J Vis
https://doi.org/10.1007/s12650-021-00779-7

REGULAR PAPER

Xiangyang He · Yubo Tao · Shuoliu Yang · Chuanchang Chen · Hai Lin

# ScalarGCN: scalar-value association analysis of volumes based on graph convolutional network

Received: 8 July 2021 / Revised: 20 July 2021 / Accepted: 28 July 2021


**Abstract** The relationships in multivariable data are intricate, and there are usually implicit associations between scalar values variables. However, existing association analysis methods lack spatial measurement of scalar values, and fail to collaboratively analyze the association between scalar values and variables. Thus association results may be one-sided. In this paper, we construct a scalar-value neighborhood graph to preserve the spatial information for scalar values and propose a graph neural network model composed of multiple graph convolutional layers and a self-attention mechanism for learning the low-dimensional vectors of scalar values and variables simultaneously. Several case studies show the scalability and flexibility of our method on analyzing the association between scalar values and variables.

**Keywords** Multivariate data · Association analysis · Graph neural network

## 1 Introduction

Multivariate data are widely used to extract hidden relationships and explore existing phenomenons or new patterns in many fields, such as meteorological simulation, nuclear fission simulation, blood flow simulation, and natural disaster simulation. In multivariate data, there is often a relationship between different scalar values or variables, and the spatial domain also contains complex interaction relations. How to obtain the implied association through the extracted potential information so that domain experts can identify the relationship between voxels, scalar values, and variables has always been a problem faced by visual analysis of multivariate data Kehrer and Hauser (2012) He et al. (2019).

Association analysis focuses on mining global or local associations in volumes, such as the association between scalar values in univariate data and the entire association between variables in multivariate data. According to the different relationships studied, we classify associations into the following categories: association between scalar values Liu and Shen (2015), association between voxels Sauber et al. (2006), and

X. He · Y. Tao (✉) · S. Yang · C. Chen · H. Lin (✉)
State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
E-mail: taoyubo@cad.zju.edu.cn

H. Lin
E-mail: lin@cad.zju.edu.cn

X. He
E-mail: xiangyanghe@zju.edu.cn

S. Yang
E-mail: lucida@zju.edu.cn

C. Chen
E-mail: chenchuanchang@zju.edu.cn

Published online: 04 September 2021

association between variables Biswas et al. (2013). However, instead of co-analyzing scalar values, voxels (space), and variables, previous association methods only investigate one aspect. Firstly, existing methods cannot integrate scalar values with spatial information. There are always strong associations between scalar values and their spatial positions. For example, features with a low wind speed in the Hurricane Isabel dataset are distributed in both the center and the periphery of the hurricane. Secondly, exiting methods independently calculate the association between scalar values of different variables and the association between variables, and fail to collaboratively analyze the association between them.

Deep learning-based methods have been widely used to learn the hidden information of volume data more effectively. These have been applied in many fields of volume visual analysis, such as keyframe detection Porter et al. (2019), super-resolution generation Han and Wang (2020), and in-situ visualization He et al. (2020). However, these works usually conduct convolution and pooling operations directly on the 3D regular structure, and the number of parameters increases dramatically with the resolution of volumes. In this case, these works usually downsample the volume data to the environment suitable for normal training Han et al. (2020), which may result in the loss of the inherent features of the origin volumes. Although deep learning-based methods have stronger representation ability, they have limitations, such as the large amount of data, long training time, uncontrollable training results, and failing to deal with non-Euclidean data directly. In this paper, we convert the volume to a scalar-value neighborhood graph, in which local features (scalar values) in the volume are expressed as nodes and the neighborhood relations between the scalar values are modeled as edges. The nodes in the scalar-value neighborhood graph preserve the scalar-values characteristics of the volumes, and the edges in the scalar-value neighborhood graph preserve the spatial information of these scalar values. We jointly consider the context numerical distribution, spatial distribution, and gradient distribution as the attributes of nodes to integrate scalar values with spatial information. After this, we construct a graph convolutional network (ScalarGCN) to learn the low-dimensional vector representations of scalar values and variables jointly and then analyze the association between them collaboratively.

Using the graph structure to represent multivariate data can effectively reduce redundancy on the premise of retaining important information and it can ensure that ScalarGCN can learn valid information more effectively. Instead of analyzing the association between scalar values or variables independently, we measure the association between the two within the same framework to ensure that ScalarGCN can learn the embedded representations of the scalar values of different variables in the same vector space. Therefore, after the training of ScalarGCN, we can not only analyze the association of scalar values within each variable, but also jointly analyze the association of scalar values between different variables, which has strong scalability and flexibility. The contributions of this paper are as follows:

- A graph representation method for building the relationships between scalar values for multivariate data;
- A graph neural network model composed of multiple graph convolutional layers and a self-attention mechanism for learning for the first time the low-dimensional vector representations of scalar values and variables simultaneously;
- A novel method for quantifying the assocaition of scalar values and variables.

## 2 Related work

In this section, we focus on volume data and limit our coverage to the most related work on association analysis methods, machine learning on volume, and the application on graph.

### 2.1 Association analysis

There are complex interaction relationships in volumes. Many works have been done to extract potential information to acquire implied association and to identify the relationships between scalar values/variables.

For univariate data, Bruckner and Möller (2010) used the isosurface as the intermediate representation for each scalar value and applied the spatial proximity of isosurfaces to measure the association between two scalar values.

For multivariate data, Sauber et al. (2006) introduced a similarity measure method named the gradient similarity measure (GSIM) to mine the association between voxels. Lu and Shen (2017) extracted the association between scalar values in two different variable domains and used the Influence-Passivity Model

to choose the most representative scalar value. Biswas et al. (2013) used the mutual information to measure the informativeness of one variable about another variable and organized variables based on the mutual information in a graph-based form. Haidacher et al. (2011) established distance fields for the scalar values and measured these distance fields using the mutual information to calculate the similarity between scalar values. Liu and Shen (2015) extracted the association between scalar values in different variable domains and used the Influence-Passivity Model to choose the most representative scalar value. He et al. (2018b, 2018a) used a bicluster algorithm to simultaneously cluster variables and voxels and automatically extracted all biclusters with similar scalar values.

For time-varying data, Sukharev et al. (2009) used the correlation coefficient to measure the linear relationship for the same spatial sampling point between two time steps.

Only a single aspect, the scalar value, voxel or volume is considered in these methods. Various features need to be considered for better mining association in volumes. In this paper, the semantic information of a volume's context and the numerical/spatial/gradient distribution information of scalar-values are extracted, which provides a more complete and multi-dimensional quantization standard of associations between scalar values and variables.

### 2.2 Machine learning for volume visualization

Learning the intrinsic representations in volumes has always been a difficult problem in scientific visualization. The rise of deep learning has improved the ability to learn and express features in volumes. Han et al. (2018) applied an autoencoder to a volume to learn the potential representations of streamlines and stream surfaces for dimensionality reduction and representative selection. Porter et al. (2019) also leveraged an autoencoder to learn the representation of volumes to select representative time steps for time-varying multivariate data. Han et al. (2020) aimed to learn the mapping mechanism of one variable to another. Berger et al. (2019) used generative adversarial networks (GANS) to learn the mapping of transfer functions to volume rendering resulting images. Han and Wang (2020) combined a recursive neural network (RNN) and GAN to learn the implicit representation from low-resolution volume sequences to high-resolution ones.

However, the above methods are learning methods oriented on volumes themself. Furthermore, convolution and pooling operations on three-dimensional volume data will dramatically increase the amount of data and incur huge computational costs. In this paper, we express the volume as scalar graph in a hierarchical way and learn the implicit representations of scalar values and variables. This greatly reduces data redundancy while ensuring the context and spatial structure as much as possible and it provides an analytical basis for the downstream association analysis tasks.

### 2.3 Graph on volume

A general node-link diagram represented by graph has been used in data organization and management in scientific visualization for some time Wang and Tao (2017). For example, Carr and Duke (2014) extended the contour tree to Joint Contour Net to extract topological structures by quantifying the variation of multiple variables for topological analysis and visualization in multivariate domains. Carr et al. (2015) extracted fiber surfaces in bivariate fields based on the marching cubes algorithm to generate well-defined geometric surfaces and analyzed the captured geometrical characteristics. Gu and Wang (2011) abstracted volume into a graph form (TransGraph), where a node denotes a spatiotemporal region and a directed edge between two nodes denotes their transition probability, to organize and explore the relationship of 4D time-varying volume data. Lukasczyk et al. (2017) Lukasczyk et al. (2019) presented a nested tracking graph, which allows users to set multiple tracking graphs in context to each other and effectively follow the evolution of features for different levels simultaneously in time-varying volume data. The above methods represent a volume's structures with graph and help users understand volumes better at an interactive level.

However, users' perceptions are limited, and so useful information in graphs tends to be neglected. Therefore, representation learning of volume's structure needs to be mined to directly build the representation learning on the volume's structures and implement the deep coupling between the graph neural network and volumes, which is also the core idea of this paper.

## 3 Association analysis on multivariate data

The scalar values of volume data can be approximately equivalent to isosurfaces in space, so the numerical association is equivalent to the degree of correlation between isosurfaces. It is thus important to study the association between scalar values to reveal the essential characteristics of volumes. The association between scalar values can be used to analyze the macro (volumes) and micro (scalar values) relationships flexibly. However, as described above, most of the association methods fail to take the spatial distribution as well as other information into account, which may lead to overly one-sided correlations. In this section, compared with the previous methods, we consider the context distribution, spatial distribution, and gradient distribution as a whole for scalar values to provide a more complete numerical association measurement method for volumes.

A graph contains two parts: its structure and attribute. Structure information describes the context topological relationship of volumes and plays a key role in the description of the whole graph. Attribute information describes the inherent properties of objects, and these properties are important for describing the nodes in the graph. In this section, we describe the mapping mechanism of volume to graph structure in detail, that is, the structure building (Sect. 3.1) and feature initialization Sect. 3.2) process of the scalar-value neighborhood graph (hereafter referred to as Scalar-Graph). After this, we design (Sect. 3.3) and train (Sect. 3.4) ScalarGCN model using a graph neural network and self-attention mechanism to achieve the simultaneous learning of attribute and structure information for nodes and the global representations of variables.

### 3.1 Scalar-graph construction

The range must be discretized to arrive at a discrete representation for learning. In our paper, 256 is used to quantify volumes, that is, binning the volume into 256 scalar values (from 0 to 255). The scalar values of multivariate data are considered the nodes in Scalar-Graph. Therefore, for a multivariate data set with $m$ variables, there are $256 * m$ nodes in total. We describe the construction process of Scalar-Graph's edges in detail. The spatial distributions of scalar values are often continuous transitions, so the adjacent scalar values are often adjacent in the isosurfaces' distribution in space. Therefore, for a volume itself, we use the neighborhood relationship between the scalar values and their context to construct Scalar-Graph to represent the local structure association of scalar values in spatial distribution (edges within each variable). We use the co-occurrence relationships in the same voxels of the scalar values with different variables to represent the gobal structure association of scalar values cross variables. For convenience, we define the edges within each variable as local edges and those cross variables as global edges. The detailed construction process for both types of edges is as follows:

– Local connections. The relationships between variables are ignored in local edges, and the variables are thought of as individuals. The building process of local edges is shown in Fig. 1. For each variable, we traverse each voxel to obtain $s_i$ and its surrounding six neighbors $s_j \in context(s_i)$ (upper, lower, left, right, front, and back), and add the connection relation $e_l(s_i, s_j)$ for the center node $s_i$ and the context node $s_j$ (Fig. 1a–c). The self-linking relationships are removed because there is no need to meausre the association with itself. Edge $e_l(s_i, s_j)$ measures the frequency of $s_i$ and $s_j$ in the context with the local connections.
  Usually, there are features with small but important scalar values, and the weight of the edge connected to these scalar values is usually low. In addition, a large number of redundant features, such as the

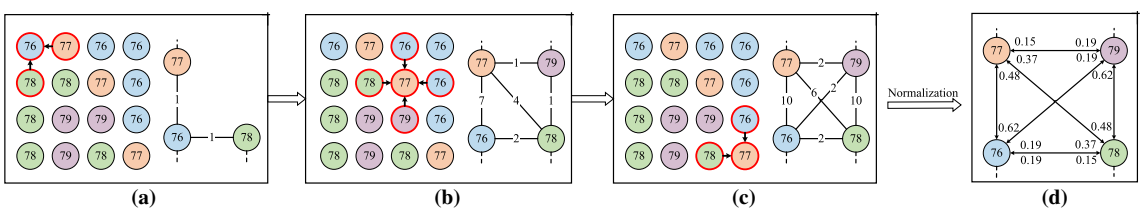



**Fig. 1** An example of the building process of Scalar-Graph's local connections. (**a**), (**b**) and (**c**) show the dynamic building process of Scalar-Graph that traverses voxels and updates the weights of edges. The left views show the volume in node form and the connection relationships between a center node and its context nodes. The right views show the updating process of edges' connection states and weights. (**d**) shows the Scalar-Graph after *log* normalization

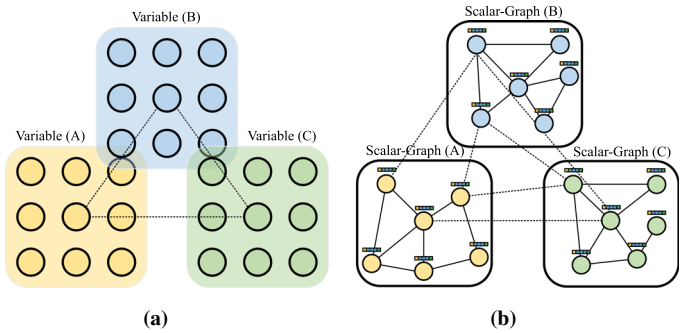



**Fig. 2** An example of the building process of Scalar-Graph's global connections. **a** shows the global connection rules for scalar values between variables. **b** shows the example of a complete Scalar-Graph, where the dotted and solid lines represent the local and global connections, respectively

background, usually have a high weight connected with other scalar values. To reduce this imbalance, we normalize the edges' weights node by node:

$$e_l(s_i, s_j) = \frac{\log e_l(s_i, s_j)}{\sum_{(s_i, s_k) \in e_l} \log e_l(s_i, s_k)}. \tag{1}$$

In this case, Scalar-Graph is converted to a bidirectional graph because $e_l(s_i, s_j) \neq e_l(s_j, s_i)$, as shown in Fig. 1d. An edge with a high weight indicates a high frequency of occurrences between the two nodes in the context. After the building of local connections, there is no connection between the $m$ subgraphs.

– Global connections. For multivariate data, each voxel has a scalar-value set $S = \{s_{v_1}, s_{v_2}, s_{v_3}, ..., s_{v_N}\}$. This set measures the relationships between variables. Many works use PCP Guo et al. (2012) Liu and Shen (2015) to record the numerical relationships between variables. In this paper, we record the relationships between variables by adding a global edge between $s_i$ and $s_j$ when $s_i, s_j \in S$, to build the association between subgraphs, as shown in Fig. 2. We continue to count the co-occurrence frequency between the scalar values of different variables and use *log* normalization to normalize the weight of global edges $e_g$.

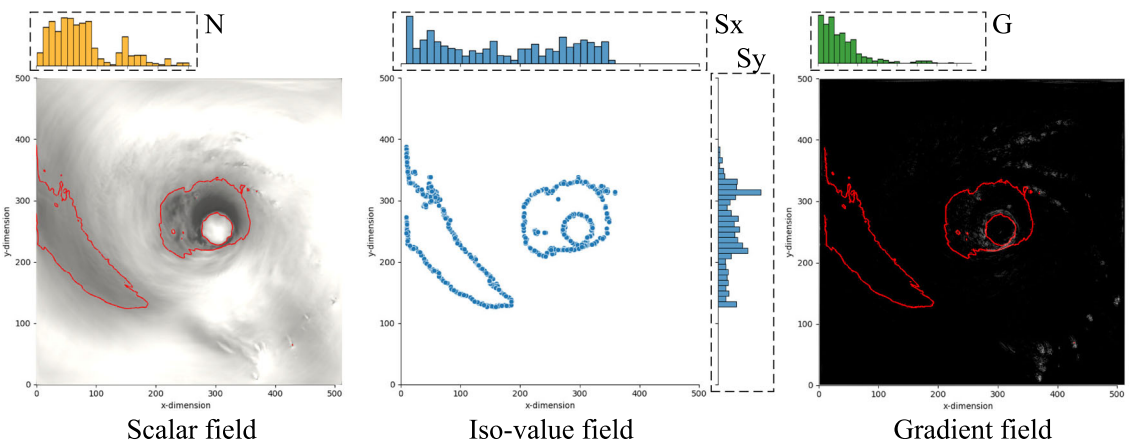



**Fig. 3** An example of the initial features for Scalar-Graph's nodes. The red curves represent the contour of a scalar value of 100 in the scalar field and gradient field. The points in the iso-value field show the corresponding spatial distributions. *N* and *G* show the numerical histogram from the contour's context and the gradient histogram from the voxels where the scalar value = 100, respectively. *Sx* and *Sy* record the spatial histograms of the volume in the *x* and *y* dimensions

### 3.2 Feature selection

The more node information there is, the more sufficient is the characterization of variables. To comprehensively measure the information of scalar values in a variable, we choose three kinds of numerical features, namely context numerical distribution $N$, spatial distribution $S$ in each dimension, and gradient distribution $G$, as shown in Fig. 3. Note that the initial features of nodes are only related to the variable to which they belong. The process of initializing the features for the nodes is described in detail below:

- Context numerical distribution. The representation of a scalar value in its context is of great significance. It can measure the distribution characteristics of scalar values in local space. In Sect. 3.1, we use the sliding window to calculate the weight of edges. In this section,
  we also use a sliding window to count the context numerical distribution for scalar values. We replace the six neighbors described in Sect. 3.1 and expand the receptive field of the scalar values in the context, and then count and normalize the context numerical distribution $N_{s_i}$ of the scalar value $s_i$ in its context window, as shown in the orange histogram in Fig. 3. Note that the window should be neither too large because the graph convolution layer is a process of expanding the receptive field, nor should it be too small to ensure that the initial features provide sufficient contextual information. In this paper, a $3 * 3 * 3$ window is suitable for our experiments.
- Spatial distribution. The spatial distributions of scalar values can be used to measure the spatial similarity between two scalar values. The spatial distributions can be represented by several voxels, such as the points in the iso-value field in Fig. 3. These voxels are disordered, and different scalar values correspond to a different number of voxels. And so, it is difficult to define their spatial distributions directly. In this paper, voxels are represented in the Cartesian coordinate system and the numbers of values that fall into each interval are counted for each dimension as the spatial distribution for the scalar value,
  as shown in the blue histograms $Sx$ and $Sy$ in Fig. 3. In 3D space, the spatial distributions of $Sx_{s_i}$, $Sy_{s_i}$, and $Sz_{s_i}$ are concatenated together as the spatial distribution features of node $s_i$.
- Gradient distribution. Gradient measures spatial stability and the changing trend of the scalar values' contours. We calculate the gradient field according to the scalar field, and then count the gradient magnitude distribution $G_{s_i}$ corresponding to the voxels with the scalar value $s_i$, as shown in the green histogram in Fig. 3.

We concatenated the three distributions together to form the initial feature vector of the node $s_i$:

$$h_{s_i}^{(0)} = concat(N_{s_i}; Sx_{s_i}, Sy_{s_i}, Sz_{s_i}; G_{s_i}). \tag{2}$$

Assuming that the dimension of feature vector is $f$ and that there are a total of $m$ variables and $n$ nodes (scalar values) for each variable, feature matrix $H^{(0)} \in \mathbb{R}^{(mn)\times f}$ describes the initial features of Scalar-Graph's nodes.

### 3.3 ScalarGCN model

ScalarGCN model combines a multi-layer graph convolutional network (GCN) and self-attention mechanism based on unsupervised-learning to obtain the high-order topological structure relations of Scalar-Graph. The multi-layer GCN is used to learn the scalar-value embeddings, and the self-attention mechanism aims to learn the variable embeddings, as shown in Fig. 4.

First, ScalarGCN learns the mapping relationships between the initial features of Scalar-Graph's nodes and their implicit expression using several GCN layers, i.e., $h_{s_i}^{(0)} \mapsto z_{s_i}$. This mapping aggregates $s_i$'s features and those of the nodes with local connections to generate the low-dimensional representations of scalar-values (node embeddings), which learns the local relationships in the variables themselves. We use three GCN layers as the hidden layer of ScalarGCN. The forward propagation formula for the hidden layer $l+1$ of the model is as follows:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}), \tag{3}$$

where $\hat{A}$ and $\hat{D}$ are the normalized degree matrix and edge weight matrix of Scalar-Graph, respectively. We use a weight matrix $W^{(l)}$ to learn the feature mapping relationships of node aggregation (similar to the full connection layer in deep learning), and then acquire the final node representation vectors through a *ReLU* activation function. The GCN layer can carry out the affine transformation on the input graph feature matrix

$H^{(0)}$ and learns the result representation matrix $Z = H^{(3)}$ for nodes. Note that Z is a matrix with a shape of $(mn, d_z)$, where $d_z$ is the feature dimension of $z$.

Second, $z_{s_i}$ is used to learn the global relationships between variables using a self-attention mechanism. We build an encoder to learn the low-dimensional representataions of variables, i.e., variable embeddings. The self-attention mechanism focuses on the global connections while ensuring that the learned variable embeddings remain in the same vector space, which provides the basis for measuring the association between variables.

In order to learn the representation of varibles, it is necessary to reshape $Z$ with $(m, n, d_z)$. Then, a multi-head attention layer with $h$ self-attention heads is employed. Given input node embeddings $z_{s_i}$ and $z_{s_j} \in \mathbb{R}^{d_z}$ where $(s_i, s_j) \in e_l$ or $e_g$, the attention score $head^{(k)}(s_i, s_j)$ is calculated by applying the scaled dot-product for the given input node embeddings as follows:

$$head^{(k)}(s_i, s_j) = \frac{(z_{s_i} W^Q)(z_{s_j} W^K)^T}{\sqrt{d_z}}. \tag{4}$$

After this, we compute the attention features $c^{(k)}$ as follows:

$$c^{(k)} = softmax(head^{(k)})ZW^V, \tag{5}$$

where $softmex(x^{(k)}) = exp(x^{(k)}) / \sum_{i=1}^{h} exp(x^{(i)})$. Note that $W^Q$, $W^K$ and $W^V$ are parameter matrices, which are unique per attention head. We add an other trainable parameter matrix $W^A$ after the mean pooling of $c$ and then calculate the weight coefficient:

$$\alpha^{(k)} = (\frac{1}{h}\sum_{i=1}^{h} c^{(i)})W^A \cdot c^{(k)}. \tag{6}$$

We normalize the weight coefficient using the *softmax* function and then aggregate attention features to obtain the fusion embeddings:

$$\hat{Z} = \frac{1}{h}\sum_{i=1}^{h} softmax(\alpha^{(i)})c^{(i)}, \tag{7}$$

where $\hat{Z} \in \mathbb{R}^{V \times d_z}$. After this, we use a multilayer perceptron (MLP) layer and $L2$ normalization layer to increase the expression of ScalarGCN. The final variable embedding matrix $Y \in \mathbb{R}^{V \times d_y}$ shows $V$ embeddings where $y_i \in Y$ describes the representation of the $i$-th variable with the feature dimension of $d_y$.

### 3.4 Training

The original GCN model Kipf and Welling (2016) adopted semi-supervised training. In this paper, we use unsupervised training to learn node embeddings and variable embeddings.

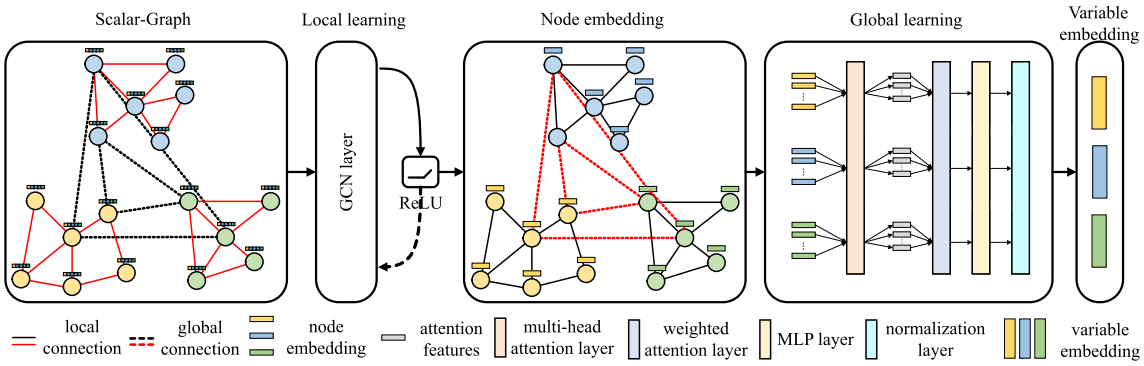



**Fig. 4** The model architecture of ScalarGCN. The local learning process uses several GCN layers to learn the local relationships (red solid lines in the left Scalar-Graph). The global learning process uses a self-attention mechanism and MPL layer, and pays more attention to learn the global relationships (red dotted lines in the right Scalar-Graph)

For local connections, we take the local neighbor nodes as positive samples and non-neighbor nodes as negative samples, and then design the loss function shown as follows:

$$\begin{aligned}\mathfrak{L}_l = &- \sum_{(s_i,s_j)\in e_l} e_l(s_i,s_j)\log(z_{s_i}^T z_{s_j})) \\ &+ \sum_{(s_i,s_k)\notin e_l} \log(1 - z_{s_i}^T z_{s_k}))),\end{aligned} \tag{8}$$

where the dot-product of $z_{s_i}$ and $z_{s_j}$ measures the similarity between the two vectors. Note that $s_i$ and $s_k$ in the second part of Eq. 8 belong to the same variables. The contrastive loss function establishes the co-occurrence relationships between nodes and their local neighbor nodes, which can constantly shorten the distance between neighbor nodes and gradually push away the distance between non-neighbor ones, to iterate and update model parameters.

For global connections, we define a global loss to maximize the mutual information between variables with numerical relationships:

$$\mathfrak{L}_g = -\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{(s_i,s_j)\in e_g} e_g(s_i,s_j) log(y_{s_i}^T y_{s_j}). \tag{9}$$

This loss assumes that variables corresponding to scalar values with global connections and high co-occurrence frequency will be assigned similar embedding vectors. Note that $s_i \in V_a$ and $s_j \in V_b$. This makes sense because the higher the numerical association, the closer the relationships between variables.

We combine $\mathfrak{L}_l$ and $\mathfrak{L}_g$ with a weight coefficient $\lambda = 0.1$, and minimize the following loss function to learn the local relationships between scalar values and global relationships between variables:

$$\mathfrak{L} = \mathfrak{L}_l + \lambda \mathfrak{L}_g \tag{10}$$

Adam optimizer Kingma and Adam (2015) is used to minimize the objective function. ScalarGCN updates the embeddings of variables and scalar values gradually through backpropagation. After the training process, $Z$ and $Y$, respectively, are mapped into the same vector space.

## 4 Application

In this section, we discuss two association analysis applications for multivariate data via ScalarGCN.

### 4.1 Association between scalar values

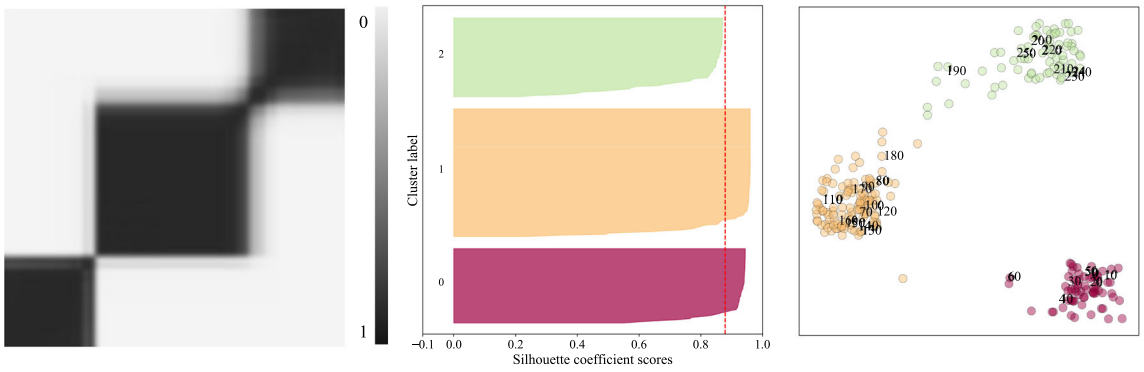
ScalarGCN learns the low-dimensional vector representations of the scalar values of all variables. The embeddings are comparable because we implement the mechanism of embedding numerical values of different variables into the same vector space through global learning and continuous backpropagation. This mechanism provides an opportunity to compare the association between scalar values of different variables, which is one of the major research areas on multivariate spatial data visualziation He et al. (2019). Therefore, in this section, we utilize the low-dimensional vector representations of scalar values to explore the association between scalar values from the micro point of view.

#### *4.1.1 Univariate data*

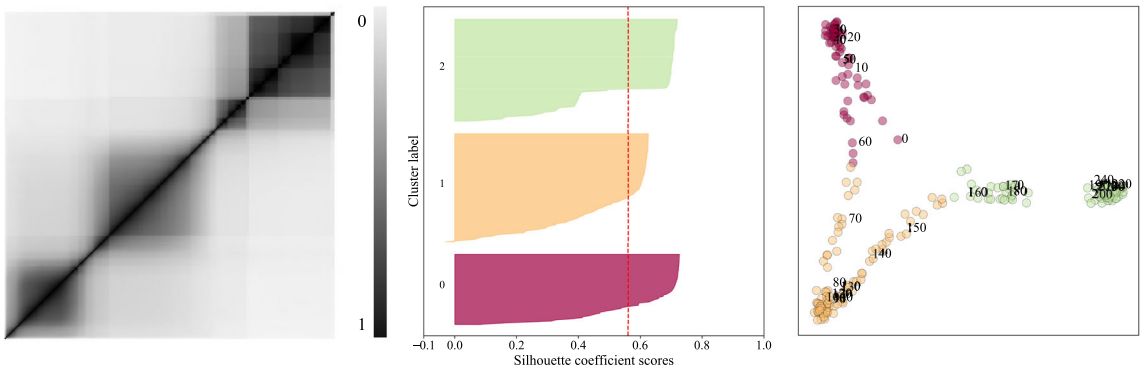
Section 3 shows how to use ScalarGCN to learn the low-dimensional vector representation $Z$ of scalar values and the low-dimensional vector representation $Y$ of variables in multivariate data. In this section, we restrict the input data set to a single variable, namely univariate data, to verify the effectiveness of the proposed model. In this case, Scalar-Graph has no global connection, and the loss function (Eq. 10) is reduced to $\mathfrak{L} = \mathfrak{L}_l$. We train the model and then use $Z$ to analyze the association (similarity) between scalar values. The similarity $sim(s_i, s_j)$ between the two scalar values is computed using the cosine similarity of $z_{s_i}$ and $z_{s_j}$ The matrix *sim* measures the similarity of any two scalar values, which can then be mapped onto the heatmap shown in the left figure of Fig. 5a. We filter out the negative values in *sim*, and the colors ranging

from light to dark in the figure indicate the range of similarities from 0 to 1. The scalar values are arranged by their horizontal and vertical coordinate, respectively.

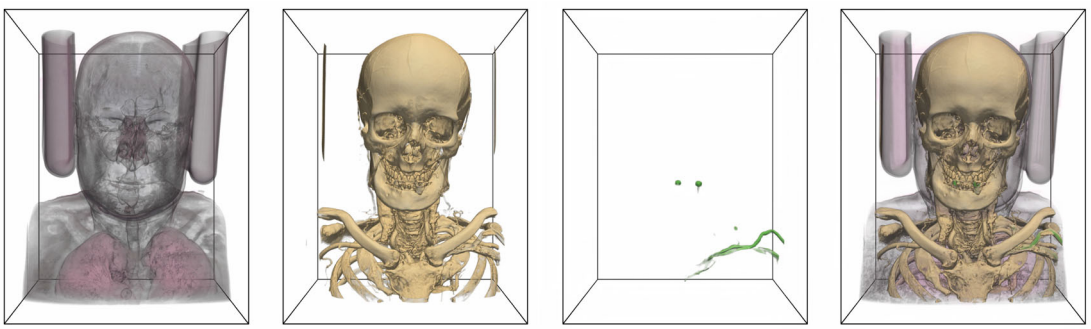
We use the CT MANIX data set as an example to analyze the similarity between scalar values, as shown in Fig. 5. It is clear from a proir understanding of CT MANIX that the volume can be divided into three parts based on scalar value: skin, bone, and implanted metal features. Therefore, after learning $Z$, the K-Means cluster algorithm Wong and Hartiganm (1979) is used to directly cluster the scalar values into three features. The silhouette coefficient Rousseeuw (1987) is a way of evaluating the clustering effect and applies to situations where the actual category information is unknown. The silhouette coefficient score is between




**(a)**




**(b)**



**(c)**

**Fig. 5** Validation of ScalarGCN. **a** shows the heatmap of *sim* matrix, silhouette coefficient figure, and scatter plot calculated based on ScalarGCN. **b** shows the similarity map, silhouette coefficient figure, and scatter plot calculated by ISM. In the scatter plot, only the scalar values that are multiples of 10 are marked to reduce visual occlusion. **c** shows the volume rendering results of the three features in the scatter plot of (a)

$[-1, 1]$, where a high value indicates that the object is well matched to its cluster and poorly matched to other clusters. In our evaluation, we use the silhouette coefficient to measure the quality of the clustering effect, and then to measure the quality of the representative vectors learned. We present the silhouette coefficient score in the middle figure of Fig. 5a. The vertical axis represents the categories with different scalar-value sets obtained by K-Means, the horizontal axis represents the ordered silhouette coefficient scores of scalar values, and the red dotted line represents the average silhouette coefficient scores when measuring the overall quality of the clustering result. To intuitively observe the distance distribution between scalar values, $Z$ is reduced to two dimensions using the t-SNE algorithm, as shown in the scatter plot of Fig. 5a. The scatter plot presents three distinct clusters colored in magenta, orange and green, where the points filled with the same color form a scalar value set, that is, a feature. We present the three features in parallel spatial views to intuitively verify the validity of ScalarGCN, as shown in the first three views of Fig. 5c, where the rendering colors are consistent with the those of the corresponding clusters. The last view of Fig. 5c shows the fusion rendering results of the features.

The spatial distribution of scalar values can be approximately equivalent to an isosurface. Therefore, we use the isosurface similarity (ISM) method Bruckner and Möller (2010) for comparison with the proposed methods. ISM uses the isosurface as the intermediate representation for each scalar value and applies the spatial proximity of isosurfaces to measure the similarity between scalar values of univariate data by geometric distance. It is completely different from our method, but we can still make comparisons because both methods try to measure the similarity between scalar values. The ISM method did not learn the representative vectors for scalar values, and so we directly take the distance field as the distributed representations to cluster scalar values using K-Means and then present the silhouette coefficient in Fig. 5b. Similarly, t-SNE is used to map the distance field to 2D space, and the scatter plot (the last figure of Fig. 5b) presents the scalar values in three clusters.

As can be seen from the two heatmaps, the two methods can yield roughly similar results, that is, the scalar values can be visually divided into three classes. This proves the usefulness of ScalarGCN. The specific difference lies in that ISM presents a more uniform similarity distribution while ScalarGCN presents a relatively concentrated distribution in the heatmap. ISM is sensitive to spatial location distribution because it uses the distance field to measure the similarity between isosurfaces, while ScalarGCN integrates the features, such as context numerical distributions, spatial distributions, and gradient distributions of scalar values, and thus pays more attention to mining the similarity of the spatial and numerical distribution of scalar values.

This example shows the quantitative similarity of scalar values and verifies the effectiveness of ScalarGCN in that it can yield results similar to ISM. In addition, this similarity can also be applied to automatic feature classification for univariate data.

#### *4.1.2 Multivariate data*

ScalarGCN learns the association between scalar values and spatial shapes. We use t-SNE to project $Z$ on the 2D plane to intuitively explore the association between scalar values of different variables. Each point represents a scalar value, and the distance between the points reflects the association between the corresponding scalar values.

The Hurricane Isabel data set in the 21-st time step has the following variables: SPEED (speed magnitude), P (pressure), TC (temperature), QGRAUP (graupel mixing ratio), QRAIN (cloud ice mixing ratio), QSNOW (snow mixing ratio), QVAROR (water vapor mixing ratio), and PRECIP (total precipitation mixing ratio, i.e., QGRAUP+QRAIN+QSNOW). We construct the Scalar-Graph corresponding to these variables, use ScalarGCN to learn the low-dimensional vector representations $Z$ of all scalar values, and then project all $z_{s_i}$ into the scatter plot, as shown in Fig. 6.

From the scatter plot, we find some interesting phenomena: (1) The low-dimensional vector representations of the scalar values of QRAIN and QSNOW show an approximately linear dependence in a local space, as shown in Fig. 6 Ⓐ. We show the spatial distribution for part A in Fig. 7(a-b). The two variables do have relatively consistent spatial distributions, such as in the regions outside the hurricane eye, but not in the center. The two variables are also consistent with the physical phenomenon that rain and snow in the sky are different forms of water molecules, and that the two are interchangeable under certain conditions, and thus should have similar behaviors. The scalar values corresponding to the two features are close and approximately linear in the scatter plot (Fig. 6 Ⓐ), because the scalar values are continuously distributed in

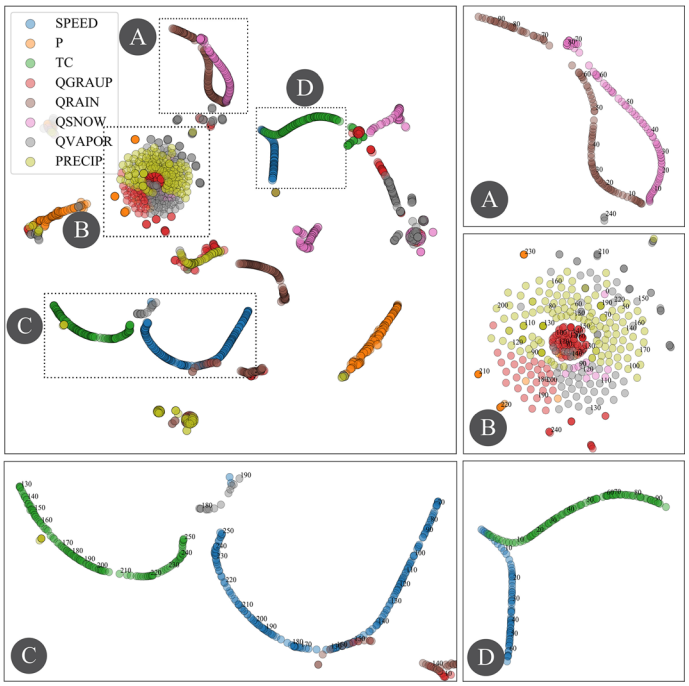



**Fig. 6** Association analysis between scalar values for the Hurricane Isabel data set. The points with the same color represent the scalar values of the same variable. We choose four local parts A, B, C and D, and present the enlarged scatter plots with the same labels

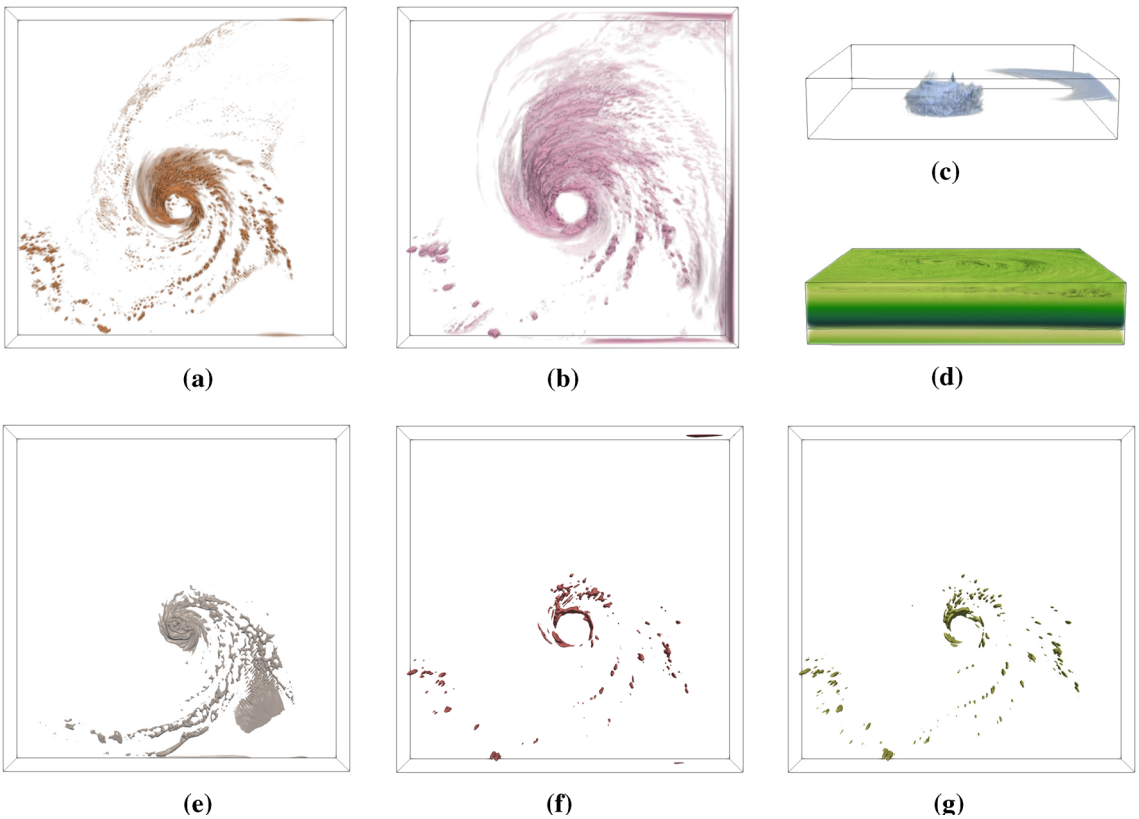



**Fig. 7** Spatial distributions correpsonding to Fig. 6 for the Hurricane Isabel data set. The rendering colors are the same as the variables' colors in Fig. 6. (**a**, **b**) shows the A part, (**c**, **d**) shows the C and D parts. (**e**–**g**) shows the B parts

the original volumes, and they have a high-frequency co-occurrence. (2) Speed and temperature are closely related, that is, there is a low temperature when the wind speed is low and a high temperature when the wind speed is high. The scalar values of both variables are divide into two parts, namely low temperature-speed, and high temperature-speed regions, as shown in Fig. 6 Ⓒ and Ⓓ. We know that as the altitude increases, the temperature decreases, and the wind speeds around the hurricane eye tend to be higher. ScalarGCN learns the combined features of high temperature and high speed, which gives us a strong inference that the hurricane eye is in a region with a low altitude. Note that we conclude this without knowing their spatial distribution. We present the spatial distributions of the scalar values of SPEED of part C, that is, the hurricane eye, in Fig. 7c. We also show the TC variable with the same viewpoint, in which the vertical direction represents the altitude and the colors changing from light to dark represents the temperature from low to high, as shown in Fig. 7d. The temperature distribution and the location of the hurricane eye confirm this conclusion. (3) The scalar values of QGROUP, QVAPOR and PRECIP are generally aggregate together (Fig. 6 Ⓑ), and the distributions present a scattering pattern without linear regularity. This is because the overall spatial distributions of the data of the three volumes are basically consistent (Fig. 7**e**–**g**), but their numerical distribution is relatively random. When training ScalarGCN, the global connection shortens the distance between the low-dimensional representations of variables because of the high-frequency co-occurrence in spatial distributions, while ScalarGCN detects nonlinear relationships between the scalar value in each variable because of the random context for scalar-values in the original volumes.

ScalarGCN ensures the comparability between the low-dimensional representations of different variables, because it learns the behaviors of scalar values in each variable through local connection and gradually pulls the low-dimensional representations of different variables in the same embedded space through global connection. Besides, it adaptively learns the mapping relationships from scalar-value features to variable features through the self-attention mechanism. The higher the co-occurrence frequency of scalar values of different variables, the greater the weight of global connection and the higher the correlation between them.

### 4.2 Association between variables

ScalarGCN learns the overall low-dimensional vector representations of variables. These representations provide an opportunity to study the associations between variables and then provide domain experts with more abundant association knowledge about volumes.

The association between variables mainly considers the interaction between variables, which is also an important research field in multivariate data He et al. (2019). Usually, statistical analysis and information theory methods are used to measure the overall associations.

In this section, we evaluate our method by comparing it with two methods: the Pearson correlation coefficient and the mutual information (MI). Similar to the association between scalar values, we use the cosine similarity to measure the association between variables.

A volume $v_a$ with a resolution of $500 * 300 * 300$ is constructed for comparative analysis. A sphere with the radius 100 and a numerical distribution range [1, 100] is located at the position (100, 150, 150). To avoid visual interference, we present the slice view of $x$-$z$ plane when $y = 150$, as shown in the sphere in Fig. 8. We then create the volumes where the spheres are the same size as $v_a$ but have a different numerical distribution ([101,200]) and are located at discrete positions (the black points in Fig. 8). The set of shifted volumes are denoted as $V_b$. For a volume $v_b \in V_b$, we regard $\{v_a, v_b\}$ as a bi-variate group. ScalarGCN models for all the bi-variate groups are trained to measure the association between volumes with different spheres' distances. The association computed from ScalarGCN is shown by the orange polyline in the chart in Fig. 8. To verify the effectiveness of the method, we compare our results with the Pearson correlation coefficient and the MI, as shown in the green and blue lines.

The Pearson correlation coefficient is only related to the numerical distributions, and the numerical distributions of $v_a$ and $v_b$ have a strictly consistent linear relationship. Therefore, the association value is always 1 no matter what the spatial position is, as shown in the chart in Fig. 8. To a certain extent, MI can detect the spatial overlap relationships. However, when the two spheres are tangent or separated from each other, it fails to detect the association between them, although they have a consistent spatial shape and gradient distribution. The global association based on ScalarGCN yields good results: a high association value will be assigned when the two spheres overlap in space; as the distance between the spheres gets longer, its association value gradually decreases. On the one hand, ScalarGCN aggregates the attributes of nodes and learns the context relationships through the weights of local connections, that is, it integrates the

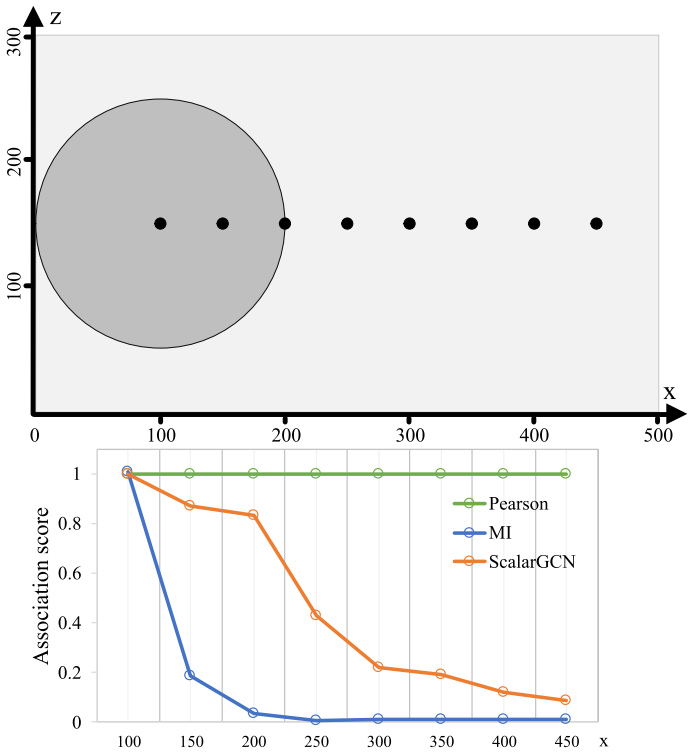



**Fig. 8** Validation of multivariate data. The top figure shows the generation of sphere volumes $v_a$ and $v_b$ with different spheres spatial positions in slice view. The bottom chart shows the association between $v_a$ and $V_b$ with different measurement methods

numerical and spatial distributions; on the other hand, it pulls in the distance between the low-dimensional representations of those scalar values with high-frequency co-occurrence in different variables through global connections. Thus, the closer two spheres are in spatial distribution, the more scalar-value pairs co-occur and the higher is the association. Even if the two spheres are far away from each other, the two volumes (A and B) still have scalar-value pairs with co-occurrence (such as the scalar-value pairs of the background of A and spheres of B). These regular scalar-value pairs and their context are used to adaptively learn the fusion mechanism between numerical features and variable features in the self-attention mechanism. Therefore, even if the two spheres are far apart, there are still association values detected. Thus, we verify the effectiveness of the variables' global embeddings of ScalarGCN, which embeds more than numerical distribution and spatial distribution, and provides a reasonable algorithm basis for downstream analysis tasks.

We then analyze the association between variables using the deep water impact ensemble data set Leistikow et al. (2019) that a 250-meter diameter asteroid initialized with a 45-degree momentum impacts into water. We choose the following six variable's in the 28516-th time step: v02 (volume fraction water), snd (sound speed in centimeters per second), tev (temperature in electronvolt), prs (pressure in microbars), rho (density in grams per cubic centimeter), and v03 (volume fraction of the asteroid). To intuitively explore the association between variables, we present the variable heatmap in Fig. 9a. To reduce visual occlusion, we present the slice views of the six variables.

The three methods all have similar association results, such as the strong association between v02 and snd, and the weak association between prs and v03. However, there are many differences between the three methods for different variables. First, MI can hardly distinguish the association between prs and rho, possibly because they have the same spatial shapes on the outer contour (shock wave), as shown in Fig. 9b. In contrary, the association between them computed by the Pearson correlation coefficient is almost 0. We found that there are certain spatial and numerical distribution relations between the two variables in the outer contour and below sea level. ScalarGCN detects and learns such relations well under the condition of distinguishing each other as much as possible. Additionally, we find that ScalarGCN detects v03 has a close relationship with v02, snd, and tev. The physical reason may be that the temperature around the asteroid's

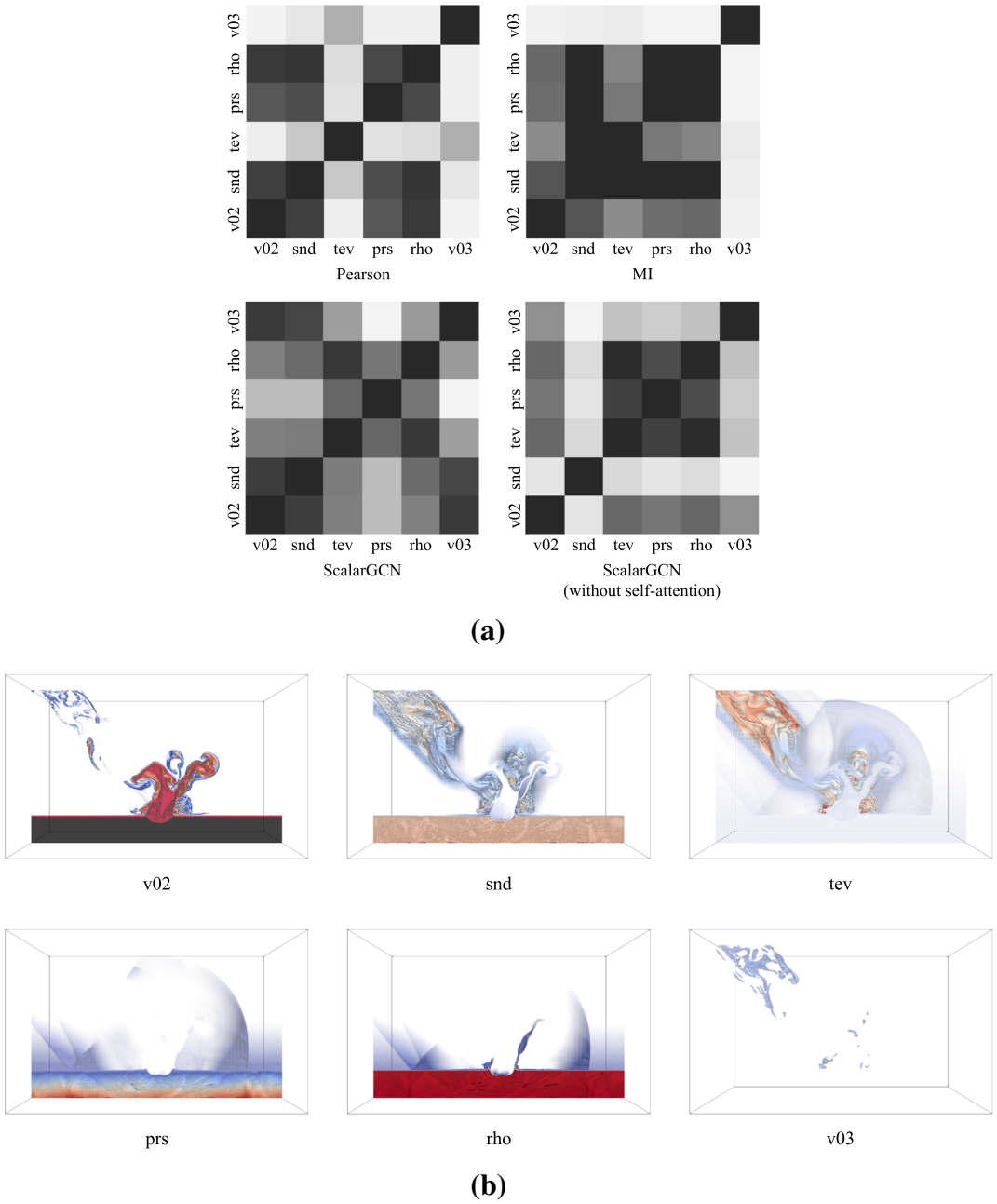



**Fig. 9** Association analysis between variables of the deep water impacted ensemble data set. **a** shows the variable heatmap with the method of the Pearson correlation coefficient, the mutual information, and ScalarGCN. The axes represent the arranged variables, respectively, and the colors range from light to deep to measure the associations between variables. **b** shows the slice views of the six variables when $z = 149$. The colors from cool to warm represent the scalar values from low to high

volume fraction is usually higher, and so it is related to sound velocity and water molecular content to a certain extent.

The most common aggregation operation for acquiring the low-dimensional representation of a variable is the mean aggregation:

$$y_v^{'} = \frac{1}{V} \sum_{s_i \in v} z_{s_i}. \tag{11}$$

**Table 1** Hyper-parameters' evaluation of initial features and the GCN layer number. The first character represents the criteria method silhouette coefficient. The capital letters *S*/*N*/*G* represents the context numerical distribution, spatial distribution and gradient distribution described in Sect. 3, respectively. The underlined values represent the maximum per initial feature combination, and the bold value represents the global maximum

| Initial features | Hidden layers | | | | |
|---|---|---|---|---|---|
| | [256] | [512, 256] | [1024, 512, 256] | [2048, 1024, 512, 256] | [4096, 2048,1024, 512, 256] |
| s-N | 0.693 | 0.716 | 0.728 | 0.721 | 0.709 |
| s-S | 0.748 | 0.803 | 0.842 | 0.818 | 0.761 |
| s-G | 0.732 | 0.775 | 0.783 | 0.770 | 0.753 |
| s-NS | 0.707 | 0.759 | 0.779 | 0.770 | 0.731 |
| s-NG | 0.712 | 0.734 | 0.755 | 0.742 | 0.718 |
| s-SG | 0.762 | 0.812 | **0.848** | 0.829 | 0.812 |
| s-NSG | 0.722 | 0.779 | 0.803 | 0.782 | 0.747 |

**Table 2** Training time for the above dataset (in seconds), where $m$ is the number of variables

| Dataset | $m$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Hurrican Isabel | 11.93 | 31.48 | 74.22 | 125.78 | 244.73 | 383.15 | 663.79 | 1102.56 |
| Deep Water Impacted | 11.69 | 36.88 | 105.46 | 208.62 | 287.11 | 437.07 | - | - |

Therefore, we added $y_v^{'}$, the low-dimensional representations of variable $v$ calculated by mean aggregation, as the baseline to verify the effectiveness of the self-attention mechanism. As shown in the last figure in Fig. 9a, snd is shown to have almost no association with other variables without the self-attention mechanism. However, this goes against common knowledge in physics, such as that the speed of sound should be higher in regions with a high number of water molecules. In this process of aggregation from the low-dimensional representations of scalar values to those of variables, a multi-head self-attention mechanism is introduced in Sect. 3.3 to adaptively learn the aggregation weight and selectively focus on important scalar values, such as the adaptive ignoring of noise and background and the increasing of the weight of important scalar values. Instead of only simple aggregation, the self-attention mechanism learns the weight of aggregation, screens useful features, and provides a more comprehensive aggregation mechanism and association between variables.

## 5 Discussion

We have introduced the learning process and application scope of ScalarGCN model. The training codes are implemented using the TensorFlow framework using the equipment of a GTX 1070 GPU. The training time ranges from a few seconds to a few minutes, which shows a positive correlation with the number of variables and the size of volume data, as shown in Table 2. There are some hyper-parameters not mentioned above that need to be evaluated. During experiments, we find that two hyper-parameters have strong influence on the overall results, namely, the selection of initial features and the number of GCN layers. We discuss the two hyper-parameters, the over-smoothing and under-fitting problem of ScalarGCN.

**Hyper-parameters**. We used the silhouette coefficient to verify the effectiveness of ScalarGCN in Sect. 4.1.1. We continue to use the CT MANIX data set with the prior knowledge of 3 clusters to evaluate ScalarGCN by measuring the quality of the clustering, as shown in Table 1. We find that the initial node feature without context numerical distributions achieves strong results on the evaluation criteria (the row that begins with "s-NG"), and that only with spatial distributions is the second (the row that begins with "s-S"). However, the inclusion of context numerical distributions gives worse results (the row that begins with "s-NSG"). Besides, great results are achieved for the three initial feature combinations when the hidden layer number is 3. Therefore, we fix the GCN hidden layer as 3 and only use the combination of $S$ and $G$ as the initial feature of nodes for training. Other hyper-parameters are not described in detail, such as that the learning rate is $10^{-4}$, the number of negative samples is 3, and the number of multi-head is 4.

**Over-smoothing**. Li et al. (2018) and Xu et al. (2018) pointed out that the GCN model could not be stacked as deep as the CNN model in visual tasks, and that there would be over-smoothing problems. That is, when using multiple GCN layers, the differentiation of nodes becomes worse and the learned representations of nodes tends to be consistent, which make the learning task more difficult. In Scalar-Graph, $N$ is small and the number of neighborhood nodes is large, so the over-smoothing problem is more serious. This can be seen in Table 1: when the number of hidden layers is greater than 3, the clustering quality decreases. Therefore, it is not recommended to use hidden layers with a number larger than 3 so as to avoid the over smoothing problem.

**Under-fitting**. ScalarGCN records the context numerical distribution through reasonable weight allocation on the edges of Scalar-Graph (*log* normalization in Sect. 3.1). Thus, it has more powerful learning and expression ability during the aggregation operation. In addition, adding the context numerical distributions to the initial feature increases the complexity of the first layer of GCN layers. With the increase in the number of trainable parameters and redundant information, it is difficult to achieve sufficient training (under-fitting) under the same number of epochs, so it will have relatively poor performance results. Therefore, in the training process, we do not consider adding the context numerical distributions to the initial feature of nodes.

## 6 Conclusion

In this paper, we provide the mapping mechanism of multivariate data to graph form (Scalar-Graph), and propose a graph neural network model composed of a three-layer GCN and a self-attention mechanism to study the low-dimensional vector representations of scalar values (local) and variables (global) in the same vector space. The low-dimensional vector representations are then applied for the exploration of the association between scalar values and variables.

In future work, we plan to extend ScalarGCN to learn the hidden information within time-varying ensemble data set. We would also like to apply ScalarGCN to the global association analysis of the volume data of multi-phase/multi-modal dental orthodontics.

**Acknowledgements** This work was supported by National Natural Science Foundation of China (61972343).

## References

Berger M, Li J, Levine JA (2019) A generative model for volume rendering. IEEE Trans Visual Comput Gr 25(4):1636–1650
Biswas A, Dutta S, Shen HW, Woodring J (2013) An information-aware framework for exploring multivariate data sets. IEEE Trans Visual Comput Gr 19(12):2683–2692
Bruckner S, Möller T (2010) Isosurface similarity maps. Comput Gr Forum 29(3):773–782
Carr H, Duke D (2014) Joint contour nets. IEEE Trans Visual Comput Gr 20(8):1100–1113
Carr H, Geng Z, Tierny J, Chattopadhyay A, Knoll A (2015) Fiber surfaces: generalizing isosurfaces to bivariate data. Comput Gr Forum 34(3):241–250
Gu Y, Wang C (2011) Transgraph: hierarchical exploration of transition relationships in time-varying volumetric data. IEEE Trans Visual Comput Gr 17(12):2015–2024
Guo H, Xiao H, Yuan X (2012) Scalable multivariate volume visualization and analysis based on dimension projection and parallel coordinates. IEEE Trans Visual Comput Gr 18(9):1397–1410
Haidacher M, Bruckner S, Gröller E (2011) Volume analysis using multimodal surface similarity. IEEE Trans Visual Comput Gr 17(12):1969–1978
Han J, Tao J, Wang C Flownet: A deep learning framework for clustering and selection of streamlines and stream surfaces. IEEE Trans Visual Comput Gr (2018)
Han J, Wang C (2020) Tsr-tvd: temporal super-resolution for time-varying data analysis and visualization. IEEE Trans Visual Comput Gr 26(1):205–215
Han J, Zheng H, Xing Y, Chen DZ, Wang C (2020) V2v: a deep learning approach to variable-to-variable selection and translation for multivariate time-varying data. IEEE Ann Hist Comput 01:1–1
He W, Wang J, Guo H, Wang KC, Shen HW, Raj M, Nashed Y, Peterka T (2020) Insitunet: deep image synthesis for parameter space exploration of ensemble simulations. IEEE Trans Visual Comput Gr 26(1)
He X, Tao Y, Wang Q, Lin H (2018) Biclusters based visual exploration of multivariate scientific data. Proceedings of IEEE Scientific Visualization Conference (SciVis) 2018:40–45
He X, Tao Y, Wang Q, Lin H (2018) A co-analysis framework for exploring multivariate scientific data. Vis Inform 2(4):254–263
He X, Tao Y, Wang Q, Lin H (2019) Multivariate spatial data visualization: a survey. J Visual 22(5):897–912
Kehrer J, Hauser H (2012) Visualization and visual analysis of multifaceted scientific data: a survey. IEEE Trans Visual Comput Gr 19(3):495–513

Kingma DP, Ba J Adam (2015) A method for stochastic optimization. In: International Conference on Learning Representation, pp. 1–15
Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
Leistikow S, Huesmann K, Fofonov A, Linsen L (2019) Aggregated ensemble views for deep water asteroid impact simulations. IEEE Comput Gr Appl 40(1):72–81
Li Q, Han Z, Wu XM (2018) Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32
Liu X, Shen HW (2015) Association analysis for visual exploration of multivariate scientific data sets. IEEE Trans Visual Comput Gr 22(1):955–964
Lu K, Shen HW (2017) Multivariate volumetric data analysis and visualization through bottom-up subspace exploration. Proceedings of IEEE Pacific Visualization Symposium (PacificVis) 2017:141–150
Lukasczyk J, Garth C, Weber GH, Biedert T, Maciejewski R, Leitte H (2019) Dynamic nested tracking graphs. IEEE Trans Visual Comput Gr 26(1):249–258
Lukasczyk J, Weber G, Maciejewski R, Garth C, Leitte H (2017) Nested tracking graphs. Comput Gr Forum 36(3):12–22
Porter WP, Xing Y, von Ohlen BR, Han J, Wang C (2019) A deep learning approach to selecting representative time steps for time-varying multivariate data. Proceedings of IEEE Scientific Visualization Conference (SciVis) 2019:40–45
Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
Sauber N, Theisel H, Seidel HP (2006) Multifield-graphs: an approach to visualizing correlations in multifield scalar data. IEEE Trans Visual Comput Gr 12(5):917–924
Sukharev J, Wang C, Ma KL, Wittenberg AT (2009) Correlation study of time-varying multivariate climate data sets. Proceedings of IEEE Pacific Visualization Symposium (PacificVis) 2009:161–168
Wang C, Tao J (2017) Graphs in scientific visualization: a survey. Comput Gr Forum 36(1):263–287
Wong AM, Hartiganm AJ (1979) Algorithm as 136: a k-means clustering algorithm. J Roy Stat Soc 28(1):100–108
Xu K, Li C, Tian Y, Sonobe T, Kawarabayashi Ki, Jegelka S (2018) Representation learning on graphs with jumping knowledge networks. In: International Conference on Machine Learning, pp. 5453–5462. PMLR