. from Tsinghua University. She is a fellow of IEEE and an associate editor-in-chief of IEEE Trans. Vis. Comput. Graph.