

Multilevel Visual Analysis of Aggregate Geo-Networks

Zikun Deng, Shifu Chen, Xiao Xie, Guodao Sun, Mingliang Xu, Di Weng, Yingcai Wu

Abstract—Numerous patterns found in urban phenomena, such as air pollution and human mobility, can be characterized as many directed geospatial networks (geo-networks) that represent spreading processes in urban space. These geo-networks can be analyzed from multiple levels, ranging from the *macro*-level of summarizing all geo-networks, *meso*-level of comparing or summarizing parts of geo-networks, and *micro*-level of inspecting individual geo-networks. Most of the existing visualizations cannot support multilevel analysis well. These techniques work by: 1) showing geo-networks separately with multiple maps leads to heavy context switching costs between different maps; 2) summarizing all geo-networks into a single network can lead to the loss of individual information; 3) drawing all geo-networks onto one map might suffer from the visual scalability issue in distinguishing individual geo-networks. In this study, we propose *GeoNetverse*, a novel visualization technique for analyzing aggregate geo-networks from multiple levels. Inspired by metro maps, *GeoNetverse* balances the overview and details of the geo-networks by placing the edges shared between geo-networks in a stacked manner. To enhance the visual scalability, *GeoNetverse* incorporates a level-of-detail rendering, a progressive crossing minimization, and a coloring technique. A set of evaluations was conducted to evaluate *GeoNetverse* from multiple perspectives.

Index Terms—Geospatial network, multilevel analysis, information visualization, graph drawing

1 INTRODUCTION

Many urban phenomena can be modeled as geospatial networks (geo-networks), such as air pollution propagation [13], [35], traffic congestion cascades and propagation [14], [36], human mobility [1], [27], [61], and causal relations [17]. A geo-network (Fig. 1A) characterizes how an urban phenomenon or physical objects spread (i.e., edges) over geographic locations (i.e., nodes), for example, the propagation of air pollutants over air quality sensors. Visualizing numerous geo-networks extracted from urban phenomena can help urban experts intuitively perform urban diagnosis in the geospatial context.

We conduct a literature review and identify three levels of analyzing multiple geo-networks. At the *macro level*, experts summarize *all* geo-networks to learn the overall picture of the whole dataset [3], [14], [61]. At the *meso level*, experts compare or summarize *parts* of geo-networks of their interests [13], [41], [63]. At the *micro level*, experts inspect *individual* geo-networks [1], [13], [17]. From here on, “individual” refers to an individual geo-network. In many scenarios, experts may need to constantly switch among multiple levels of analyses. At each level, they may need

to perform topology-based, attribute-based, and browsing tasks [34]. Take the analysis of air pollution propagation as an example. First, experts need to know where propagation frequently occurs (*macro level*) and zoom in to learn how it spreads there (*micro level*). If a location is found polluted, experts may want to see all geo-networks that involve this location and compare them to find the source of the pollution (*meso level*).

Existing techniques cannot effectively support such a multilevel analysis. The existing techniques can be categorized into the following three types. (1) *Multiple geo-networks on multiple maps* techniques (Fig. 1B) like small multiples [63] or lists [13] enable the micro-level analysis but demand frequent context switching for the macro-level summarization. (2) *One geo-network on one map* techniques aggregate geo-networks into one on a map [1], [3], [61] (e.g., Fig. 1C). Yet, the aggregated geo-network cannot be decomposed for micro and meso level analyses. (3) *Multiple geo-networks on one map* techniques plot all geo-networks on one map in an edge bundling [29] or stacking way [69]. The bundled edges make each geo-network hard to be identified. By contrast, the stacked edges support multilevel analysis because both the individuals and aggregation are visible. Nonetheless, shown in Fig. 1D, the edge stacking way is not scalable to multiple geo-networks that have many edges in common, i.e., shared edges. Elzen and Wijk [59] combined the second and third types. Yet, their method can summarize only a part of geo-networks selected by users via aggregation.

Motivated by the limitations of existing techniques, we propose *GeoNetverse*, a new *multiple-network-on-one-map* technique for multilevel analysis of multiple geo-networks. We follow the idea of metro-maps to place the edges shared between geo-networks in a stacked manner, such that these geo-networks can be analyzed from multiple levels within the same and coherent geographic context. Two challenges

- Z. Deng, S. Chen and Y. Wu are with State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. E-mail: {zikun_rain, sfchen, ycwu}@zju.edu.cn.
- X. Xiao is with Department of Sport Science, Zhejiang University, Hangzhou, China. E-mail: xxie@zju.edu.cn.
- G. Sun is with College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. E-mail: guodao@zjut.edu.cn.
- M. Xu is with School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou, China and Engineering Research Center of Intelligent Swarm Systems, MOE of China National Supercomputing Center in Zhengzhou, Zhengzhou, China. E-mail: iexumingliang@zzu.edu.cn.
- D. Weng is with Microsoft Research Asia. E-mail: mystery.wd@gmail.com. Y. Wu and D. Weng are the co-corresponding authors.

Manuscript received April 19, 2005; revised August 26, 2015.

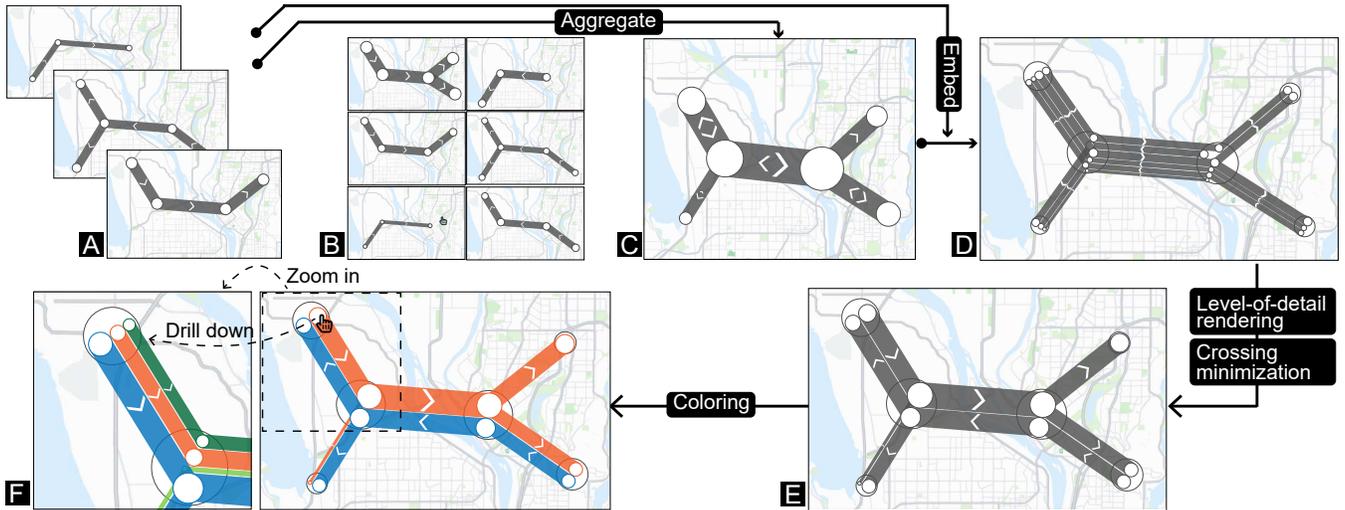


Fig. 1: Multiple geo-networks in (A) can be visualized with three types of visualization, namely, (B) multiple geo-networks on multiple maps, (C) one geo-network on one map, and (D) multiple geo-networks on one map. (E) A level-of-detail rendering and crossing minimization are incorporated to address the low legibility in (D). (F) GeoNetverse is proposed after coloring the geo-networks in (E).

need to be solved to improve the visual scalability:

Numerous shared edges. Metro-lines share much fewer edges than geo-networks. Due to the limited screen pixels, the legibility of each stacked edge will considerably suffer in the visualization of geo-networks (Fig. 1D). To accommodate the numerous edges when the map’s zoom level is low, a level-of-detail rendering technique should be developed for geo-network visualization, such that the edges can be merged and summarized gradually as the zoom level is decreased. Moreover, the line wiring techniques [7] for reducing edge crossings in metro-maps need to be adapted for geo-network visualization to maintain the edges’ positions across different zoom levels.

Discrimination of networks. Geo-networks should be easily distinguished, such that their topologies will not be confused with each other. Coloring these geo-networks can ensure the discrimination. However, the coloring of metro-lines is mostly manually determined [47], [69]. Existing automatic coloring methods [32] cannot color the geo-networks with limited distinguishable colors, because their edges are stacked and cross each other at each level of detail. Moreover, the coloring between the levels should be maintained consistently for smooth drill-down analysis.

We propose GeoNetverse to address the above challenges. For the first challenge, we develop a level-of-detail rendering technique for geo-networks based on hierarchical clustering. The clustering of geo-networks generates multiple hierarchies and the geo-networks will be rendered adaptively following these hierarchies. We also develop a two-phase progressive method that arranges geo-networks following the hierarchies. This method minimizes crossings between geo-networks while maintaining the positional consistencies between the levels of detail. Afterward, we obtain the visualization like Fig. 1E. For the second challenge, we formulate a tailored coloring problem and constraints such that existing graph coloring methods can be applied. We first generate a graph that describes the adjacency

relationships and constraints between geo-networks based on the hierarchies and geo-network arrangement results. Given this graph, we adopt a two-phase progressive way that exploits the limited distinguishable colors to color geo-networks. Finally, we derive our final design GeoNetverse like Fig. 1F. GeoNetverse is evaluated with quantitative experiments, three case studies on real-world datasets, and a task-based within-subject user study.

In summary, our contributions are as follows:

- We develop GeoNetverse that visualizes multiple geo-networks on one geographic map. It enables exploring and analyzing aggregate geo-networks on multiple levels within one map without context switching.
- We conduct a set of evaluations to demonstrate our approaches’ effectiveness and utility comprehensively.

2 RELATED WORK

2.1 Urban Visual Analytics

Urban visual analytics has become an important means of improving cities [15]. There are generally two types of applications. On the one hand, visual urban diagnosis facilitates the identification and understanding of complex and dynamic urban-related issues, such as air pollution [13], [71], traffic patterns [57], [63], autonomous driving [26], [31], public safety [18], [75], energy [39], and reachability [21]. On the other hand, visual urban planning enables urban experts to perform reliable and transparent decision-making, such as location selection [37], [64], [67], traffic planning [65], [66], [77], travel planning [16], and epidemic control [72].

Many of the existing urban analysis methods characterize urban data as geo-networks. For example, human movement data can be aggregated as geo-networks for revealing human mobility patterns [2], [61]. Air pollution propagation processes can also be represented as geo-networks [13], [35]. Similar network-based characterizations can be seen in the analyses of traffic congestion propagation [36], [63]

and generalized spatiotemporal cascading [14] and causalities [17]. Given such network representations, analyses can be performed based on the network structures, such as paths and roots. In sum, visualizing geo-networks is particularly important for urban analyses.

2.2 Visualization of Multiple Geo-networks

Network [43], [46] and geo-network visualizations [38], [51] have been widely studied. We focus on the techniques for multiple networks rather than one large complex network [11], [78]. Schöttler et al. [51] categorized geo-network visualizations into geo-mapped, geo-distorted, and geo-abstracted based on the geography representation. Geo-mapped methods are the most intuitive. They map geo-networks onto a geographic map. Geo-distorted and geo-abstracted methods abstract the geographic context to satisfy other analytical requirements [5], [20], [68], [73]. In sum, we focus on the most related part, geo-mapped visualization for multiple geo-networks, and divide them into three types.

Multiple geo-networks on multiple maps. In these visualizations, each geo-network is drawn on a mini-map, constituting small multiples [13], [63] that support micro-level inspection and meso-level side-by-side comparison. Flexible interactions like filtering, ranking, and pining can be equipped to enhance usability. However, users may lack a concise summary of all individuals (macro) and suffer from severe context switch costs when browsing multiple maps.

One geo-network on one map. These methods aggregate geo-networks by their shared edges or time intervals into one geo-network and depict it on a map as a macro summary [1], [3], [10]. The individuals cannot be obtained by decomposing the aggregation, which limits micro- and meso-level analyses. For example, the flow map in [10] aggregates OD flows by their shared edges, where individuals can be estimated since the flows are single-sourced. However, applying such aggregation to multi-sourced OD flows [61] hides individuals behind the aggregation.

Multiple geo-networks on one map. Every individual geo-network is plotted on one map. The edges of geo-networks are usually bundled or stacked to alleviate occlusion or overlaps between geo-networks. With edge bundling techniques [23], [29], [49], [79], the overlapping edges will be bundled, and edges that extend to different nodes are spread out, generating aesthetic representations [48]. Edge stacking [9], [33] is commonly used in metro-maps [47], [69]. Edge-stacked geo-networks are much easier to distinguish from each other than edge-bundled ones with the same interaction, which facilitates micro- and meso-level analyses. Stacked edges, while alleviating edge overlap, reveal the aggregated edge's width to enable the macro level summarization. However, geo-networks in a general dataset may have numerous overlapping edges more than metro-lines. The manual coloring method for metro-lines cannot be applied to distinguish numerous geo-networks, neither.

In 3D environments, the techniques of this type can display multiple geo-networks by stacking them along the z-axis (i.e., the extra dimension compared to 2D space) [3], [58] or using 3D curves [30], [74]. However, in the former way, geo-networks stacked far from the ground are difficult to be visually related back to the geographic context. In

both ways, the visual occlusion and interaction burden in 3D environments are challenging issues [19].

Elzen and Wijk's method [59] is a **hybrid** technique of the second and third types. The shortcomings of each are compensated for each other. One view presents multiple geo-networks on one map. Users can select nodes and group them as one. Another view presents the aggregation of the geo-networks involved in the grouped nodes. It is a bottom-up exploratory method with high scalability but cannot support summarizing all geo-networks to initialize the analyses from an overview.

Existing techniques cannot well support multilevel analysis of multiple geo-networks. We consider the edge stacking way to have great potential and extend it with visual encodings and algorithms to improve its visual scalability.

2.3 Graph Drawing

Unexpected crossings are generated if geo-networks are not appropriately arranged. We discuss crossing minimization techniques for metro-map, storyline, and Sankey diagram.

In **metro-maps**, line wiring algorithms arrange metro-lines to pass stations with the minimum number of crossings [47], which is denoted as a metro-line crossing minimization (MLCM) problem. In the early years, MLCM and its variants were solved by introducing tailored assumptions due to its NP-hardness [22]. Recently, Bast et al. [7] developed a state-of-the-art solution based on Integer Linear Programming (ILP) without any assumptions.

In **storylines** [54], [55], [56], lines that represent characters move from left to right along a timeline to reveal the story's evolution. Two lines are placed adjacently if the characters interact with each other at that time and are separated from each other otherwise. Reducing the crossings of character lines is one of the aims of storyline layout algorithms. Existing techniques can produce highly readable storylines after crossing minimization [4], [40]. They generally demand the underlying network of lines to be topologically sortable [53].

In **Sankey diagrams** (without loss of generality, the Sankey diagrams that flow from left to right), the flows between two horizontal layers may have crossings, which depend on the vertical positions where the flows start and terminate. Ordering the nodes may reduce the crossings. The flows passing layers from left to right constitute a layered network. Algorithms for layered diagrams can be applied [45], [60]. Sugiyama's method [53] is applicable due to the topologically sortable feature. This problem can also be solved using Integer Programming (ILP) [76].

In our scenario, the underlying network of general geo-networks is neither layered nor topologically sortable. Among the aforementioned methods, only ILP [7], [76] can be leveraged. Although ILP can well arrange numerous geo-networks, limited screen space does not allow them to be drawn directly. We develop a level-of-detail rendering and an ILP-based two-phase method to reduce crossings while maintaining geo-networks' position across levels.

3 BACKGROUND

3.1 Data Description

We have a set of *locations* or *geo-nodes* $\mathbf{N} = \{n_1, n_2, \dots\}$ with different geographic positions. Each location n represents a spatial area in geographic space. We also have a set of *geo-networks* \mathbf{G} . Each geo-network $g = (N_g, E_g) \in \mathbf{G}$ is composed of several locations $N_g \subset \mathbf{N}$ and edges $E_g = \{\dots, e_{u,v}, \dots\}$. $e_{u,v}$ is a directed or bi-directed edge between locations n_u and n_v . Geo-networks may share edges. For instance, an edge $e_{u,v}$ is shared by two geo-networks if they both contain the edge $e_{u,v}$. Geo-networks have weights; otherwise, their weights are considered as 1.

3.2 Three Levels of Analyzing Multiple Geo-networks

We review the prior studies for visual analysis of general networks and geo-networks. Afterward, we summarize the three levels of geo-networks exploration and analyses: macro, meso, and micro levels. These three levels correspond to the information-seeking mantra [52]: “overview first, zoom and filter, then details-on-demand.”

MA. The **MA**cro level focuses on *all* geo-networks. Users learn the overall picture of the whole dataset, for example, how different regions connect to each other [3], [14], [61]. Landesberger et al. [61] provided an interesting example, where a “mono-centric structure” of London was identified based on the summarized human mobilities. The overall picture also guides further exploration. In Deng et al.’s work [17], places where bi-directional edges occur frequently were selected. Users unfold the geo-networks that involve these edges for in-depth causal reasoning.

ME. The **ME**so level focuses on *parts* of geo-networks. Users often compare geo-networks when analyzing them [17], [61]. By comparison, users can identify abnormal and interesting ones, such as those with particular wide edges [14], [41] or uncommon topological structures [13]. Comparative analysis can also be performed for edges or nodes. For example, the edges of inflow and outflow should be compared to determine whether a location is a business or residential area [25]. Summarization can also be applied to parts of geo-networks, which is common in hierarchical organization and exploration [13], [14]. Geo-networks with similar structures on a map can also be further perceptually summarized [13], [63]. Relationships between geo-networks (e.g., overlapping and intersecting) can also be discovered.

MI. The **MI**cro level focuses on *individual* geo-networks. Inspecting a geo-network can be seen in many applications [1], [14], [17], [61], [63]. Available details ensure the reliability of analysis and provide in-depth insights. For example, in Deng et al.’s work [13], users can identify and unfold interesting geo-networks to reason about how air pollutants propagate and the propagation results.

At each level of analysis, users may also need to perform general network analysis tasks [34] (e.g., topology-based) and geographic analysis tasks [50] (e.g., compare).

3.3 Problem Formulation and Approaches

As mentioned in Sec. 1, existing techniques developed for visualizing multiple geo-networks can be divided into three types: 1) multiple geo-networks on multiple maps, 2) one

geo-network on one map, and 3) multiple geo-networks on one map. Existing techniques have limitations in supporting the aforementioned multiple levels. Hence, this study aims at filling the gap by developing a new technique.

We investigated how these types can be improved and whether they can be combined to support multilevel analysis. We finally derive two approaches. The first is to co-ordinate the first and second types with user interactions. The aggregation geo-network (the first type) provides summaries and entries for users to explore individuals (**MA**). The displayed individuals (the second type) enable analyzing details of geo-networks (**ME** and **MI**). They complement each other. Although this new method allows the three levels of analysis to a certain extent, the costs of context switching between multiple maps still impose a burden on users. We call it alternative. The second is to improve the scalability of the third type (i.e., multiple geo-networks on one map). In particular, we first extend the visual design of metro-maps and then empower it with a level-of-detail rendering, two-phase crossing minimization, and tailored automatic coloring to handle aggregate geo-networks. Finally, we develop GeoNetverse, a new *multiple-network-on-one-map* technique that supports the aforementioned tasks well without heavy context switching costs.

4 GEONETVERSE

This section introduces the design of GeoNetverse and the algorithms that enhance its visual scalability.

4.1 Basic Design

GeoNetverse is inspired by metro-map designs [47], [69]. In metro-maps, all metro-lines are drawn on the same map according to their passing stop locations. GeoNetverse takes an analogy between metro-lines and individual geo-networks. GeoNetverse has two layers, background and foreground. The background aims at revealing overall distribution of all geo-networks by aggregating all geo-networks, which enables **MA**. The foreground layer comprises individual geo-networks. Geo-networks are embedded into the background in an edge-stacked way like a metro-map, which supports **ME** with a superimposing manner and **MI**.

Background. In the background layer, we depict the aggregation of all individual geo-networks on the map according to their geographic positions with a node-link diagram. We call the aggregation a background geo-network. The weight of the background edge is the sum of weights of the individuals that contain this edge. When determining whether an individual “contains” an edge, the edge direction can be ignored in some scenarios. The size of the background node encodes the sum of weights of individuals that contain this node. Such a background not only provides a visual summary of all geo-networks but also leaves space for embedding individual geo-networks.

Rendering. We attempt to use straight lines for the edges of the background geo-network. Yet, background edges may overlap at their common nodes. These overlaps make the embedded individuals (e.g., the green geo-network in Fig. 2A) hard to read and follow. Fig. 2C illustrates how this example can be better rendered. Thus, we make background edges repelling each other to avoid overlaps.

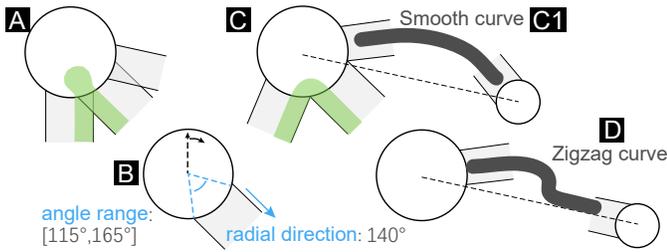


Fig. 2: Routing of background edges. (A) A self-crossing is observed in an embedded geo-network. (B) A node-edge connection is modeled into an angle range and a radial direction. (C) The self-crossing is removed after edges being mutually exclusive. (D) Zigzag curves are undesired.

First, we have to determine whether two edges overlap. For each node-edge connection, we derive 1) the radial direction (i.e., the angle at which the edge enters or leaves the node) and 2) the angle range of the node occupied by the edge, as shown in Fig. 2B. Two edges overlap if the two angle ranges overlap. To avoid overlaps, the direction and then the angle range should be slightly offset. Specifically, we detect the connections that consecutively overlap on a node. We then repel their angle ranges with the slightest offset so that they do not overlap each other (Fig. 2C). After offsetting the direction, the edges can be rendered using the Bezier curve. The other end of a revised edge should also be offset in the opposite direction (Fig. 2C) to avoid zigzag curves (Fig. 2D). The offset operation is performed for every node in descending order in terms of their sizes. Afterward, a solution is obtained if no overlap is detected. Otherwise, we enlarge all nodes to shrink the angle ranges of all connections. Repeat the operation above until a solution is found.

Foreground. In the foreground layer, each geo-network is represented as a node-link diagram. Individual geo-networks are stacked by edges like metro-lines on a metro-map. They have the same weight-width mapping as the aggregation geo-network in the background layer. Recall that the background accumulates the weights of individuals by edges. With the same mapping, the stacked edges of individuals exactly cover the corresponding edges of the background. The stacking orders of geo-networks can be optimized using crossing minimization techniques [7].

Rendering. The edges in the foreground layer follow the background edges. We place arrows inside edges to indicate the edge direction (Fig. 3C). If there are two opposite directions, we encode the respective weights by size and opacity. Inside a background node, the foreground edges and nodes of each individual are drawn according to the rules below. 1) If the individual does not pass through the background node, its node is placed on the periphery of the background node (Fig. 3A). 2) If it passes the background node with two edges, the edges are connected with a quadratic (Fig. 3D) or cubic (Fig. 3E) Bezier curve. The node is placed at the median of the curve. 3) If there are more than two edges (Fig. 3B), the node is placed at the intersection of two edges. The edges are extended to the node straight or using quadratic Bezier curves.

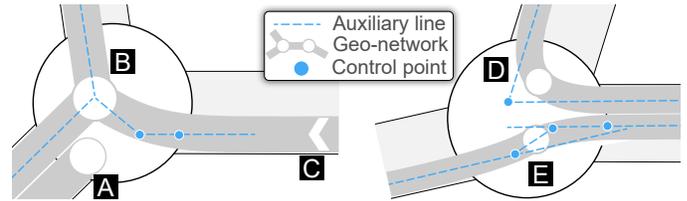


Fig. 3: Routing of foreground edges.

4.2 Enhancement

Individuals and aggregation are encoded together, which avoids frequent context switching. However, the following goals should be satisfied to handle multiple geo-networks.

G1: High legibility of geo-networks. It is not advisable to draw geo-networks on a screen with limited pixels (Fig. 1D). For example, if 20 individual geo-networks contain the same edge, embedding them into the background edge with a width of 40 pixels leads to an average of only 2 pixels per geo-network. In this situation, users suffer from the difficulty of reading the too thin geo-networks. Since pixels are constrained, a feasible solution is to reduce the number of rendered elements. A level-of-detail strategy is demanded to control the rendering and ensure the legibility.

G2: Fewer crossings while maintaining edges' positions across levels. Crossing minimization algorithms are effective for arranging the embedded geo-networks. However, the above level-of-detail strategy poses new constraints for crossing minimization: the position of a lower-level geo-network should be maintained at the positions of its upper-level geo-networks. Such position maintenance underpins smooth drill-down exploration across levels. Thus, a new crossing minimization approach should be developed.

G3: Sufficient discrimination between geo-networks. Embedding geo-networks compactly in a limited space imposes a requirement on their discrimination. Apart from the spatial region, the color hue is the most effective in distinguishing categorical data [44]. Based on the four-color theorem, the vertices of a planar graph can be colored by four colors under the constraint that adjacent elements cannot be colored the same [32]. Since geo-networks can be crossed by each other, their positional relationships may not be described by a planar graph. The number of colors required is likely to exceed the number of distinguishable colors (6 to 12 [44]). Besides, geo-networks in our scenario have four relationships with each other: adjacency, intersecting, co-edge, and hierarchy. Coloring geo-networks that have different relationships using similar colors may lead to different degrees of discrimination. These relationships should be considered with different priorities. A new automated coloring method is required to color geo-networks considering the relationships and limited colors.

GeoNetverse is equipped with automated methods to satisfy these goals. We first adopt a hierarchical clustering to enable level-of-detail rendering [62] (Sec. 4.2.1), which satisfies G1. Based on the hierarchies generated by the clustering, we develop a two-phase progressive algorithm that arranges embedded geo-networks to reduce crossings (Sec. 4.2.2), which satisfies G2. Finally, given the hierarchies and arrangement, we develop a k coloring algorithm

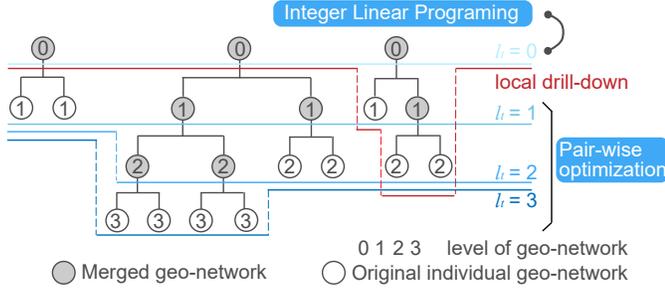


Fig. 4: Three hierarchies generated by hierarchical clustering. Blue lines with the same color indicate the rendered geo-networks given target levels l_t . Red line indicates the rendering with local drill-down. The geo-networks of roots ($l = 0$) are arranged using ILP and others ($l > 0$) are arranged after a pair-wise optimization during a top-down traversal.

that makes rendered geo-networks distinguishable with k available colors (Sec. 4.2.3), which satisfies **G3**. These three algorithms are executed before the interactive visual analysis of geo-networks. Hence, their running does not affect real-time user interaction. Please refer to the appendices for their pseudocode and time complexity analysis.

4.2.1 Level-of-detail Rendering

The goal of level-of-detail rendering is to obtain the proper number of foreground edges on each background edge (**G1**). We adopt a bottom-up strategy to perform the hierarchical clustering that generates multiple levels of geo-networks.

Similarity measure. Inspired by AirVis [13], we introduce a structure-aware similarity measure for geo-networks.

$$d(g_1, g_2) = 1 - \frac{E_{g_1} \cap E_{g_2}}{E_{g_1} \cup E_{g_2}}. \quad (1)$$

To enable hierarchical clustering, we extend the equation (1) and derive final similarity measure $D(*, *)$ as follows:

$$D(g_1, g_2) = \frac{\sum_{g_a \in \mathcal{L}(g_1)} \sum_{g_b \in \mathcal{L}(g_2)} d(g_a, g_b)}{|\mathcal{L}(g_1)| \times |\mathcal{L}(g_2)|}. \quad (2)$$

$\mathcal{L}(g)$ denotes the leaf (original) geo-networks under g in the clustering hierarchies. $\mathcal{L}(g) = \{g\}$ if g is a leaf.

Clustering process. The traditional hierarchical clustering is an iterative procedure that iteratively picks the two most similar geo-networks and merges (aggregates) them as a new geo-network. The new one will replace the original two and participate in the following clustering iterations. In our problem, we aim to reduce the number of geo-networks drawn in an edge by summarizing them. So, we make the following adaptation. In each iteration, we extract a set of candidate geo-networks that passes the background edges where more than m edges have been embedded. The clustering procedure only picks from these candidates. The iteration stops if 1) all similarity measures $D(*, *)$ are lower than a pre-defined threshold λ or 2) there is no candidate. In our study, $m = 5$ and $\lambda = 0.25$ after multiple attempts. These parameters can be easily tuned based on the appearance of the rendering result.

Rendering. The clustering generates multiple hierarchies: for example, three hierarchies in Fig. 4. Each node in a hierarchy represents a geo-network which can be an

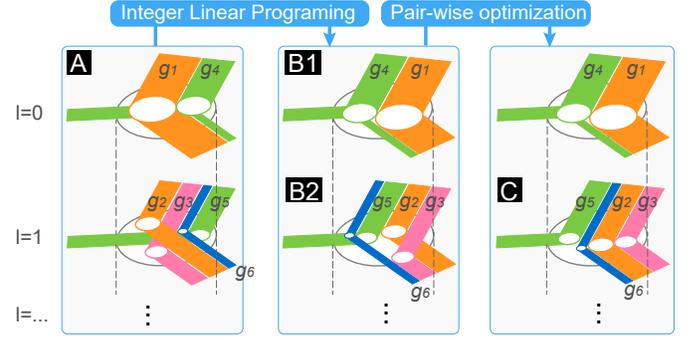


Fig. 5: (A) to (B1): The geo-networks with $l = 0$ (g_1 and g_4) are arranged and have fewest crossings after ILP-based optimization. (B2) to (C): The geo-networks with $l > 0$ are arranged by pair-wise optimization (g_2-g_3 and g_5-g_6) and maintained in their parent’s positions.

original individual geo-network (leaf) or a merged geo-network (non-leaf). The hierarchies enable a level-of-detail rendering. The root nodes of hierarchies have levels $l = 0$. Others can be assigned accordingly in a top-down traversal.

We implement two interactions for controlling the rendering. First, users can specify the target level l_t with a slider to globally determine which geo-networks should be rendered. If a merged or original geo-network has a level $l = l_t$ or has a level $l < l_t$ but has no children, it will be retrieved and rendered. The blue lines in Fig. 4 illustrate the retrieving results with $l_t = 0, 1, 2$, and 3 , respectively. Second, users can perform local drill-down to see details of interests and local summarization. They can click a node of an embedded geo-network to trigger a UI widget with two buttons. On the one hand, they can choose to split the geo-network into two if it is not a leaf. The red line in Fig. 4 illustrates the local drill-down. On the other hand, users can choose to merge the geo-network with its sibling.

Position constraint. Such a rendering strategy enforces constraints of geo-networks’ positions across levels. Take Fig. 5A as an example. g_1 with level $l = 0$ has children g_2 and g_3 with levels $l = 1$. Thus, g_2 and g_3 should take the place g_1 takes to ensure visual consistency during drill-down exploration. In this way, g_2 and g_3 are seen as the results of splitting g_1 . g_1 is seen as the combination of g_2 and g_3 . Crossing minimization should consider the constraints by maintaining edge positions across levels of detail.

4.2.2 Progressive Crossing Minimization

As per the requirement of **G2**, foreground geo-networks should be appropriately arranged to reduce crossings. Researchers have developed effective algorithms for crossing minimization [7], [76]. However, these algorithms cannot be directly applied due to the aforementioned position constraints. We develop a two-phase progressive algorithm for reducing crossings based on the clustering hierarchies.

Phase I: Arrange root geo-networks. We first apply the integer linear programming (ILP) proposed in the prior studies [7], [76] to the root geo-networks retrieved by $l_t = 0$. The optimal arrangement for these geo-networks can be achieved. This phase is illustrated in Fig. 5A and Fig. 5B1.

Here, we illustrate the core idea of the prior methods [7], [76]. There are two types of avoidable crossings, as illustrated in Fig. 6 without loss of generality. For the first type (Figs. 6A1 and A2), g_1 and g_2 that pass through the node n share two edges e_1 and e_2 . A variable $x_{1,2,e} \in \{0, 1\}$ indicates whether the stacking order of g_1 is larger than g_2 on e (1 for true and 0 for false). A decision variable $c_{n,g_1(e_1,e_2),g_2(e_1,e_2)}$ indicates whether g_1 and g_2 have a crossing at the node n . Two constraints are considered:

$$c_{n,g_1(e_1,e_2),g_2(e_1,e_2)} + x_{2,1,e_1} + x_{2,1,e_2} \geq 1, \quad (3)$$

$$c_{n,g_1(e_1,e_2),g_2(e_1,e_2)} + x_{1,2,e_1} + x_{1,2,e_2} \geq 1. \quad (4)$$

In Fig. 6A1, $x_{2,1,e_1}$ and $x_{2,1,e_2}$ are 0. In Fig. 6A2, $x_{1,2,e_1}$ and $x_{1,2,e_2}$ are 0. The decision variable c is enforced to be 1 by the inequalities (3) and (4), respectively. For the second type (Figs. 6B1 and B2), g_1 and g_2 share only one edge. Two constraints are included:

$$c_{n,g_1(e_1,e_3),g_2(e_1,e_2)} + x_{2,1,e_1} + t(e_2, e_3|e_1) \geq 1, \quad (5)$$

$$c_{n,g_1(e_1,e_3),g_2(e_1,e_2)} + x_{1,2,e_1} + t(e_3, e_2|e_1) \geq 1. \quad (6)$$

$t(e_i, e_j|e_k) \in \{0, 1\}$ indicate whether the stacking order of e_i is larger than that of e_j starting from e_k in the clockwise direction. In Fig. 6B1, $x_{2,1,e_1}$ and $t(e_2, e_3|e_1)$ are 0. In Fig. 6B2, $x_{1,2,e_1}$ and $t(e_3, e_2|e_1)$ are 0. Thus, the decision variable c is enforced to be 1 by the inequalities (5) and (6), respectively.

The goal is to minimize the objective $\sum_{c \in \mathbf{C}} c A_c$. \mathbf{C} denotes the decision variables of all avoidable crossings. A_c denotes the overlapping area of a crossing $c \in \mathbf{C}$. A_c is estimated by the product of the weights of the edges that generate this crossing, e.g., 13×21 in Fig. 6A1. Minimizing the objective aims to make some decision variables c to be 0 (i.e., to avoid crossings) by determining $x_{i,j,e}$. Once $x_{i,j,e}$ is determined, the stacking orders of g_i and g_j on e are obtained. Please refer to the studies [7], [76] for details.

Phase II: Arrange descendant geo-networks. After root geo-networks has been well arranged, we recursively traverse and arrange the descendants of each of them. This phase is illustrated in Fig. 5B2 and Fig. 5C.

Once a geo-network is arranged, we arrange its two child geo-networks and only consider the crossings between them. Whether they intersect with others has been determined by the arrangement of their parent. Thus, all descendants can be arranged via such a pair-wise optimization during a recursive traversal. Take Fig. 5 as an example. g_2 and g_3 are sibling geo-networks and have the same parent g_1 . Once g_1 is arranged, we only need to arrange g_2 and g_3 . After g_2 or g_3 is done, we then arrange its children. The arrangement with minimal crossings is obtained when the traversal completes.

To arrange two sibling geo-networks, e.g., g_2 and g_3 , we enumerate all $2^{|E_{g_2} \cap E_{g_3}|}$ candidate arrangement solutions, where the exponent is the number of edges the two geo-networks have. For each candidate, we accumulate the overlapping area over the nodes they involve as the cost. The optimal arrangement has the minimum cost. It is fast enough when $|E_{g_2} \cap E_{g_3}|$ is small. Otherwise, the ILP can be applied again.

4.2.3 k Coloring

To satisfy **G3**, we develop a k coloring algorithm for coloring geo-networks with k colors based on traditional graph col-

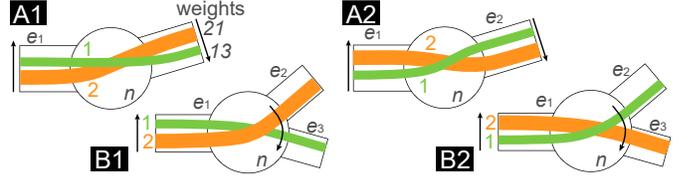


Fig. 6: (A and B) Two types of avoidable crossings.

oring methods. We first formulate the coloring constraints tailored to our scenario and then apply a two-phase progressive coloring strategy.

Coloring constraints. For each $l_t = 0, 1, \dots$, we generate a *positional graph* $G_{l_t}^p$ that describes coloring constraints. In such graphs, each vertex denotes a retrieved geo-network and each edge denotes the positional relationships between geo-networks. We define three relationships for geo-networks to adapt our scenario. The first one is *adjacency* if they have at least one edge placed adjacently in any background edge. The second one is *intersecting* if they produce at least one crossing. The third one is *co-edge* if they shared the same background edge. G^p can be generated based on the arrangement result. So far, we have a basic constraint (1): Geo-networks with any positional relationships cannot have the same color.

Given each $\{G_{l_t}^p | l_t \in 0, 1, \dots\}$, backtracking methods [8] can label every geo-network starting from 0 under the constraint (1). Each label corresponds to a color. However, positional graphs for numerous geo-networks can be dense and the maximal label exceeds the number of distinguishable colors k . Thus, we have to exploit k colors by reusing them. For the v -th color, $v < k$, we sample around it in the LAB color space to obtain slightly different colors. These colors correspond to labels $v+k \times n$, $n = 1, 2, \dots$. That is, the labels with the same remainder to k correspond to similar colors. In this way, we have to consider two constraints: (2) Adjacent geo-networks and (3) intersecting geo-networks cannot be assigned similar colors. Finally, since users may split a geo-network for local drill-down, we consider the fourth constraint: (4) A color that cannot be assigned to a geo-network cannot be assigned to its descendants. The priorities of constraints (1) to (4) are getting lower.

Coloring. We develop a two-phase progressive method that considers the four constraints and ensures the color coherence across levels.

Phase I: Color root geo-networks. We first color the root geo-networks retrieved by $l_t = 0$. Initially, to make full use of k colors, the top- k wide geo-networks are assigned labels $0, 1, \dots, k-1$. The others are assigned label 0. Afterward, we adopt the well-established backtracking method [8]. Briefly, after geo-networks g_1, \dots, g_{cur} are labelled, we label g_{cur+1} from 0 to $Global_Cur_Max + k + 1$ and test the safety. $Global_Cur_Max$ is the global current maximal label. Labelling a geo-network with v is safe if the four constraints are satisfied. If a safe label for g_{cur+1} cannot be obtained, then backtrack to increase the label of g_{cur} .

Phase II: Color descendant geo-networks. We traverse the descendant geo-networks level by level in a top-to-bottom manner. For each level, the wider of the sibling geo-networks inherits the label of its parent during the traversal

TABLE 1: Dataset description. #*geo-nodes*: number of geo-nodes. #*g*: number of geo-networks.

Name	# <i>geo-nodes</i>	# <i>g</i>	Desc.
AirVis	39 districts	317	Air pollutant propagation patterns
AirCas	5 districts	15	Air pollution events' cascades
JamCas	5 roads	12	Traffic jam events' cascades
HuMo₁	23 locations	52	Origin-Destination flows of citizens
HuMo₂	23 locations	56	Origin-Destination flows of citizens

to ensure the coloring consistency. Taking Fig. 5 as an example, g_5 inherits g_4 's label. Afterward, more than half of the geo-networks at the level have labels. Finally, we adopt a greedy method [42] to label the remainders at the level with the minimum safe label.

If positional graphs are too dense, solutions in both phases cannot be obtained under all constraints. If so, we gradually remove the lowest priority constraints to obtain solutions. In the rest of this paper, we use 8 colors (See Fig. 8) from the *Tableau-10* color scheme for categorical fields.

4.2.4 User Interactions

In addition to the user interactions for steering the level-of-detail rendering, we also implement an interaction for pinning and highlighting the geo-networks of users' interests. We add a switch to GeoNetverse. When the switch is turned on, all rendered geo-networks fade to an opacity of 0.5. Users can click geo-networks and the opacity of the clicked geo-networks reverts to 1. After the switch is turned off, those unclicked geo-networks fade to an opacity of 0.2, highlighting those clicked ones.

5 EVALUATION

We evaluate our approaches with quantitative experiments, case studies, and a within-subject user study.

5.1 Dataset Description

AirVis dataset is from Deng et al.'s work [13]. It comprises many propagation patterns of air pollutants in Beijing-Tianjin-Hebei (BTH) region. Each pattern is a directed and weighted geo-network, over which air pollutants frequently propagated. The weight is the occurring frequency. We ignore directions in the hierarchical clustering because these geo-networks inherently have many bi-directional edges.

AirCas dataset is from Deng et al.'s work [14]. Each geo-network reflects how air pollution events cascaded over districts around Hangzhou. The weight also indicates the occurrence frequency.

JamCas dataset is also from Deng et al.'s work [14] and is similar to AirCas dataset. Each geo-network here describes the cascades of traffic jam events in Hangzhou.

HuMo₁ and **HuMo₂** (**human mobility**) datasets are synthetic datasets we generate based on the rules in the urban planning and transportation fields. Studying OD (Origin-Destination) flows of citizens is a popular topic. OD flows are widely characterized as geo-networks [61]. We do not have such datasets available and hence decide to generate datasets. We hope the synthetic geo-networks can describe the human mobilities during morning rush hour [25]: residential areas like "volcanoes" send out citizens, while CBDs

(central business districts) like "black holes" attract citizens for work. The data generation has three steps:

- 1) We manually select locations on the map of a city. They are overall evenly distributed in the city. We set four centered locations as CBDs N_c and those peripheral locations as residential areas N_r . Each $n_c \in N_c$ is origin (i.e., O) and each $n_r \in N_r$ is destination (i.e., D).
- 2) We follow Huff [28] to compute the OD volume between each pair (n_r, n_c) . Particularly, we estimate the willingness score of citizens at n_r to work at n_c as the geographic distance between n_r and n_c divided by n_c 's attraction. Afterward, for each n_r , we normalize the willingness scores of working at n_c and then compute the OD volume as the normalized score multiplied by the citizen number of n_r . For diversity, in our setup, two CBDs are twice as attractive as the other two; one residential area has about 600 citizens, three about 300, and others about 100.
- 3) For each pair (n_r, n_c) , the shortest path from n_r to n_c constitutes a geo-network and the volume from n_r to n_c is the weight the geo-network.

We perform the above three steps on two cities respectively and obtain the two synthetic datasets.

The datasets are summarized in Table 1. We use them for five reasons. First, they describe typical urban phenomena in popular urban domains. Second, the real-world datasets enable case studies that can demonstrate GeoNetverse's practicality. Third, compared with the real-world datasets, the synthetic dataset has a moderate scale with 50 geo-networks and 23 geo-nodes. As a result, the five datasets ranging in scale from small to large are suitable for evaluating the scalability. Fourth, it is necessary to evaluate GeoNetverse on the datasets with differently distributed geo-networks. In AirVis dataset, multiple geo-networks have background edges in common. There are 9 edges with more than 20 geo-networks. On average, there are 8.17 geo-networks per edge. In the synthetic datasets, multiple geo-networks end at "black holes," constituting high-degree geo-nodes. Finally, the two synthetic datasets that differ only in the geo-node distribution and geographic context enable a fair within-subject user study (See Sec. 5.4).

5.2 Quantitative Experiment

We conduct experiments to evaluate the runtime and memory consumption of our method and the performance of crossing minimization and coloring algorithms. The experiments were performed on a desktop running Ubuntu 20.04 with Intel Core i7 3.70GHz CPU, and 16 GB RAM. The crossing minimization is implemented with Python 3.8 and ILP problems are solved by the PuLP solver (<https://coin-or.github.io/pulp/>). The hierarchical clustering and k coloring are implemented with TypeScript.

Runtime. We record the runtimes (average of 10 times) of our three algorithms and summarize them in Table 2. The whole pipeline needs less than 2s for the four datasets with a smaller number of geo-networks and takes about 25s for AirVis dataset. The most time-consuming step is to construct an ILP model with a matrix where the number of rows is the number of constraints and the number of columns is the number of variables. Although some algorithms are

TABLE 2: Algorithm runtimes (ms). The runtimes after the plus sign are for constructing ILP models.

Algorithm	HuMo ₂	AirCas	JamCas	AirVis
hierarchical clustering	49	12	8	8300
crossing minimization	60+1300	11+225	50+8	50+17100
k coloring	19	11	10	140

estimated to have high time complexity (See appendices), they can be completed quickly.

Memory consumption. The runtimes of constructing an ILP model raise concerns about whether it causes too much memory consumption. After inspection, we find that the ILP model for AirVis dataset includes 2,056 constraints and 863 variables and consumes 116.64 MB of memory. ILP is a memory-intensive method, and it is not advisable to directly apply it to too many geo-networks. For example, 233 geo-networks covering 39 locations are retrieved for AirVis dataset when $l_t = 4$. Directly formulating the arrangement problem of these geo-networks as ILP constitutes 83k constraints and 7k variables. Generating a matrix of $83k \times 7k$ and feeding it into an ILP solver exceed the memory limit. Such memory consumption increases sharply as l_t increases. For AirVis dataset, we were not able to directly use ILP to arrange the geo-networks when $l_t \geq 4$.

Our method is more computationally scalable than ILP-based methods alone regarding the runtime and memory consumption. GeoNetserve adopts the hierarchical clustering. It only needs to process fewer geo-networks generated by clustering rather than all original ones.

Crossing minimization. The level-of-detail rendering increases the legibility of geo-networks and avoids users being overwhelmed by geo-networks. Our crossing minimization algorithm follows this rendering mechanism to reduce crossings while maintaining edges' position across levels. Thus, some avoidable crossings inevitably remain. Applying ILP [7] directly to each level of geo-networks can produce the least crossings. But in this way, edges' position across levels cannot be maintained. We expect our method far outperforms a random arrangement (i.e., a baseline) and is not much worse than applying ILP directly.

For each target level for each dataset, we obtain the number of crossings under three different arrangement methods and show the results in Fig. 7. The results of our two-phase progressive arrangement are indicated by red circles. The results of directly applying ILP for each target level are indicated by blue crosses. As for the random arrangement, we obtain 10 random results and represent them with boxplots. As mentioned before, we were not able to arrange the geo-networks when $l_t \geq 4$ in AirVis dataset using ILP directly.

Our method is far better than random arrangement. It is slightly worse than applying ILP directly because we consider the constraints of position maintenance enforced by the level-of-detail rendering. As the target level increases, the gap between our method and ILP increases because our method has to consider more constraints for deeper geo-networks. In AirCas and JamCas datasets, the background geo-networks are simple, and every individual is sparse, shown in Figs. 10 and 11. On these datasets, our method is able to achieve the same results as ILP.

Coloring. The performance of k coloring algorithm is

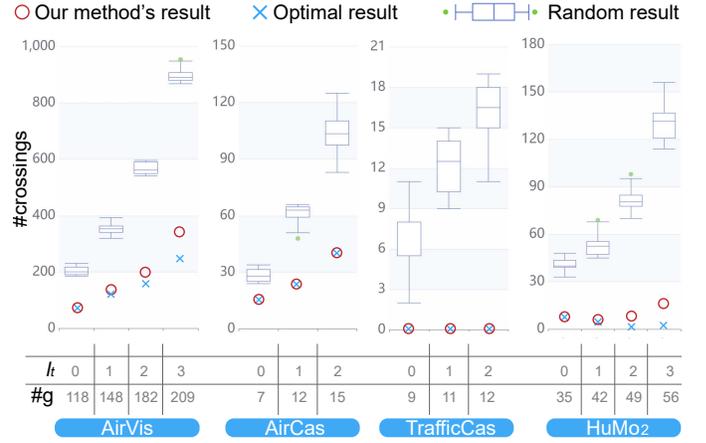


Fig. 7: Quantitative performance on reducing crossings. Our method needs to maintain edges' position across levels and thus is slightly inferior to using ILP directly, but far outperforms random arrangement. #g denotes the number of geo-networks with respect to l_t .

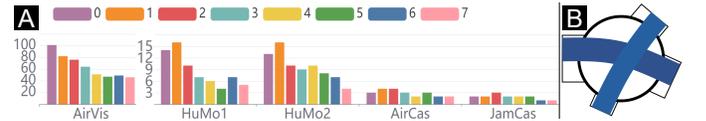


Fig. 8: Left: Color usage of all geo-networks in hierarchies. Right: The violation of Constraint (3).

evaluated from two aspects. First, for each dataset, we count k colors assigned to all geo-networks in the hierarchies. Similar colors are considered the same in the counting. Fig. 8A shows the statistics. Overall, these k colors are fully and evenly utilized and no color is over-reused. Our algorithm eliminates confusion by reducing similar on-screen colors.

Second, we measure the discriminability of geo-networks by reporting how their colors violate the constraints. How constraint (4) is violated depends on the user interaction of drilling down. Therefore, this experiment only considers constraints (1), (2), and (3). In particular, for each dataset, we generate a set of positional graphs $\{G_{l_t}^p | l_t = 0, 1, \dots\}$. For each $G_{l_t}^p$, we enumerate all edges and test whether the colors of two geo-networks connected by an edge violate the constraints. We consider two geo-networks confused if their colors violate any constraint.

Confused geo-networks are only detected in the AirVis dataset. In other datasets, geo-networks would not confuse users. AirVis dataset comprises more geo-networks than others, so the coloring algorithm inevitably violates constraints. Table 3 shows the number of confused geo-networks in AirVis dataset by target levels. Only the lowest priority constraint (3) is violated and the violations are detected when $l_t \geq 3$. There are more than 200 geo-networks and more than 4 geo-networks per edge. In these violations, intersecting networks have similar colors. Nonetheless, we think users can distinguish them because the intersection has a discontinuous shape (e.g., Fig. 8B).

TABLE 3: Number of confused geo-networks in AirVis dataset by target levels and violated constraints. #g: number of geo-networks regarding l_t . #G: number of geo-networks per background edge on average. Cx: Constraint (x).

l_t	0	1	2	3	4	5	6	7	8	9	10	11	...
#g	118	148	182	209	233	250	265	282	295	303	306	309	...
#G	2.5	3.2	3.9	4.7	5.3	5.8	6.3	6.9	7.3	7.6	7.7	7.8	...
C3	0	0	0	4	13	22	32	49	65	73	76	80	...
C2	0												
C1	0												

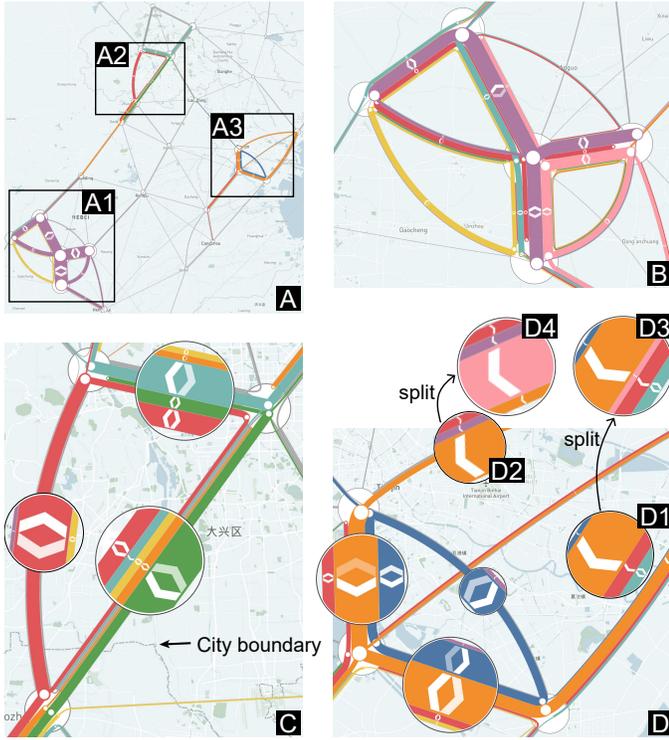


Fig. 9: Case study on AirVis dataset. (A) Air pollutants frequently propagated within three regions. (B) Air pollutants frequently propagated back and forth within the region (A1). (C) Air pollutants propagated back and forth between two cities in the region (A2). (D) The air pollutants of the three districts in the region (A3) were from the northeast.

5.3 Case Studies

Three real-world cases demonstrate how multilevel analyses are performed in a coherent spatial context, even for more than one hundred geo-networks.

5.3.1 Propagation Patterns of Air Pollution in BTH Region

We explore 317 geo-networks in AirVis dataset. Each geo-network denotes a propagation pattern of air pollution. This case shows that starting from the macro level and drilling down can gain insights that were unavailable with a previous bottom-up approach [13].

Macro-level. After the dataset is loaded, overall patterns can be obtained (Fig. 9A). Thick edges can be observed in the three regions, namely, Shijiazhuang region (Fig. 9A1), Beijing (Fig. 9A2), and Tianjin (Fig. 9A3). These regions cover the three largest cities in the dataset. The edges between them are relatively thin. These observations suggest that air

pollutants spread mainly within cities, and less often across cities. Such macro-level findings motivate us to drill down into these three cities, respectively. The macro-level analysis enabled by the summary of all geo-networks 1) avoids users being overwhelmed by too many geo-networks, 2) reveals an overall pattern that air pollutants spread mainly within three major cities respectively, and 3) guides users to the next step of exploration and analysis. Localized analysis can be performed easily by zooming in on the regions of interest because all geo-networks are visualized on the same map. In this way, users retain the context of macro analysis for meso-level analysis.

Meso-level. We first zoom in on Shijiazhuang region (Fig. 9A1). A purple geo-network connects most districts of this region using its thick edges. Moreover, these edges each have two opposite and similarly bright and sized arrows. It seems that air pollutants frequently were transported back and forth within the districts nearby Shijiazhuang. For further confirmation, we drill down the purple geo-network. We interactively split it four times and obtain the result in Fig. 9B. Each finer-grained geo-network is placed under its parent while other geo-networks' positions remain unchanged. Thereby, no sudden visual changes interrupt the analysis flow. These finer-grained geo-networks are embedded in the same background layer that can be used as a reference. We can easily compare them to learn the structural difference between them and the regions they involve in common. Particularly, we notice the same set of districts are still connected by the bi-directional edges of these geo-networks. To conclude, these districts oftentimes share the same fate of air pollution. It is worth mentioning that the edge staking way makes geo-networks less occluded by each other. Besides, since geo-networks are colored differently, users can distinguish each one.

Meso-level. We then zoom in on Beijing (Fig. 9C). The propagation patterns in Beijing are mainly clustered into cyan, green and red geo-networks. Particularly, the green and red geo-networks connect two districts of Beijing and one district in another city via thick and bi-directed edges. In other words, better pollution control needs to consider close cooperation between the two cities. Geo-networks are still presented in aggregation forms but in more detail. Users do not need to stare at many arrows to obtain the edge directions. Moreover, these three geo-networks generate few avoidable crossings. Their edges are easy to track.

Meso-level. Finally, we zoom in on Tianjin (Fig. 9D). There are two main geo-networks, the blue one and orange one. The blue geo-network connect the three districts in Tianjin with bi-directed edges. This indicates that the air pollutants were also transported back and forth like that in Shijiazhuang region. The orange geo-network reveals an interesting phenomenon by the one-way arrows in Figs. 9D1 and D2: all three districts suffered from the air pollutants propagated from the northeast and but pollutants there were rarely transported to the northeast. We split the orange geo-network to see details. It is split into a pink one and an orange one. The orange one inherits its parent's color. The pink one can be easily distinguished from the adjacent orange and purple ones. Surprisingly, the two geo-networks each dominate one of the edges, respectively (Fig. 9D3 and D4), rather than occupying them equally. That is to say, air

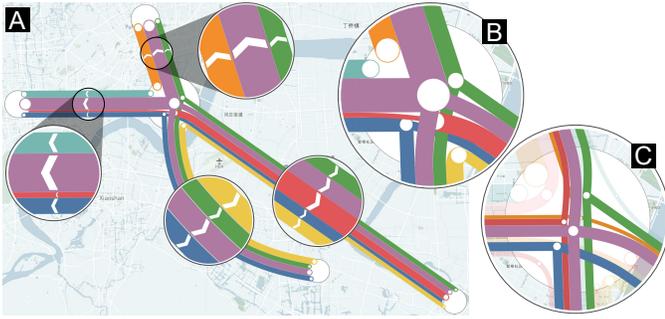


Fig. 10: Case study on AirCas dataset. (A) All geo-networks originate from the central district. (B) An enlarged view of the central district shows that many cascading patterns have 3 or more edges. Air pollutants can simultaneously propagate to the other four districts from the center. (C) Individuals of interest are accessed after drilling down.

pollutants were less often transported from multiple paths at the same time. Further analysis requires the involvement of geography and meteorology experts.

5.3.2 Cascades of Air Pollution Events around Hangzhou

In the second case, we analyze AirCas dataset. Each geo-network summarizes a kind of cascading process of air pollution events. Deng et al. [14] analyzed these geo-networks in a geo-abstracted manner. By contrast, GeoNetverse enables analyzing them within a geographic context.

Macro-level. We load the dataset and obtain the results (Fig. 10A). All geo-networks start from the central node. Fig. 10B is the enlarged view of the central node of Fig. 10A. Some geo-networks (e.g., the blue, green, and purple ones) contain multiple edges. The same conclusion in [14] can be drawn: “the central location’s pollution usually influenced many other locations but not always all of the locations were.”

Micro-level. If a geo-network has three or four edges, it represents the cascades leading to the districts around Hangzhou. Such a geo-network deserves further analysis because the event cascades behind it expanded the influences of air pollution in the central district to around, resulting in severe air conditions throughout the Hangzhou region. To find such geo-networks in detail, we drill down globally by increasing l_t . At the most detailed level, five geo-networks are identified and highlighted in Fig. 10C. Afterward, some domain-specific views can be used to analyze them. For example, a timeline can help learn when these cascades occurred.

5.3.3 Cascades of Traffic Jam Events in Hangzhou

In this case, we analyze JamCas dataset. Each geo-network summarizes a kind of cascading process of traffic jam events.

Macro-level. We can easily identify the same patterns in the prior work [14]. First, among these geo-networks (Fig. 11A), the purple, yellow, orange, and cyan ones have higher weights and have only one edge. Thus, the cascading processes along the four edges rarely co-occurred. It is because they are caused by different traffic demands [14]. Second, the purple geo-network has far more weight than the others and is worthy of being further investigated.

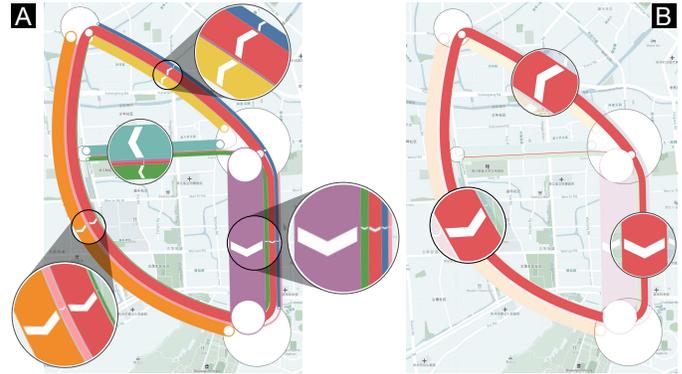


Fig. 11: Case study on JamCas dataset. (A) The purple, yellow, orange, and cyan geo-networks have only one edge and a large weight. (B) An unexpected cascading pattern is identified based on the spatial context.

Micro-level. Besides, we also identify an unexpected geo-network that is colored red and highlighted in Fig. 11B. This geo-network means that traffic congestion events first cascade to the top-left location and then the location below. Such a tortuous cascading path is strange in traffic scenarios. Further cascading pattern validation or model tuning is required. The prior method [14] fails to reveal this issue because it adopted geo-abstracted representations. The identification of this unexpected geo-network shows that the geo-mapped representation is effective and intuitive in reasoning about geo-networks.

5.4 User Study

In this section, we compare GeoNetverse with the alternative design described in Sec. 3.3 with a task-based within-subject user study. We aim to further understand GeoNetverse’s weaknesses and strengths by collecting user feedback of our method. Besides, the alternative combines the first and second types while our method extends the third type. We expect that our method outperforms the straightforward combination of existing methods. The comparison can also reveal the cons and pros of these three types.

Baseline. We implement the alternative and consider it a baseline method in the following user study. It comprises an individual view and an aggregation view. The individual view consists of many mini-maps, on each of which an individual geo-network is depicted, similar to Fig. 1B. The aggregation view depicts the aggregation geo-network on a map, similar to Fig. 1C. In both views, edge widths encode weights. Three interactions were implemented. 1) Users can filter individuals by clicking nodes or edges in the aggregation view. 2) When hovering over an individual, the edges and nodes it contains are highlighted in the aggregation view. 3) Users can sort individuals by their weights. The aggregation view provides entries for users to explore individuals (MA). The individual view enables analyzing details of geo-networks (ME and MI).

Participants and materials. We recruited 12 participants (5 females and 7 males) majored in computer science. 4 participants have basic visualization knowledge. Every one is familiar with maps (e.g., Google Map) and uses them frequently in daily lives. We used HuMo₁ and HuMo₂.

No.	Descriptions	Levels			General Network Tasks				General Geography Tasks				
		Macro	Meso	Micro	Topology		Attribute		Browsing Follow Path	Identify	Compare	Rank	Associate
					Accessibility	Adjacency	Node	Link					
Q1	Which edge the most people pass through?	√						√				√	
Q2	Which direction do most people pass through it?		√					√			√		
Q3	What are the origins of these OD flows?			√					√				√
Q4	Which OD flow mainly dominates the direction?		√					√				√	
Q5	Which four locations are CBDs?	√				√	√	√		√			
Q6	Which four are residential areas?	√				√	√	√		√			
Q7	Which CBDs do citizens in a specified residential area tend to go to work? Answer in descending order of		√	√				√	√			√	√
Q8	Where do the residents who work in a specified CBD live?			√	√				√				√
Q9	Among these locations, residents of which ones need to travel across most of the city space (more than or equal to three edges) to work in this CBD?			√	√				√	√			
Q10	What are the shared work paths for these residents?		√						√		√		

Fig. 12: Ten questions raised for user studies.

Exploring such datasets does not require much domain knowledge because human mobilities are easy to understand for general users. Thus, general users can be recruited as participants to obtain reliable conclusions with generality.

Fig. 12 describes the ten questions participants need to answer. The ten questions are divided into three parts (Q1-Q4, Q5-Q7, and Q8-Q10) and the questions in each part are related. These questions are representative in geo-network analyses. First, they are commonly seen in real-world urban analysis tasks. Q1-Q4 are similar to identifying high-traffic areas that are potential bottlenecks. Q5-Q7 are similar to disclosing black holes/volcanos where anomalous events tend to occur [25]. Q8-Q10 are similar to analyzing co-movement patterns [12], [70] that, for example, could help set up shuttle buses for commuters. Second, they involve three analysis levels participants need to go through back and forth. Third, they cover three low-level general network tasks raised in [34]: topology-based, attribute-based, and browsing. Finally, they are associated with general geography tasks summarized in [50] except for “delineate.”

Procedure. The study is in presence and not virtual. For each participant, we first introduce the background knowledge of geo-networks and human mobilities. We then conduct a tutorial to introduce the alternative system, including visual encodings and interactions. The tutorial is conducted offline. We ensure that they fully understand the system, the human mobility data, and the questions. Afterward, they use the system to analyze one of the datasets and answer the questions in the same order. Afterward, we introduce GeoNetverse in the same way. Then participants use GeoNetverse to analyze another dataset and answer the questions. Only the location distribution and geographic context are different. Participants know nothing about either city before. We assume that analyzing them has the same difficulty. To further mitigate the effects of the datasets, six participants analyzed the datasets in the reverse order of the other six. Finally, they fill in a questionnaire (Table 4) with a 7-point Likert scale to comment on GeoNetverse.

During the experiments, the screen and audio are recorded. The response times for each question are also recorded. If a participant spends more than 2 minutes on a question, we remind him to answer “give up.” If a question depends on the answer to the previous question, we provide the correct answer so that the analysis can proceed. We also instructed participants to think aloud so that we could

obtain how they use the systems to perform analyses. The duration of each participant’s experiment is expected to be 45 minutes to 1 hour.

Results. Correct answers to some questions are in the form of sets. Some participants gave partially correct answers to these questions because they ignored the geo-networks with small weights. We marked these answers as almost correct. Besides, we consider the “give up” response incorrect. When analyzing the response times, we removed those records of incorrect answers. Fig. 13 summaries the accuracies, response times, and questionnaire results.

There could be potential learning effects for this study as the alternative was presented first and the participants were performing the same types of tasks. For investigating learning effects, we use Student’s *t*-test to examine the response times of using GeoNetverse and the alternative. The four questions in the first part (Q1-Q4) are basic and directly supported by both systems, which can be revealed by the short response times. No significant difference is observed for Q1-Q4 ($p \gg 0.05$), which indicates the participants performed comparably with two systems on Q1-Q4. We speculate the learning effects on Q1-Q4 are limited. In particular, participants answer the questions only once using the alternative. They were not sufficiently trained during this process to exhibit the learning effect on the questions. Besides, both datasets and systems are different, resulting in different manners of seeking answers. However, we acknowledge that this user study is not rigorous enough to rule out the learning effect entirely for the rest of the questions. In the future, it is desirable to recruit more participants for additional experiments. The system order for the participants of odd/even numbers should be alternated.

Accuracy. There is no clear difference as for the accuracies. As for Q8, participants tended to give almost correct answers. They needed to browse more than ten geo-networks and easily ignore or misunderstand those thin

TABLE 4: Questionnaire.

No.	Question
P1	Is GeoNetverse intuitive and easy to use?
P2	Does the LOD rendering help analyze multiple geo-networks?
P3	Do existing crossings not affect your analysis?
P4	Are the geo-networks easily distinguishable from each other?
P5	Does GeoNetverse require less context switching?

distributed on the map. We use $\#\mathcal{G}$, the average number of geo-networks per background edge, to characterize the distribution. The case study on AirVis dataset shows that GeoNetverse can handle hierarchies that comprise 14 levels with $\#\mathcal{G} = 8.17$. Third, the coloring result depends on the number of geo-networks and again how they are distributed on the map. If too many geo-networks have some background edges in common, the positional graph will be highly dense and then coloring the geo-networks differently with a few colors will be challenging. As per the quantitative experiment, k coloring ($k = 8$) can effectively handle the geo-networks retrieved from AirVis dataset by $l_t = 2$. There are 182 geo-networks and on average 3.9 geo-networks per background edge ($\#\mathcal{G} = 3.9$).

Computation. Our algorithms take over 2 seconds (26 seconds) only on the largest dataset with 317 geo-networks. The hierarchical clustering generates only a small number of root geo-networks that need to be arranged by ILP. The rest of geo-networks can be quickly arranged through hierarchical traversal of hierarchies. Thus, our algorithms perform fast on all datasets. Besides, the hierarchical clustering brings another benefit. Because only root geo-networks rather than all are input into an ILP solver, the algorithms demand less memory consumption than the prior method [7]. More importantly, the algorithms are executed before exploratory analysis, and therefore they do not result in delays in interactive analysis.

GeoNetverse has two **limitations**. First, it requires locations sparsely distributed. In areas where locations are densely distributed, the nodes tend to overlap each other. Furthermore, background edges may seriously overlap each other at these close nodes. Although nodes can be enlarged to make edges less overlap, too large nodes will obscure the geographic information and affect the aesthetics. Spatial simplification [24], [61] can address this limitation by aggregating close locations into one. Second, the hierarchical clustering can perform better if it considers weights or other attributes for similarity measurement. We will investigate this issue in the future because no well-established methods can consider both structural and associated attributes to measure the similarity of networks.

7 CONCLUSION

This study proposes GeoNetverse, a novel visualization for aggregate geo-networks. By presenting aggregate geo-networks on one map, GeoNetverse enables multilevel exploration and analysis in the same and coherent spatial context. We extend metro-map designs to derive an initial design and enhance it with a level-of-detail rendering, a two-phase crossing minimization, and a tailored coloring algorithm to make it scalable to multiple geo-networks. We perform quantitative experiments, case studies, and a within-subject user study to evaluate GeoNetverse.

ACKNOWLEDGMENT

The work was supported by NSFC (62072400). It was also partially supported by (61972356, 62036010) and the Collaborative Innovation Center of Artificial Intelligence by MOE and Zhejiang Provincial Government (ZJU).

REFERENCES

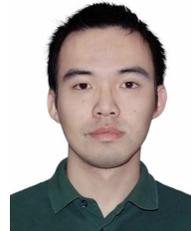
- [1] G. L. Andrienko, N. V. Andrienko, G. Fuchs, and J. Wood. Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2120–2136, 2017.
- [2] N. V. Andrienko and G. L. Andrienko. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):205–219, 2011.
- [3] N. V. Andrienko, G. L. Andrienko, and S. Rinzivillo. Leveraging spatial abstraction in traffic analysis and forecasting with visual analytics. *Information Systems*, 57:172–194, 2016.
- [4] D. Arendt and M. Pirrung. The “y” of it matters, even for storyline visualization. In *Proc. of IEEE VAST*, pages 81–91, 2017.
- [5] B. Bach, E. Pietriga, I. Liccardi, and G. Legostaev. OntoTriX: A hybrid visualization for populated ontologies. In *Proc. of International Conference on World Wide Web (Companion Volume)*, pages 177–180. ACM, 2011.
- [6] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proc. of the working conference on Advanced visual interfaces*, pages 110–119. ACM Press, 2000.
- [7] H. Bast, P. Brosi, and S. Storandt. Efficient generation of geographically accurate transit maps. *ACM Transactions on Spatial Algorithms and Systems*, 5(4):25:1–25:36, 2019.
- [8] E. A. Bender and H. S. Wilf. A theoretical analysis of backtracking in the graph coloring problem. *Journal of Algorithms*, 6(2):275–282, 1985.
- [9] R. Bourqui, D. Ienco, A. Sallaberry, and P. Poncelet. Multilayer graph edge bundling. In *Proc. of IEEE PacificVis*, pages 184–188, 2016.
- [10] K. Buchin, B. Speckmann, and K. Verbeek. Flow map layout via spiral trees. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2536–2544, 2011.
- [11] E. Cakmak, H. Schäfer, J. Buchmüller, J. Fuchs, T. Schreck, A. Jordan, and D. A. Keim. MotionGlyphs: Visual abstraction of spatio-temporal networks in collective animal behavior. *Computer Graphics Forum*, 39(3):63–75, 2020.
- [12] W. Chen, J. Xia, X. Wang, Y. Wang, J. Chen, and L. Chang. RelationLines: Visual reasoning of egocentric relations from heterogeneous urban data. *ACM Trans. Intell. Syst.*, 10(1):2:1–2:21, 2019.
- [13] Z. Deng, D. Weng, J. Chen, R. Liu, Z. Wang, J. Bao, Y. Zheng, and Y. Wu. AirVis: Visual analytics of air pollution propagation. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):800–810, 2020.
- [14] Z. Deng, D. Weng, Y. Liang, J. Bao, Y. Zheng, T. Schreck, M. Xu, and Y. Wu. Visual cascade analytics of large-scale spatiotemporal data. *IEEE Transactions on Visualization and Computer Graphics*, 28(6):2486–2499, 2022.
- [15] Z. Deng, D. Weng, S. Liu, Y. Tian, M. Xu, and Y. Wu. A survey of urban visual analytics: advances and future directions. *Computational Visual Media*, 9(1):3–39, 2023.
- [16] Z. Deng, D. Weng, and Y. Wu. You are experienced: Interactive tour planning with crowdsourcing tour data from web. *Journal of Visualization*, 2022.
- [17] Z. Deng, D. Weng, X. Xie, J. Bao, Y. Zheng, M. Xu, W. Chen, and Y. Wu. Compass: Towards better causal analysis of urban time series. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1051–1061, 2022.
- [18] Y. Dong, C. J. Liang, X. Yuan, and Q. V. Nguyen. CO-Burst: Comparative and progressive exploration for multiple spatiotemporal visualization units. *Journal of Visualization*, 2022.
- [19] S. Dübél, M. Rohlig, H. Schumann, and M. Trapp. 2D and 3D presentation of spatial data: A systematic review. In *Proc. of IEEE VIS International Workshop on 3DVis*, pages 11–18, 2014.
- [20] C. Ducruet. Multilayer dynamics of complex spatial networks: The case of global maritime flows (1977–2008). *Journal of Transport Geography*, 60:47–58, 2017.
- [21] Z. Feng, H. Li, W. Zeng, S. Yang, and H. Qu. Topology density map for urban data visualization and analysis. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):828–838, 2021.
- [22] M. Fink and S. Pupyrev. Metro-line crossing minimization: Hardness, approximations, and tractable cases. In *Proc. of International Symp. on Graph Drawing*, volume 8242, pages 328–339. Springer, 2013.

- [23] E. R. Gansner, Y. Hu, S. C. North, and C. E. Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *Proc. of IEEE PacificVis Symp.*, pages 187–194, 2011.
- [24] D. Guo, X. Zhu, H. Jin, P. Gao, and C. Andris. Discovering spatial patterns in origin-destination mobility data. *Transactions in GIS*, 16(3):411–429, 2012.
- [25] L. Hong, Y. Zheng, D. Yung, J. Shang, and L. Zou. Detecting urban black holes based on human mobility data. In *Proc. of ACM SIGSPATIAL*, pages 35:1–35:10, 2015.
- [26] Y. Hou, C. Wang, J. Wang, X. Xue, X. L. Zhang, J. Zhu, D. Wang, and S. Chen. Visual evaluation for autonomous driving. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1030–1039, 2022.
- [27] X. Huang, Y. Zhao, C. Ma, J. Yang, X. Ye, and C. Zhang. TrajGraph: A graph-based visual analytics approach to studying urban network centralities using taxi trajectory data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):160–169, 2016.
- [28] D. L. Huff. A probabilistic analysis of shopping center trade areas. *Land economics*, 39(1):81–90, 1963.
- [29] C. Hurter, O. Ersoy, and A. C. Telea. Graph bundling by kernel density estimation. *Computer Graphics Forum*, 31(3pt1):865–874, 2012.
- [30] C. Hurter, N. H. Riche, S. M. Drucker, M. Cordeil, R. Alligier, and R. Vuillemot. FiberClay: Sculpting three dimensional trajectories to reveal structural insights. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):704–714, 2019.
- [31] S. Jamonnak, Y. Zhao, X. Huang, and M. Amiruzzaman. Geo-context aware study of vision-based autonomous driving models and spatial video data. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1019–1029, 2022.
- [32] T. R. Jensen and B. Toft. *Graph coloring problems*. John Wiley & Sons, 2011.
- [33] S. Ko, S. Afzal, S. Walton, Y. Yang, J. Chae, A. Malik, Y. Jang, M. Chen, and D. Ebert. Analyzing high-dimensional multivariate network links with integrated anomaly detection, highlighting and exploration. In *Proc. of IEEE VAST*, pages 83–92, 2014.
- [34] B. Lee, C. Plaisant, C. S. Parr, J. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proc. of AVI Workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 1–5. ACM Press, 2006.
- [35] X. Li, Y. Cheng, G. Cong, and L. Chen. Discovering pollution sources and propagation patterns in urban area. In *Proc. of ACM SIGKDD*, pages 1863–1872, 2017.
- [36] Y. Liang, Z. Jiang, and Y. Zheng. Inferring traffic cascading patterns. In *Proc. of ACM SIGSPATIAL*, pages 2:1–2:10, 2017.
- [37] D. Liu, D. Weng, Y. Li, J. Bao, Y. Zheng, H. Qu, and Y. Wu. SmartAdP: Visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):1–10, 2017.
- [38] H. Liu, X. Chen, Y. Wang, B. Zhang, Y. Chen, Y. Zhao, and F. Zhou. Visualization and visual analysis of vessel trajectory data: A survey. *Visual Informatics*, 5(4):1–10, 2021.
- [39] S. Liu, D. Weng, Y. Tian, Z. Deng, H. Xu, X. Zhu, H. Yin, X. Zhan, and Y. Wu. ECoalVis: Visual analysis of control strategies in coal-fired power plants. *IEEE Transactions on Visualization and Computer Graphics*, 2022.
- [40] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. StoryFlow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2436–2445, 2013.
- [41] M. Lu, C. Lai, T. Ye, J. Liang, and X. Yuan. Visual analysis of multiple route choices based on general GPS trajectories. *IEEE Transactions on Big Data*, 3(2):234–247, 2017.
- [42] E. Malaguti and P. Toth. A survey on vertex coloring problems. *International Transactions in Operational Research*, 17(1):1–34, 2010.
- [43] F. McGee, M. Ghoniem, G. Melançon, B. Otjacques, and B. Pinaud. The state of the art in multilayer network visualization. *Computer Graphics Forum*, 38(6):125–149, 2019.
- [44] T. Munzner. *Visualization Analysis and Design*. A.K. Peters visualization series. A K Peters, 2014.
- [45] P. Mutzel. An alternative method to crossing minimization on hierarchical graphs. *SIAM Journal on Optimization*, 11(4):1065–1080, 2001.
- [46] C. Nobre, M. Meyer, M. Streit, and A. Lex. The state of the art in visualizing multivariate networks. *Computer Graphics Forum*, 38(3):807–832, 2019.
- [47] M. Nöllenburg. A survey on automated metro map layout methods. In *Schematic Mapping Workshop*, 2014.
- [48] I. Pérez-Messina, E. Graells-Garrido, M. Lobo, and C. Hurter. Modalflow: Cross-origin flow data visualization for urban mobility. *Algorithms*, 13(11):298, 2020.
- [49] V. Peysakhovich, C. Hurter, and A. Telea. Attribute-driven edge bundling for general graphs with applications in trail analysis. In *Proc. of IEEE PacificVis*, pages 39–46, 2015.
- [50] R. E. Roth. An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2356–2365, 2013.
- [51] S. Schöttler, Y. Yang, H. Pfister, and B. Bach. Visualizing and interacting with geospatial networks: A survey and design space. *Computer Graphics Forum*, 40(6):5–33, 2021.
- [52] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of IEEE Symp. on Visual Languages*, pages 336–343, 1996.
- [53] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981.
- [54] T. Tang, R. Li, X. Wu, S. Liu, J. Knittel, S. Koch, L. Yu, P. Ren, T. Ertl, and Y. Wu. PlotThread: Creating expressive storyline visualizations using reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):294–303, 2021.
- [55] T. Tang, S. Rubab, J. Lai, W. Cui, L. Yu, and Y. Wu. iStoryline: Effective convergence to hand-drawn storylines. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):769–778, 2019.
- [56] T. Tang, Y. Wu, Y. Wu, L. Yu, and Y. Li. VideoModerator: A risk-aware framework for multimodal video moderation in e-commerce. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):846–856, 2022.
- [57] Y. Tao and Y. Tang. Progressive visual analysis of traffic data based on hierarchical topic refinement and detail analysis. *Journal of Visualization*, 2022.
- [58] C. Tominski, H. Schumann, G. L. Andrienko, and N. V. Andrienko. Stacking-based visualization of trajectory attribute data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2565–2574, 2012.
- [59] S. van den Elzen and J. J. van Wijk. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2310–2319, 2014.
- [60] C. Vehlow, F. Beck, P. Auwärter, and D. Weiskopf. Visualizing the evolution of communities in dynamic graphs. *Computer Graphics Forum*, 34(1):277–288, 2015.
- [61] T. von Landesberger, F. Brodtkorb, P. Roskosch, N. V. Andrienko, G. L. Andrienko, and A. Kerren. MobilityGraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):11–20, 2016.
- [62] H. Wang, Y. Ni, L. Sun, Y. Chen, T. Xu, X. Chen, W. Su, and Z. Zhou. Hierarchical visualization of geographical areal data with spatial attribute association. *Visual Informatics*, 5(3):82–91, 2021.
- [63] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. van de Wetering. Visual traffic jam analysis based on trajectory data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2159–2168, 2013.
- [64] D. Weng, R. Chen, Z. Deng, F. Wu, J. Chen, and Y. Wu. SRVis: Towards better spatial integration in ranking visualization. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):459–469, 2019.
- [65] D. Weng, R. Chen, J. Zhang, J. Bao, Y. Zheng, and Y. Wu. Pareto-optimal transit route planning with multi-objective monte-carlo tree search. *IEEE Trans. Intell. Transp. Syst.*, 22(2):1185–1195, 2021.
- [66] D. Weng, C. Zheng, Z. Deng, M. Ma, J. Bao, Y. Zheng, M. Xu, and Y. Wu. Towards better bus networks: A visual analytics approach. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):817–827, 2021.
- [67] D. Weng, H. Zhu, J. Bao, Y. Zheng, and Y. Wu. HomeFinder revisited: Finding ideal homes with reachability-centric multi-criteria decision making. In *Proc. of ACM CHI*, page 247, 2018.
- [68] J. Wood, J. Dykes, and A. Slingsby. Visualisation of origins, destinations and flows with od maps. *The Cartographic Journal*, 47(2):117–129, 2010.
- [69] H. Wu, B. Niedermann, S. Takahashi, M. J. Roberts, and M. Nöllenburg. A survey on transit map layout - from design, machine, and human perspectives. *Computer Graphics Forum*, 39(3):619–646, 2020.

- [70] W. Wu, J. Xu, H. Zeng, Y. Zheng, H. Qu, B. Ni, M. Yuan, and L. M. Ni. TelCoVis: Visual exploration of co-occurrence in urban human mobility based on telco data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):935–944, 2016.
- [71] Y. Wu, D. Weng, Z. Deng, J. Bao, M. Xu, Z. Wang, Y. Zheng, Z. Ding, and W. Chen. Towards better detection and analysis of massive spatiotemporal co-occurrence patterns. *IEEE Trans. Intell. Transp. Syst.*, 22(6):3387–3402, 2021.
- [72] 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*, 2022.
- [73] Y. Yang, T. Dwyer, S. Goodwin, and K. Marriott. Many-to-many geographically-embedded flow visualisation: An evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):411–420, 2017.
- [74] Y. Yang, T. Dwyer, B. Jenny, K. Marriott, M. Cordeil, and H. Chen. Origin-destination flow maps in immersive environments. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):693–703, 2019.
- [75] G. G. Zanabria, J. Silveira, J. Poco, A. Paiva, M. B. Nery, C. T. Silva, S. Adorno, and L. G. Nonato. CrimAnalyzer: Understanding crime patterns in são paulo. *IEEE Transactions on Visualization and Computer Graphics*, 27(4):2313–2328, 2021.
- [76] D. C. Zarate, P. L. Bodic, T. Dwyer, G. Gange, and P. J. Stuckey. Optimal sankey diagrams via integer programming. In *Proc. of IEEE PacificVis Symp.*, pages 135–139, 2018.
- [77] W. Zhao, G. Wang, Z. Wang, L. Liu, X. Wei, and Y. Wu. A uncertainty visual analytics approach for bus travel time. *Visual Informatics*, 2022.
- [78] Y. Zhao, H. Jiang, Q. Chen, Y. Qin, H. Xie, Y. Wu, S. Liu, Z. Zhou, J. Xia, and F. Zhou. Preserving minority structures in graph sampling. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1698–1708, 2021.
- [79] J. X. Zheng, S. Pawar, and D. F. M. Goodman. Further towards unambiguous edge bundling: Investigating power-confluent drawings for network visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2244–2249, 2021.



Dr. Xiao Xie is an Assistant Professor at the Department of Sports Science, Zhejiang University. His research interests include data visualization, visual analytics, and human-computer interaction, with a focus on creating novel visual analysis techniques for supporting sports analysis. He received his Ph.D. degree in Computer Science from Zhejiang University.



Dr. Guodao Sun is an associate professor at College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. He received B.Sc. in computer science and technology (2010), and Ph.D. degree in January, 2016. He is interested in exploring large data using interactive visual analytics with focus on data such as urban data (vehicles, human movement, environment etc.), social data (social media, social network etc.), video data (associated with computer vision).



Dr. Mingliang Xu is a professor and director of the School of Computer and Artificial Intelligence at Zhengzhou University, and the director of the Engineering Research Center of Ministry of Education on Intelligent Swarm Systems, China. He received his Ph.D. degree from the State Key Lab of CAD&CG at Zhejiang University, Hangzhou, China, and the B.S. and M.S. degrees from the Computer Science Department, Zhengzhou University, Zhengzhou, China, respectively. He was awarded as the National

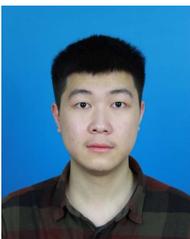
Science Foundation for Excellent Young Scholars. His current research interests include computer graphics, multimedia and artificial intelligence. He has published 100+ papers in ACM/IEEE Transactions and full papers on top-tier conferences, such as CVPR and ACM Multimedia.



Zikun Deng received his B.S. degree in Transportation Engineering from Sun Yat-Sen University in 2018. He is currently pursuing the doctoral degree with the State Key Lab of CAD&CG, Zhejiang University. His research interests mainly include spatiotemporal data mining, visualization, and urban visual analytics. For more information, please visit <https://zkdeng.org>.



Dr. Di Weng is a researcher at Microsoft Research Asia. He received his PhD degree in Computer Science from Zhejiang University in 2021 and BS degree from Taishan Honors College, Shandong University. His research interests mainly include the data mining, visualization, and visual analytics of large-scale spatiotemporal data. For more information, please visit <https://dwe.ng>.



Shifu Chen received his B.S. degree in Software Engineering from University of Electronic Science and Technology of China in 2022. He is currently a graduate student in Software Engineering, Zhejiang University. His research interests mainly include visualization and visual analytics.



Dr. Yingcai Wu is a Professor at the State Key Lab of CAD&CG, Zhejiang University. His main research interests are in information visualization and visual analytics, with focuses on urban computing, sports science, immersive visualization, and social media analysis. He received his Ph.D. degree in Computer Science from the Hong Kong University of Science and Technology. Prior to his current position, Dr. Wu was a postdoctoral researcher in the University of California, Davis from 2010 to 2012, and a researcher in Microsoft Research Asia from 2012 to 2015. For more information, please visit <http://www.ycwu.org>.