IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 23, 2019, accepted February 14, 2019, date of publication February 21, 2019, date of current version March 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900358

A Graph Layout Framework Combining t-Distributed Neighbor Retrieval Visualizer and Energy Models

GUANGLUAN XU[1,2], ZHENZHOU SONG[1,2,3], YANG WANG[1,2], DAOYU LIN[1,2], JING CHEN[1,2,3], TINGYUN MAO[1,2,3], AND WENJIA XU[1,2,3]

[1]Institute of Electronics, Chinese Academy of Sciences, Beijing 100049, China
[2]Key Laboratory of Network Information System Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing 100049, China
[3]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Zhenzhou Song (837429357@qq.com)

ABSTRACT Graph layout investigates the structure of the graph in order to better obtain the information implied in the graph. To solve the shortcomings of dimension reduction layouts on local adjustment and the insufficiency of energy models to maintain the overall structure of the graphs, this paper proposes a new graph layout framework called "tNEM" that layouts graphs by combining t-distributed neighbor retrieval visualizer (t-NeRV) and energy models. In the process of layout, our algorithm considers global and local structures at the same time. The layout results are more conform to aesthetic standards, meanwhile, maintain the structural information of the graph. We evaluate our algorithm on a wide variety of datasets and compare it with many other methods. We produce better visualization results than tsNET and tsNET* methods by reducing the tendency to crowd points together, and can better capture the global structure of the graph.

INDEX TERMS Graph layout, dimension reduction, energy model, t-NeRV.

I. INTRODUCTION

The input of graph layout is vertices and the relationship between vertices. The output are coordinates corresponding to the 2-dimensional or 3-dimensional space of each vertex. Through certain rules, some dense vertices can be visualized, which makes it easier to discover the relationships among vertices and obtain implicit information from vertices and edges. Evaluating the quality of graph visualization is subjective. It depends on the graph, and what information we want to extract [1]. When different information is extracted, different layouts are required.

In the past 30 years, many graph layout methods have been proposed. Most of them could be divided into 2 categories: force-directed [2], [3], and those based on dimension-reduction [4], [5].

In the force-directed methods, each node has an energy value or receives multiple forces by simulating a physical model, and the target is to minimize energy or achieve a force balance. The energy or forces can be expressed as a loss function, and the layout process is equivalent to minimizing the loss function. Force-directed methods are easy to understand and implement, but these methods pay more attention to the relationship between adjacent nodes, which often leads to the failure to maintain the global structure. Section II-A provides an overview of Kamada and Kawai (KK) [6], Fruchterman-Reingold (FR) [7] and LinLog [8] layout methods.

In dimension reduction (DR) methods, they map the graph into a high-dimensional space [5], and then reduce to low dimension for visualization. The main purpose is to capture high-dimensional space information in low-dimensional space [9]. DR layout methods mainly focus on preserving the global structural information of graphs but pay less attention to the local structure and the aesthetics of graphs, which makes it difficult to extract local information. Section II-B provides an overview of multidimensional scaling (MDS) series.

An excellent graph layout that meets aesthetic standards should satisfy the following criteria: reducing edge crossings, being symmetry, uniforming edge lengths, uniforming node distribution, separating non-adjacent nodes, and reducing nodes overlap [9]. This paper studies the layout method in 2-dimension space that conforms to aesthetic

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

standards called ''tNEM''(t-distributed Neighbor Retrieval Visualizer(t-NeRV) [10] and Energy Model). This algorithm optimizes a target function which combines both t-NeRV and energy model terms, where t-NeRV term retains the overall structure of the graph, and the energy term adjusts the position between the nodes. It is worth noting that the energy model here has a broader concept than that in the energy-based layout from force-directed layout, defined as any model that can be expressed as a scalar energy function. Furthermore, we modify the graph-theoretic distance (GTD) to make it have better properties and more suitable for graph layout, and provide theoretical basis.

To summarize, we make the following contributions:

- We propose a novel graph layout framework called the ''tNEM'' which combines t-NeRV and energy models. This method can simultaneously obtain the global and local structure of the graph.
- We use the modified GTD in the input space, which can make the node distribution more uniform and reduce the nodes congestion, and we give a theoretical proof.
- We use the cost function of the t-NeRV to graph layout so that we can better maintain the overall structure of the graph.

The rest of this paper is organized as follows: In section II, we review several classic graph layout methods. In section III, we present the tNEM algorithm that is based on the t-NeRV and energy models. Section IV presents the experimental results. Section V discusses energy model and the effect of modified GTD matrix parameters on the layout. Section VI gives conclusions.

II. RELATED WORK

A. FORCE-DIRECTED LAYOUTS

Force-directed algorithm has been developed for a long time. Tutte [11], [12] is the earliest research force-directed algorithm. The purpose of this method is to draw planar graphs and tri-connected graphs. Firstly, by fixing at least three vertices at the beginning, the other vertices are placed at the barycentre of their neighbors. After that, Eades proposed a based on spring forces graph layout algorithm [13], which transforms the layout problem into a mathematical optimization problem and derived many other layout algorithms. Force-directed algorithm is simple to implement, easy to understand, and has a good theoretical basis, hence most of the graph layout algorithms are based on force-directed algorithm. Although some multi-level layout algorithms such as GRIP [14], [15] and FM^3 [16] are proposed for the accelerated force-directed algorithm, these algorithms have little relevance to our method. Here we introduce several algorithms which are closely related to our proposed algorithm.

1) KK

The purpose of the KK method is to make the Euclidean distance of the node pair and the GTD a certain ratio [6] and to minimize the loss function by iteration.

2) FR

The FR method compares the network to a physical system in which nodes are represented as homopolar charges and edges are represented as strings with a fixed length of zero. Adjacent nodes create attractiveness, and all nodes generate repulsive forces. The layout process is looking for a balance between the repulsive force and attractiveness. The FR method always follows the principle that adjacent nodes are close to each other, and the moving distance is determined by the current temperature [1].

3) LinLog

Many real-world systems can be divided into several subsystems, in which the internal interaction of subsystems is relatively strong, and the interaction between subsystems is weak [8], [17], [18]. LinLog can show the network structure well and help produce readable visualizations. It contains node-repulsion LinLog, and edge-repulsion LinLog. LinLog method has several advantages over node grouping in revealing cluster structures. Instead of simply assigning the node to the cluster, they can display the degree of association between the node and its cluster, and the clarity of cluster separation. Actually, they promote the integration of clusters because viewers naturally interpret closely-positioned nodes as strongly correlated [19], [20].

Although force-directed algorithm is easy to implement, it can't get the structure information of the graph well, which can easily lead to the distortion of the graph.

B. DIMENSION REDUCTION LAYOUTS

DR is to represent high dimensional data in a low dimensional space while maintaining the relative distance between nodes [4], [21]. DR techniques applied in graph layout include linear DR, self-organizing graphs and MDS. Linear DR layout methods are to obtain low-dimensional coordinates, such as high dimensional embedding (HDE) [5], by linear DR after transformation of high-dimensional data. HDE uses principal component analysis (PCA) [22] technology for graph layout. Self-organizing graphs are layout graphs through neural networks, such as Bonabeau's [23], [24] method and Meyer's [25] inverted self-organizing map (ISOM). Next, we mainly introduce the MDS-based layout method.

The MDS method achieves the layout goal by minimizing the difference between the Euclidean and the GTD. There are two methods to minimize the difference: classical scaling and distance scaling. Classical scaling obtains an exact solution, such as PMDS (Pivot MDS) and Landmark MDS, by spectral decomposition. PMDS and Landmark MDS are sparse MDS methods that maintain the global structure while reducing complexity. Distance scaling directly calculates the difference between GTD and Euclidean distance in the layout, such as we-SNE [26] and tsNET [27]. The tsNET algorithm is the most relevant to our proposed algorithm. Here we mainly introduce tsNET algorithm.

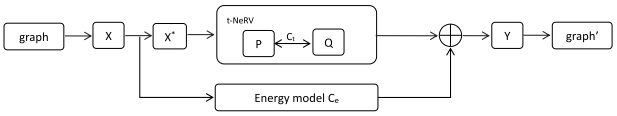


FIGURE 1. The tNEM algorithm overview.

1) tsNET

tsNET applies the t-SNE method to the graph layout by modifies the cost function to

$$C = \lambda_{KL} C_{KL} + \frac{\lambda_c}{2N} \sum_i \|y_i\|^2 - \frac{\lambda_r}{2N^2} \sum_{\substack{i,j \\ i \neq j}} log(\|y_i - y_j\| + \epsilon_r) \tag{1}$$

where the first item is the Kullback-Leibler divergence from t-SNE, the second item is compression item, and the last item is to prevent nodes from overlapping [27]. tsNET* are obtained by using the PMDS layout result instead of the random initialization coordinates of the tsNET method.

DR layout algorithm usually only focuses on the overall structure, and the phenomenon of edge crossing and node congestion is serious. Although tsNET algorithm has some improvements on the above problems, it is not good enough.

III. tNEM:t-DISTRIBUTED NEIGHBOR RETRIEVAL VISUALIZER AND ENERGY MODEL GRAPH LAYOUT ALGORITHM

In this paper, we propose a graph layout algorithm called ''tNEM'' which based on t-NeRV and energy models. We first discuss the impact of these two parts on graph layout separately and then introduce a simple way to combine them. Experiments show that our algorithm does well in maintaining the global structure and making the nodes distribution uniform.

A. ALGORITHM OVERVIEW

The framework of our algorithm can be seen in Fig. 1.

In the first step, we create a GTD matrix X and calculate the low-dimensional Euclidean distance matrix Y according to two-dimensional data $\mathcal{Y}$. By modifying X, we get modified GTD matrix X^*. Detailed descriptions of the matrix X and X^* are in section III-C. In the second step, we use the input matrix X^* to calculate conditional probabilities matrix P and Y to calculate conditional probabilities matrix Q. We get the Kullback-Leibler (KL) divergence of P and Q. Thirdly, we add KL divergence and energy terms to get the loss function. We get

$$C = C_t + C_e \tag{2}$$

where C_t is the KL divergence of P and Q, and C_e is the loss function of energy model.

Finally, we minimize loss function by momentum-based gradient descent.

B. t-NeRV

t-NeRV is the generalization of t-SNE. In the graph layout, t-NeRV does better than t-SNE in obtaining the global structure of the graph. The cost function of this method is the KL divergence C_t of the probability between high-dimensional and low-dimensional data points. Let d_{ij} be the GTD of vertices i and j, y be the coordinates of the nodes in the low-dimensional space. We get

$$C_t = \lambda_t \sum_i \kappa C_{KL}(P_{i:} \| Q_{i:}) + (1 - \kappa) C_{KL}(Q_{i:} \| P_{i:}) \tag{3}$$

where

$$C_{KL}(P_{i:} \| Q_{i:}) = \sum_{\substack{j \\ i \neq j}} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{4}$$

$$C_{KL}(Q_{i:} \| P_{i:}) = \sum_{\substack{j \\ i \neq j}} q_{ij} \log \frac{q_{ij}}{p_{ij}} \tag{5}$$

and the momentum gradient descent method is used to minimize C_t. In the formula 4 and 5,

$$p_{ij} = p_{ji} = \frac{p_{i|j} + p_{j|i}}{2N}, \quad p_{ii} = 0, \tag{6}$$

where the conditional probabilities

$$p_{j|i} = \exp(-\frac{d_{ij}}{2\sigma_i^2}) / \sum_{\substack{k \\ k \neq i}} \exp(-\frac{d_{ij}}{2\sigma_i^2}), \quad p_{i|i} = 0, \tag{7}$$

In the above formula, σ_i is obtained by making the perplexity $\mathcal{H}_i = 2^{-\sum_j p_{j|i} log_2 p_{j|i}} = Constant$ [28]. The probability of low-dimension space is given by

$$q_{ij} = q_{ji} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{\substack{k,l \\ k \neq i}} (1 + \|y_k - y_l\|^2)^{-1}}, \quad q_{ii} = 0, \tag{8}$$

which is a Student's t-distribution.

In formula 3, $\kappa \in [0, 1]$ a tradeoff parameter [10], [26], here we make $\kappa = 0.5$. In formula 3, $\sum_i \kappa C_{KL}(P_{i:} \| Q_{i:})$ is negatively correlated with precision and $\sum_i (1 - \kappa) C_{KL}(Q_{i:} \| P_{i:})$ is negatively correlated with recall [10]. Hence, the minimization of the cost function C_t is to find a tradeoff between precision and recall. We cannot in general reach the optimum of both simultaneously. When we minimize $\sum_i C_{KL}(P_{i:} \| Q_{i:})$, it will increase the proportion of nodes with small GTD in the Euclidean space as neighbors, but also incurs some misses. When we minimize $\sum_i C_{KL}(Q_{i:} \| P_{i:})$,it will reduce the proportion of nodes that have large GTD in the European space as neighbors, but the number of correct nodes will also decrease.

The layout results using the t-SNE and t-NeRV methods are shown in Fig. 2. We can see that using t-NeRV instead of t-SNE can better obtain the global structure of the graph and produce better layout results.

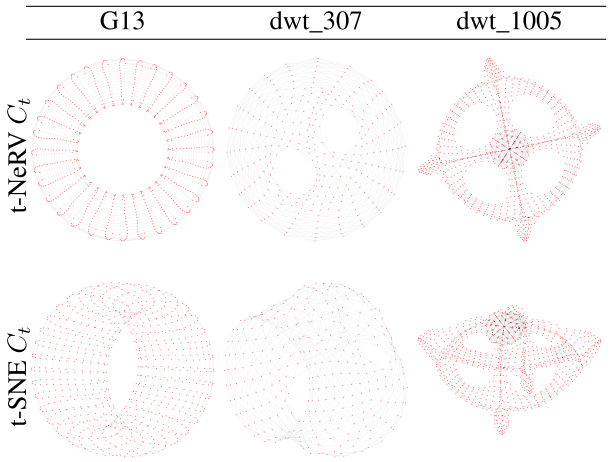


FIGURE 2. Visual comparison of t-NeRV and t-SNE. Rows and columns represent different graphs and methods.

C. MODIFIED GRAPH-THEORETIC DISTANCE MATRIX

Although the GTD matrix can maintain the overall structure of the graph to some extent, it often causes nodes to be crowded together and even cause distortion of the graph. This will make the GTD not proportional to the Euclidean distance. As the GTD increases, the Euclidean distance in the low-dimensional space increases sharply. When we use the GTD matrix X to calculate the probability matrix P, the magnitude of probability is negatively correlated with the vertex distance. And the probability p_{ij} decreases sharply as the GTD increases, causing the distance also increase sharply in the low-dimensional space. When we try to model the distances from vertex i to other vertices in a two-dimensional graph layout space, we get the following ''crowding problem'': if we want to lay out according to the GTD in the low-dimensional space, the nodes with small distances will be crowded together, and the nodes with large distances will be very far apart, resulting in the GTD being out of proportion with the Euclidean distance. And GTD matrix X does not consider the distribution relationship between nodes. Therefore, we modified X to be

$$X^* = X^a \tag{9}$$

where we let $a = 0.5$. As the GTD increases, the probability p^* corresponding to X^* changes more smoothly than the probability p corresponding to X, and the layout of the graph is more uniform, rather than being crowded together or far away.

To illustrate the properties of X^*, we give a mathematical proof, which is mainly divided into three parts: (1)

1) Find the probability matrices P and P^* corresponding to X and X^*.
2) Based on the property of KL divergence when KL divergence is minimized, we find Euclidean distance matrix Z and Z^* corresponding to X and X^* by assuming $P = Q$ and $P^* = Q^*$, where Z and Z^* are related to the variance $2\sigma^2$ of Gaussian function.
3) By discussing the variance and the distribution of the graph, we get the properties of X^*. Below we give the detailed process of proof.

Given a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges between the vertices. Let N be the number of vertices, d_{ij} be the GTD of vertices i and j, md be the maximum GTD of the graph G, $z(d_{ij} = n)$ be the Euclidean distance of the nodes i and j with a GTD n in the layout space. Let $P_X = P, P_{X^*} = P^*$. Now we derive the relationship between $\big(z(d_{ij} = 1), z(d_{ij} = 2), \ldots, z(d_{ij} = md)\big)$ and $\big(z^*(d_{ij} = 1), z^*(d_{ij} = 2), \ldots, z^*(d_{ij} = md)\big)$ to illustrate the difference between the nature of X^* and X.

We use the symmetrized conditional probabilities

$$p_{ij} = p_{ji} = \frac{p_{j|i} + p_{i|j}}{2N}, \quad p_{ii} = 0, \tag{10}$$

as the similarity between vertices x_i and x_j, and the conditional probabilities $p_{j|i}$ and $p_{i|j}$ are defined differently in X and X^*. In X, the similarity of vertex x_i to x_j is

$$p_{j|i} = \exp(-\frac{d_{ij}^2}{2\sigma_i^2}) / \sum_{\substack{k \\ k \neq i}} \exp(-\frac{d_{ik}^2}{2\sigma_i^2}), \quad p_{i|i} = 0 \tag{11}$$

In X^*, the similarity of vertex x_i to x_j is

$$p_{j|i}^* = \exp(-\frac{d_{ij}}{2\gamma_i^2}) / \sum_{\substack{k \\ k \neq i}} \exp(-\frac{d_{ik}}{2\gamma_i^2}), \quad p_{i|i}^* = 0 \tag{12}$$

By proving the relationship between conditional probabilities $p_{j|i}$ and $p_{j|i}^*$ or the relationship between $p_{i|j}$ and $p_{i|j}^*$, we can get the relationship between probabilities p_{ij}^* and p_{ij} corresponding to X^* and X. We optimize the loss function by minimizing the KL divergence of P and Q. Assuming that the final solution of the iteration is $P = Q$, i.e. $p_{j|i} = q_{j|i}$ and $2\sigma_i^2 = \beta_i$, we get

$$\begin{aligned}
&\left(\frac{p(d_{ij}=1)}{p(d_{ik}=2)}, \frac{p(d_{ij}=1)}{p(d_{ik}=3)}, \ldots, \frac{p(d_{ij}=1)}{p(d_{ik}=md)}\right) \\
&= \left(\frac{q(d_{ij}=1)}{q(d_{ik}=2)}, \frac{q(d_{ij}=1)}{q(d_{ik}=3)}, \ldots, \frac{q(d_{ij}=1)}{q(d_{ik}=md)}\right) \\
&= \left(e^{\frac{3}{\beta_i}}, e^{\frac{8}{\beta_i}}, \ldots, e^{\frac{md^2-1}{\beta_i}}\right) = \left(\frac{1+z_2^2}{1+z_1^2}, \frac{1+z_3^2}{1+z_1^2}, \ldots, \frac{1+z_{md}^2}{1+z_1^2}\right)
\end{aligned} \tag{13}$$

Similarly, from $P^* = Q^*$ and let $2\gamma_i^2 = \alpha_i$, we get

$$\left(e^{\frac{1}{\alpha_i}}, e^{\frac{2}{\alpha_i}}, \ldots, e^{\frac{md-1}{\alpha_i}}\right) = \left(\frac{1+z_2^2}{1+z_1^2}, \frac{1+z_3^2}{1+z_1^2}, \ldots, \frac{1+z_{md}^2}{1+z_1^2}\right), \tag{14}$$

Let $y_1 = y_1^* = (e-1)^{0.5}$, we get

$$\begin{aligned}
&\big(z(d_{ij}=1), z(d_{ij}=2), \ldots, z(d_{ij}=md)\big) \\
&= \left((e-1)^{0.5}, (e^{\frac{3}{\beta_i}+1}-1)^{0.5}, \ldots, (e^{\frac{md^2-1}{\beta_i}+1}-1)^{0.5}\right)
\end{aligned} \tag{15}$$

$$\begin{aligned}&\left(z^*(d_{ij}=1), z^*(d_{ij}=2), \ldots, z^*(d_{ij}=md)\right)\\&=\left((e-1)^{0.5}, (e^{\frac{1}{\alpha_i}+1}-1)^{0.5}, \ldots, (e^{\frac{md-1}{\alpha_i}+1}-1)^{0.5}\right)\end{aligned} \tag{16}$$

When we get the relationship of α_i and β_i, we can get the relationship between z and z^*. Here we discuss the value of β_i.

The parameters α_i and β_i of the Gaussian function can be solved by making the perplexity of each line of the matrix P constant, which is specified by the user [29]. From the perplexity $Perp(P_i) = 2^{-\sum_j p_{j|i}\log_2 p_{j|i}}$, we get

$$\begin{aligned}\log_2 Perp(P_i) &= -\sum_j p_{j|i}\log_2 p_{j|i}\\&= -\sum_j \frac{e^{-\frac{d_{ij}^2}{\beta_i}}}{\sum_j e^{-\frac{d_{ij}^2}{\beta_i}}}\log_2\frac{e^{-\frac{d_{ij}^2}{\beta_i}}}{\sum_j e^{-\frac{d_{ij}^2}{\beta_i}}}\end{aligned} \tag{17}$$

$$\begin{aligned}\log_2 Perp(P_i^*) &= -\sum_j p_{j|i}^*\log_2 p_{j|i}^*\\&= -\sum_j \frac{e^{-\frac{d_{ij}}{\alpha_i}}}{\sum_j e^{-\frac{d_{ij}}{\alpha_i}}}\log_2\frac{e^{-\frac{d_{ij}}{\alpha_i}}}{\sum_j e^{-\frac{d_{ij}}{\alpha_i}}}\end{aligned} \tag{18}$$

Let Formula 17 = Formula 18, $\alpha_i = 1, \beta_i = \tau\alpha_i$, and $n(d_{i*} = k)$ is the number of vertices with a distance k from i. Next, we list several situations to illustrate.

1)$n(d_{i*} = 1) = N - 1$: For any $\tau > 0$, the Formula 17 = Formula 18 makes sense.

2)$n(d_{i*} = 2) = k$, $n(d_{i*} = 1) = N - 1 - k$, where $0 < k \leqslant N - 2$. We get

$$\begin{aligned}&-\frac{(N-1-k)e^{-1}}{ke^{-2}+(N-1-k)e^{-1}}\log_2\frac{e^{-1}}{ke^{-2}+(N-1-k)e^{-1}}\\&-\frac{ke^{-2}}{ke^{-2}+(N-1-k)e^{-1}}\log_2\frac{e^{-2}}{ke^{-2}+(N-1-k)e^{-1}}\\&=-\frac{(N-1-k)e^{-\frac{1}{\tau}}}{ke^{-\frac{4}{\tau}}+(N-1-k)e^{-\frac{1}{\tau}}}\log_2\frac{e^{-\frac{1}{\tau}}}{ke^{-\frac{4}{\tau}}+(N-1-k)e^{-\frac{1}{\tau}}}\\&\quad-\frac{ke^{-\frac{4}{\tau}}}{ke^{-\frac{4}{\tau}}+(N-1-k)e^{-\frac{1}{\tau}}}\log_2\frac{e^{-\frac{4}{\tau}}}{ke^{-\frac{4}{\tau}}+(N-1-k)e^{-\frac{1}{\tau}}}\end{aligned} \tag{19}$$

The right side of Equation 19 is a monotonically increasing function for τ. We can get the equation with one and only one solution. We solved $\tau = 3$.

3)$n(d_{i*} = 3) = m$, $n(d_{i*} = 2) = k$, $n(d_{i*} = 1) = N - 1 - k - m$, where $0 < k \leqslant N - 3$, $0 < m \leqslant N - 2 - k$. Similar to Equation 19, we get

$$f(\alpha_i, k, m) = f^*(\beta_i, k, m) \tag{20}$$

In Equation 20, when $n(d_{i*} = 3)/n(d_{i*} = 2)$ approaches zero, as $n(d_{i*} = 3)/n(d_{i*} = 2)$ increases, τ monotonically increases, and $\lim \tau = 4$. When $n(d_{i*} = 3)/n(d_{i*} = 2)$ approaches infinity, as $n(d_{i*} = 3)/n(d_{i*} = 2)$ increases, τ monotonous Incremental, and $\lim \tau = 5$, so we get $3 < \tau < 5$.

TABLE 1. Edge length relationship of z and z^*. Rows and columns represent the ratio of edge lengths and different τ values, respectively.

	z_1^*/z_1	$(z_2-z_1)/z_1$	$(z_3-z_2)/z_1$	$(z_3-z_2)/(z_2-z_1)$	$\frac{z_2^*-z_1^*}{z_1^*}\Big/\frac{z_2-z_1}{z_1}$	$\frac{z_3^*-z_2^*}{z_3^*-z_1^*}\Big/\frac{z_3-z_2}{z_2-z_1}$	$\frac{z_3^*-z_2^*}{z_1^*}\Big/\frac{z_3-z_2}{z_1}$
3.5	1	0.77	2.10	2.72	1.20	0.56	0.67
4.0	1	0.66	1.66	2.50	1.39	0.61	0.85
4.5	1	0.58	2.38	2.38	1.59	0.64	1.02
4.8	1	0.54	1.24	2.30	1.70	0.67	1.13

TABLE 2. Edge length relationship of z^*.

$(z_2^*-z_1^*)/z_1^*$	$(z_3^*-z_2^*)/z_1^*$	$(z_3^*-z_2^*)/(z_2^*-z_1^*)$
0.92	1.41	1.53

Tables 1 and 2 record the relationship between edges length corresponding to different τ. From Tables 1 and 2, comparing $(z_2^*-z_1^*)/z_1^*$ and $(z_2-z_1)/z_1$, $(z_3^*-z_2^*)/z_1^*$ and $(z_3-z_2)/z_1$, we can find that X^* always makes the length of the edge more uniform than X, which makes the layout match the aesthetic standards better. From the discussion of τ above we can know that when the proportion of vertices with a large GTD from the vertex i is increased, τ will become larger. As can be seen from Table 1, with the increase of τ, $\frac{z_2^*-z_1^*}{z_1^*}\Big/\frac{z_2-z_1}{z_1}$ and $\frac{z_3^*-z_2^*}{z_1^*}\Big/\frac{z_3-z_2}{z_1}$ are monotonically increasing, which will solve the "crowding problem" of the vertices. When $\tau = 4.8$, we can get the same conclusion by comparing $(z_2^*-z_1^*)/z_1^*$ and $(z_2-z_1)/z_1$, $(z_3^*-z_2^*)/z_1^*$ and $(z_3-z_2)/z_1$. The situation discussed above can also be extended to the situation of $n(d_{i*} > 3) \neq 0$.

Therefore, we get the conclusion that X^* will make the layout more uniform and reduce the "crowding problem" of vertices. As shown in Fig. 3, when the exponent parameter of X increases, the nodes with larger degrees are always gathering together with their neighbors.

D. ENERGY MODELS

Although the t-NeRV method can obtain the global structure of the graph, the local structure is not well adjusted, we optimize the part of the graph through the energy model to get a better layout. We give three examples of energy models: 1) a modified KK model, 2) a modified FR model, and 3) a modified Linlog model. We have added compression items in the above three methods [27], [28]. It turns out that the graph layout is better when adding the compression item.

1) EXAMPLE 1: tNEM-KK

In this example, we make the C_e term equal to

$$\begin{aligned}&\frac{\lambda_c}{2N}\sum_i\|y_i\|^2+\frac{5\lambda_e}{2N^2}\sum_{\substack{i,j\\i\neq j}}\frac{1}{d_{ij}^2}(\|y_i-y_j\|-d_{ij})^2\\&\qquad+\frac{5\lambda_e}{2N^2}\sum_{d_{ij}=1}(\|y_i-y_j\|-1)^2\end{aligned} \tag{21}$$

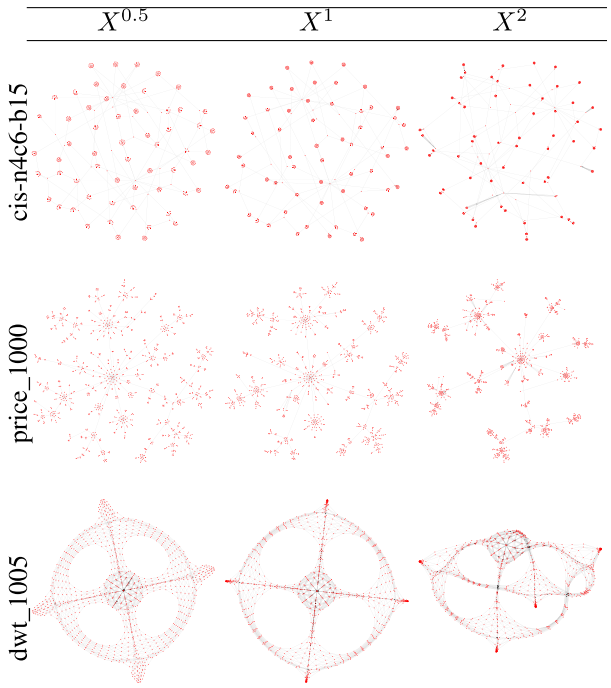


FIGURE 3. Compare the effects of different adjacency matrices on the layout of the graph, where cis-n4c6-b15 and price_1000 use the tNEM-FR layout and dwt_1005 use the tNEM-KK layout.

In function 21, the first part is early compression. In graph layout, early compression is done by adding an additional L2-penalty to the cost function and it is proportional to the sum of squared distances of the map points from the origin. The size of the penalty item is manually set, but it is fairly robust in this optimization parameter [28]. The second part is a modified KK term. In KK methods, the cost function is $E = \sum_{i-1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{2} * \frac{K}{d_{ij}^2} (|y_i - y_j| - \frac{L_0}{\max\limits_{i<j} d_{ij}} * d_{ij})^2$. We make $K = 2$, $L_0 / \max\limits_{i<j} d_{ij} = 1$. The third term reduce the distance between adjacent nodes.

2) EXAMPLE 2: tNEM-FR

In this example, we make the C_e term equal to

$$\frac{\lambda_c}{2N} \sum_i \|y_i\|^2 + \frac{\lambda_e}{2N^2} \sum_{d_{ij}=1} \frac{1}{3} (\|y_i - y_j\|)^3 - \frac{\lambda_e}{2N^2} \sum_{\substack{i,j \\ i \neq j}} \log_2(\|y_i - y_j\| + \epsilon) \quad (22)$$

The first part is early compression term. The second part is attractive forces. The third part is repulsive forces. In FR method, the attractive forces $f_a(z) = z^2/k$ and the repulsive forces $f_r(z) = k^2/z$ [7] are proportional to the distance the vertex moves, which is the same as the derivative of the cost function. Therefore, we add the integral of the force in the FR method to the cost function. This is exactly the same as the third term of the cost function of the tsNET method in [27]. The cost function in [27] is only a special case of the tSEM-FR method, which makes the attraction coefficient zero.

3) EXAMPLE 3: tNEM-LinLog

LinLog is divided into two types: node-repulsion and edge-repulsion LinLog energy model [8]. In node-repulsion, we make the C_e term equal to

$$\frac{\lambda_c}{2N} \sum_i \|y_i\|^2 + \frac{\lambda_e}{2N^2} \sum_{d_{ij}=1} (\|y_i - y_j\|) - \frac{\lambda_e}{2N^2} \sum_{\substack{i,j \\ i \neq j}} \ln(\|y_i - y_j\| + \epsilon) \quad (23)$$

In edge-repulsion, we make the C_e term equal to

$$\frac{\lambda_c}{2N} \sum_i \|y_i\|^2 + \frac{\lambda_e}{2N^2} \sum_{d_{ij}=1} (\|y_i - y_j\|) - \frac{\lambda_e}{2N^2} \sum_{\substack{i,j \\ i \neq j}} deg(i) deg(j) \ln(\|y_i - y_j\| + \epsilon) \quad (24)$$

Among the above three methods, the KK and tNEM methods have the best compatibility, because both methods make the GTD proportional to the Euclidean distance. The edge-LinLog method has the worst compatibility with tNEM and will destroy the global structure of the graph obtained by tNEM. When modifying the FR method, we find that FR and LinLog methods have great similarities. When the force of a vertex in the FR method is integrated, a cost function similar to the LinLog method is obtained. We also found that the tsNET and tsNET* methods are a special case of the modified FR and node-LinLog methods.

When combined with the energy models, the graph layout obtained by the tNEM method can be locally adjusted to make the layout more conform to aesthetic standards, taking into account the global and local structures at the same time. Fig. 4 shows the effect of different energy model coefficients on the graph layout. It can be seen that when the coefficient increases, the congestion between nodes in the layout results will decrease. Experimental comparison and detailed analysis of the above three methods are shown in Section IV.

We minimize the total cost function of C by momentum-based gradient descent. Same as tsNET [27] method, our method is divided into 3 steps. First, the coordinates y_i are randomly initialized. Second, we set the parameter $(\lambda_t, \lambda_c, \lambda_e) = (1, 0.25, 0)$ to minimize the cost function. Finally, we set $(\lambda_t, \lambda_c, \lambda_e) = (1, 0.01, \lambda_e)$, where λ_e is a adjustable coefficient and different for different graphs. In the second step, we use the t-NeRV term to find the overall structure of the graph and use the compression term to compress the coordinates to get a rough layout. The third step is to add energy terms and adjust some internal structures of the graph to get an ideal layout.

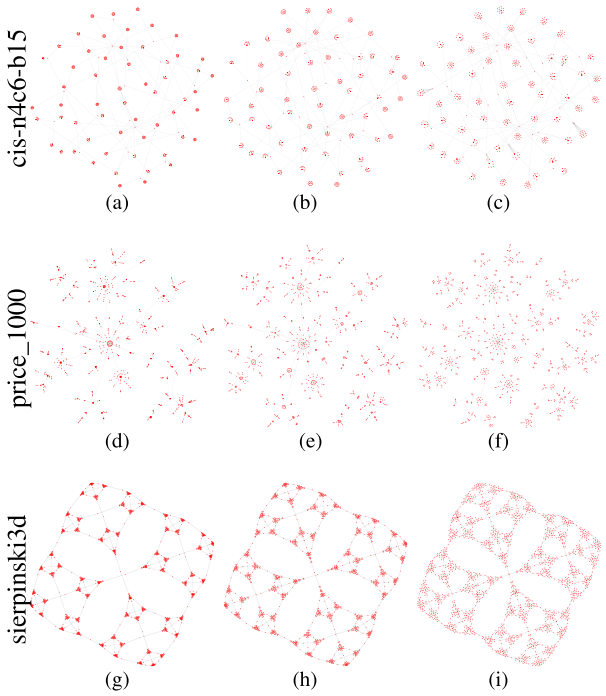


FIGURE 4. Compare the influence of different force-directed coefficients on the layout of the graph, where cis-n4c6-b15 and price_1000 use tNEM-FR layout, sierpinski3d use tNEM-KK layout. (a) $\lambda_e = 4$. (b) $\lambda_e = 10$. (c) $\lambda_e = 20$. (d) $\lambda_e = 1$. (e) $\lambda_e = 4$. (f) $\lambda_e = 10$. (g) $\lambda_e = 4$. (h) $\lambda_e = 10$. (i) $\lambda_e = 30$.

IV. EXPERIMENTS

We evaluate our methods through visual comparisons and two quantitative indicators comparing with some related approaches. The experiments data are from several 2D, including non-connected graphs, and we also give the run time of all the methods.

A. BENCHMARK AND EVALUATION METRIC

1) DATA SETS

In Table 3, we present the data for testing, which contains multiple types of data, with different dimensions, different structures, both synthetic and real data. Some of these data are collected from Harwell-Boeing collection [30]. Some are collected from Florida collection [31] and network repository [32], which contain many real and synthetic data. The rest of the graphs are using the data in paper [27]. The same dataset can be found from different places, we just explain the location we directly reference.

2) COMPARED ALGORITHMS

We compare tNEM to some related algorithms. In Table 4, we list those algorithms. All of these methods are closely related to our method, where our method is based on improvements in tsNET and tsNET*. In our method, the force-directed term has different coefficients for different graph structures, but it is very stable.

TABLE 3. Sizes,source and types of the graphs used in our benchmark.

Name	$\|V\|$	$\|E\|$	Source	Type
n4c6-b1	210	408	[32]	Miscellaneous
grid1	252	478	[33]	Mesh-like,planar,structural
dwt_307	307	1108	[30]	Mesh-like,planar,structural
problem1	415	1182	[32]	Mesh-like, planar,structural
dwt_419	419	1572	[30]	Structural
n4c5-b10	630	1317	[32]	Miscellaneous
G13	800	1600	[32]	Mesh-like,3D
cis-n4c6-b15	920	959	[32]	Miscellaneous
L	956	1820	[32]	Mesh-like,planar,structural
price_1000	1000	999	[33]	Planar,tree
dwt_1005	1005	3808	[30]	Structural
tols1090	1090	2249	[32]	Planar,Non-connected,tree
G34	2000	4000	[32]	Mesh-like,3D
sierpinski3d	2050	6144	[34]	Structural

TABLE 4. The class, source, and implementation of the comparison method.

class	Name	References	Implementation
Force-directed	KK	[6]	Tulip4.8.1 [35]
	FR	[7]	Tulip4.8.1 [35]
	Linlog	[8]	Tulip4.8.1 [35]
	GRIP	[14], [15]	Tulip4.8.1 [35]
Dimension reduction	PMDS	[4]	Tulip4.8.1 [35]
	tsNET	[27]	Author [27]
	tsNET*	[27]	Author [27]

3) EVALUATION METRICS

We use two quantitative metrics to measure the graph layout method, where the normalized stress metric is used to measure the nature of distance preservation, and the neighborhood preservation metric is used to measure the proportion of neighbor retention after layout. Below we introduce separately.

Normalized stress metric:

$$\sigma = \frac{1}{|V|^2 - |V|} \sum_{\substack{i,j \\ i \neq j}} \left(\frac{d(x_i, x_j) - \|y_i - y_j\|}{d(x_i, x_j)} \right)^2 \tag{25}$$

where V is the number of nodes, $d(x_i, x_j)$ is the shortest path between nodes i and j, and $\|y_i - y_j\|$ is the Euclidean distance in low-dimensional space.

Neighborhood preservation metric:

$$v = \frac{1}{|V|} \sum_{i} \frac{|N_G(x_i, r_G) \cap N_Y(y_i, k_i)|}{|N_G(x_i, r_G) \cup N_Y(y_i, k_i)|} \tag{26}$$

where $N_G(x_i, r_G) = \{x_j \in V | d_{ij} \leqslant r_G\}$, which is the set of nodes whose GTD from vertex x_i that are less than r_G. $N_Y(y_i, k_i)$ is the k-nearest neighbors of point y_i after graph layout, where $k = N_G(x_i, r_G)$.

In Equation 26, v is twice the F1-measure of the vertex, where $N_G(x_i, r_G)$ is the relevant positives, $N_Y(y_i, k_i)$ is the non-relevant negatives, $|N_G(x_i, r_G) \cap N_Y(y_i, k_i)|$ is true positives. In F1-measure, precision rate $P = (|N_G(x_i, r_G) \cap N_Y(y_i, k_i)|)/N_G(x_i, r_G)$, recall rate $R = (|N_G(x_i, r_G) \cap N_Y(y_i, k_i)|)/N_Y(y_i, k_i)$. From $F1 = 2PR/(P + R)$, $F1 = 2v$ can be obtained.

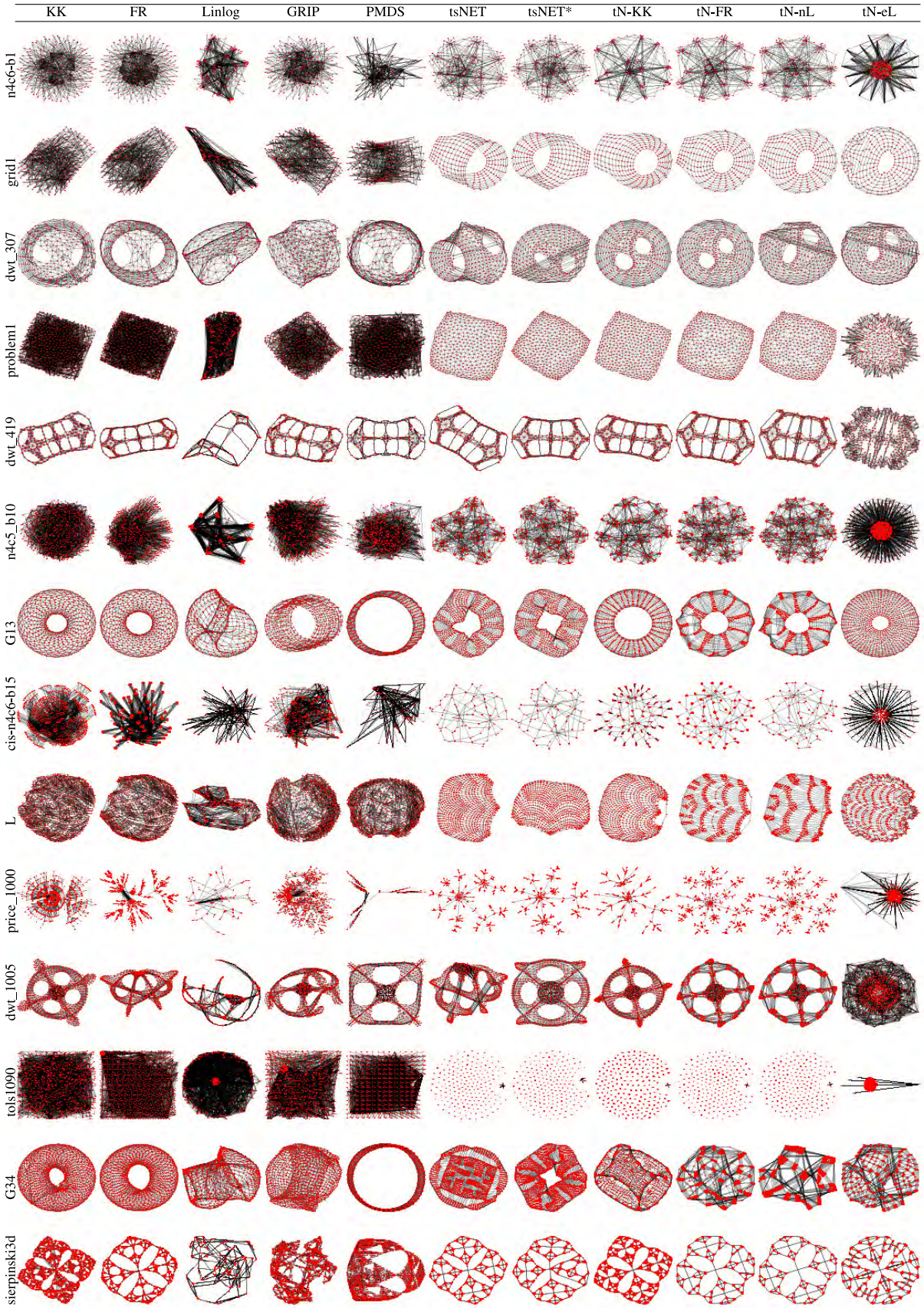


FIGURE 5. Visual comparison of multiple methods and graphs, rows and columns representing different graphs and methods.

B. VISUAL COMPARISON

In Fig. 5, we compare the visual of different layout methods. According to G13 and G34, the tNEM methods always distributes 3D nodes into a plane, because our method considers the global structure better than the traditional layout method, not limited to neighbor nodes. Therefore, We can conclude that the tNEM method is not suitable for sample data generated in high-dimensional space, especially for mesh-like graphs. Through different types of multiple graphs, we can find that our method has achieved better visual layouts, such as grid1, dwt_307, cis-n4c6-b15, L. Since the force-directed term in the tsNET and tsNET* methods are a special case of the tNEM-nodeLinLog and tNEM-FR methods. Therefore, the above two methods produce similar layouts to the tNEM graph layout. Because the LinLog method is not intended to obtain a map layout that conforms to aesthetic standards, the layout of the tNEM-LinLog method is not as good as tNEM-KK and tNEM-FR, such as L, dwt_1005, n4c5-b10, dwt_307. We can see that our method always produces an excellent layout for a graph with a certain structure(tree, mesh-like, planar). We also tested our method with the non-connected graph tols1090. We can see that in addition to the tNEM-edgeLinLog method, the layout method based on the t-NeRV always makes the non-connected nodes relatively distant.

C. DISTANCE PRESERVING METRIC

We use normalized stress metric σ(Equation 25) [27], [36], [37] to assess the nature of distance preserving. Table 5 lists the σ corresponding to different methods. As is shown in the table we can see that the t-NEM-KK method works best because both the KK and the t-NEM method are designed to make GTD proportional to the Euclidean distance, which means the property compatibility is excellent. The tNEM-KK and tNEM-FR methods usually get better results than other methods in the table. But for some specific structure graphics, the KK method results will be better. Since the purpose of the LinLog method is not to make the normalized stress metric effect better, when the t-SNE method is combined with the LinLog layout, it does not produce a good layout effect. It can be seen from the normalized stress metric σ values corresponding to tNEM-nodeLinLog and tNEM-edgeLinLog that the properties of LinLog are embodied in these two methods, and the purpose is not to reduce the stress σ value. This property is not compatible with the nature of the t-SNE method to find the overall structure, hence the layout is not ideal.

D. NEIGHBORHOOD PRESERVING RATIO

We use neighborhood preservation metric v (Equation 26) [27] to assess the nature of neighborhood preserving. From Table 6, we can find that tsNET and tsNET* methods get better results than our proposed method, but the difference is very small because energy terms always make the nodes more evenly distributed, which will increase the wrong neighbors. When we reduce the coefficient of the energy terms, in addition to the tNEM-edgeLinLog method, the value v of the neighbor reservation matrix obtained by the tNEM method is basically the same as the tsNET and tsNET* methods.

TABLE 5. Normalized stress metric (Equation 25). Each value represents the relative deviation between the GTD and the Euclidean distance.

	KK	FR	Lg	GRIP	PMDS	tsNET	tsNET*	tN-KK	tN-FR	tN-nL	tN-eL
n4c6-b1	0.251	0.274	0.337	0.305	0.329	0.291	0.285	**0.247**	0.271	0.274	0.631
grid1	2.384	2.547	2.593	2.601	2.333	0.151	0.156	**0.142**	0.169	0.183	0.263
dwt_307	**0.087**	0.095	0.151	0.093	0.098	0.165	0.224	0.098	0.104	0.198	0.213
problem1	2.332	2.806	2.709	2.809	2.262	0.034	0.034	**0.012**	0.043	0.037	0.128
dwt_419	**0.018**	0.034	0.283	0.033	0.028	0.037	0.037	0.019	0.184	0.094	0.138
n4c5-b10	0.274	0.255	0.281	0.279	0.319	0.192	0.195	0.197	**0.184**	0.192	0.477
G13	**0.119**	0.121	0.217	0.123	0.220	0.191	0.203	0.173	0.228	0.236	0.156
cis-n4c6-b15	0.266	0.260	0.265	0.291	0.321	0.160	0.178	0.159	**0.147**	0.159	0.647
L	0.350	0.359	0.425	0.442	0.402	0.102	0.105	**0.058**	0.089	0.106	0.079
price_1000	0.140	**0.129**	0.280	0.148	0.256	0.254	0.222	0.154	0.197	0.198	0.626
dwt_1005	0.022	0.098	0.589	0.082	**0.031**	0.305	0.065	**0.031**	0.126	0.190	0.828
tols1090	0.680	0.819	0.803	0.690	0.719	0.359	0.385	**0.326**	0.359	0.354	1.413
G34	0.127	**0.122**	0.264	0.142	0.271	0.251	0.195	0.201	0.486	0.587	0.563
sierpinski3d	0.157	**0.115**	0.354	0.207	**0.115**	0.124	0.128	0.131	0.117	0.118	0.121

TABLE 6. Neighborhood preservation metric (Equation 26), where $r_G = 2$. This table represents how well the neighbors of the GTD and the k-Nearest neighbors of the graph.

	KK	FR	Lg	GRIP	PMDS	tsNET	tsNET*	tN-KK	tN-FR	tN-nL	tN-eL
n4c6-b1	0.192	0.186	0.254	0.185	0.210	0.342	**0.346**	0.238	0.334	0.332	0.261
grid1	0.095	0.096	0.080	0.093	0.090	0.651	0.644	0.594	0.633	**0.662**	0.602
dwt_307	0.422	0.393	0.426	0.438	0.370	0.661	**0.688**	0.587	0.608	0.652	0.659
problem1	0.092	0.093	0.081	0.092	0.088	0.853	0.853	**0.913**	0.813	0.834	0.486
dwt_419	0.724	0.715	0.484	0.677	0.683	0.726	0.729	**0.756**	0.294	0.680	0.503
n4c5-b10	0.042	0.065	0.067	0.043	0.039	**0.318**	0.314	0.307	0.294	0.297	0.232
G13	0.270	0.254	0.232	0.236	0.197	**0.563**	0.562	0.324	0.520	0.529	0.289
cis-n4c6-b15	0.056	0.101	0.102	0.067	0.078	**0.813**	0.811	0.685	0.740	0.777	0.640
L	0.132	0.144	0.116	0.110	0.105	**0.705**	0.703	0.595	0.600	0.629	0.472
price_1000	0.203	0.497	0.530	0.205	0.345	0.597	0.596	0.534	0.628	**0.629**	0.394
dwt_1005	0.554	0.464	0.362	0.456	0.459	0.621	**0.642**	0.566	0.579	0.594	0.270
tols1090	0.006	0.010	0.011	0.011	0.011	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.618
G34	0.260	0.237	0.191	0.182	0.200	0.543	**0.577**	0.280	0.530	0.576	0.461
sierpinski3d	0.496	0.504	0.381	0.252	0.285	0.558	0.558	0.551	0.585	**0.587**	0.261

E. RUNNING TIME

In Table 7, we calculated the running time of the t-SNE series methods. Since our layout method uses random initialization, the number of iterations varies greatly, and the running time also differs greatly. We take an average of three runtime times. We use laptops to run programs. The hardware configuration is CPU Intel i7-7700HQ, GPU GeForce GTX 1050Ti, RAM 16GiB. Since the t-NeRV series graph layout methods usually take many iterations, it is not suitable for the layout of the large graphs. But it makes sense to use the accelerated t-NeRV method to speed up the layout. Compared with the tsNET and tsNET* methods, the running time of our method is not significantly reduced, and even some data sets (price_1000,tols1050, G34) have a large increase. This is because, in the third stage of the graph layout, the energy terms always change the layout of the second stage, resulting in an increase in the number of iterations, but not all datasets are increased (grid1, G13,dwt_1005). Running time is acceptable when laying out some small and medium graphs.

TABLE 7. Running time.

	tsNET	tsNET*	tN-KK	tN-FR	tN-nL	tN-eL
n4c6-b1	27.12	19.39	**15.78**	60.44	62.17	34.21
grid1	**12.15**	12.58	13.00	37.94	64.03	33.06
dwt_307	**26.02**	43.17	33.15	176.52	71.25	68.33
problem1	41.31	41.11	**27.19**	150.52	160.21	121.98
dwt_419	65.64	41.29	**32.58**	155.09	204.81	126.09
n4c5-b10	329.01	**154.30**	168.41	547.99	633.52	447.64
G13	1010.85	499.60	**436.79**	978.64	1046.32	930.27
cis-n4c6-b15	**810.62**	1714.06	891.70	1350.45	886.71	845.17
L	718.91	**598.31**	954.69	1392.12	1436.01	1569.48
price_1000	739.63	**732.91**	1108.03	2031.88	1939.76	1237.98
dwt_1005	1246.80	516.79	**480.12**	1547.50	1564.31	1349.78
tols1090	1092.63	**895.24**	1516.94	2198.81	2712.61	1610.72
G34	7176.07	**6690.56**	14317.03	18938.80	5220.16	12505.12
sierpinski3d	10780.90	**5692.10**	6037.15	18072.40	18850.49	9933.12

V. DISCUSSION

In Equation 9, the layout result is very stable for the change of a. As the value of a increases, neighboring nodes are crowded together, rejecting non-adjacent nodes. As a decreases, the influence of GTD on Euclidean distance will become smaller, causing more crossovers after graph layout.

The energy model here is different from the energy-based graph layout algorithms in the graph layout, such as KK. The energy models here uses a definition similar to Energy-Based Models (EBMs) [38], where EBMs is defined as models that can capture dependencies between variables by associating a scalar energy to each configuration of the variables, and any method that can be represented as this form can be called an energy model, e.g., modified FR in Section III-D.

Section III-D illustrates how to combine t-NeRV and energy models with three examples, but our method is not limited to these three methods. In fact, the energy model is more widely defined. For example, the PMDS method can also be thought of as an energy model, and the tsNET* method is a combination of the modified t-SNE and the PMDS energy model. Our method provides a graph layout framework based on energy models, and here we just list the combination of t-NeRV and three energy models as examples.

VI. CONCLUSION

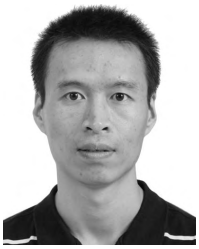
In this paper, we present a graph layout algorithm framework that combines energy models and t-NeRV, which provides a research direction for the layout of graph. Our method can maintain the global and local structure of the graph and can obtain a map layout that is more conform to aesthetic standards. We use the t-NeRV method for layout so that we can better get the global structure of the graph. We use the modified GTD matrix instead of using the GTD matrix directly, which gives better layout results. The layout of the t-NeRV series is very slow and not suitable for layout of large graphs. We can use the accelerated t-NeRV algorithm to speed up the layout, making this series of graph layout algorithms more practical.

In the future, we will mainly have two directions in our research: (1) Continue to study the layout method based on energy models (2) Learn how to accelerate our proposed graph layout algorithm to make our algorithm suitable for large graph.

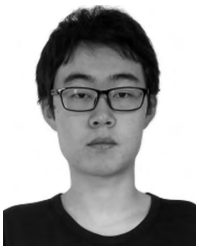
REFERENCES

[1] B. Pajntar, "Overview of algorithms for graph drawing," *Knowl. Creation, Diffusion, Utilization*, vol. 3, no. 6, pp. 1–4, 2006.
[2] A. Frick, A. Ludwig, and H. Mehldau, "A fast adaptive layout algorithm for undirected graphs (extended abstract and system demonstration)," in *Proc. Int. Symp. Graph Drawing*, 1994, pp. 388–403.
[3] D. Harel and Y. Koren, "A fast multi-scale method for drawing large graphs," in *Proc. Int. Symp. Graph Drawing*, 2000, pp. 183–196.
[4] U. Brandes and C. Pich, "Eigensolver methods for progressive multidimensional scaling of large data," in *Proc. Int. Symp. Graph Drawing*, 2006, pp. 42–53.
[5] D. Harel and Y. Koren, "Graph drawing by high-dimensional embedding," in *Proc. Int. Symp. Graph Drawing*, 2002, pp. 207–219.
[6] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, 1989.
[7] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Softw., Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, 1991.
[8] A. Noack, "An energy model for visual graph clustering," in *Proc. Int. Symp. Graph Drawing*, 2003, pp. 425–436.
[9] H. Gibson, J. Faith, and P. Vickers, "A survey of two-dimensional graph layout techniques for information visualisation," *Inf. Vis.*, vol. 12, nos. 3–4, pp. 324–357, 2013.
[10] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization," *J. Mach. Learn. Res.*, vol. 11, pp. 451–490, Feb. 2010.
[11] W. T. Tutte, "Convex representations of graphs," *Proc. London Math. Soc.*, vol. 3, no. 1, pp. 304–320, 1960.
[12] W. T. Tutte, "How to draw a graph," *Proc. London Math. Soc.*, vol. 3, no. 1, pp. 743–767, 1963.
[13] P. Eades, "A heuristic for graph drawing," *Congressus Numerantium*, vol. 42, pp. 149–160, 1984.
[14] P. Gajer *et al.*, "A fast multi-dimensional algorithm for drawing large graphs," in *Proc. Graph Drawing Conf.*, 2000, pp. 211–221.
[15] P. Gajer and S. G. Kobourov, "GRIP: Graph dRawing with intelligent placement," in *Proc. Int. Symp. Graph Drawing*, 2000, pp. 222–228.
[16] S. Hachul, and M. Jünger, "Drawing large graphs with a potential-field-based multilevel algorithm," in *Proc. Int. Symp. Graph Drawing*, 2004, pp. 285–295.
[17] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.
[18] H. A. Simon, "The architecture of complexity," in *Facets of Systems Science*. Boston, MA, USA: Springer, 1991, pp. 457–476.
[19] J. Blythe, C. McGrath, and D. Krackhardt, "The effect of graph layout on inference from social network data," in *Proc. Int. Symp. Graph Drawing*, 1995, pp. 40–51.
[20] E. Dengler and W. Cowan, "Human perception of laid-out graphs," in *Proc. Int. Symp. Graph Drawing*, 1998, pp. 441–443.
[21] W. S. Torgerson, "Multidimensional scaling: I. Theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, Dec. 1952.
[22] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[23] E. Bonabeau, and F. Hénaux, "Self-organizing maps for drawing large graphs," *Inf. Process. Lett.*, vol. 67, no. 4, pp. 177–184, 1998.
[24] E. Bonabeau, "Graph multidimensional scaling with self-organizing maps," *Inf. Sci.*, vol. 143, nos. 1–4, pp. 159–180, 2002.
[25] B. Meyer, "Competitive learning of network diagram layout," in *Proc. IEEE Symp. Vis. Lang.*, Sep. 1998, pp. 56–63.
[26] Z. Yang, J. Peltonen, and S. Kaski, "Optimization equivalence of divergences improves neighbor embedding,"in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 460–468.
[27] J. F. Kruiger, P. E. Rauber, R. M. Martins, A. Kerren, S. Kobourov, and A. C. Telea, "Graph Layouts by t-SNE," *Comput. Graph. Forum*, vol. 36, no. 3, pp. 283–294, 2017.
[28] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[29] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 857–864.
[30] I. S. Duff, R. G. Grimes, and J. G. Lewis, "Sparse matrix test problems," *ACM Trans. Math. Softw.*, vol. 15, no. 1, pp. 1–14, 1989.
[31] T. A. Davis and Y. Hu, "The University of Florida sparse matrix collection," *ACM Trans. Math. Softw.*, vol. 38, no. 1, 2011, Art. no. 1.
[32] R. Ryan and A. Nesreen, "The network data repository with interactive graph analytics and visualization," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 4292–4293.
[33] T. P. Peixoto, "The graph-tool python library," *Figshare*, 2014. [Online]. Available: https://figshare.com/articles/graph_tool/1164194
[34] J. Ellson. (2008). *Graphviz-Graph Visualization Software*. [Online]. Available: http://www.graphviz.org/
[35] D. Auber, "Tulip—A huge graph visualization framework," in *Proc. Graph Drawing Softw.*, 2004, pp. 105–126.
[36] M. Ortmann, M. Klimenta, and U. Brandes, "A sparse stress model," in *Proc. Int. Symp. Graph Drawing Netw. Vis.*, 2016, pp. 18–32.
[37] E. R. Gansner, Y. Hu, and S. North, "A maxent-stress model for graph layout," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 6, pp. 927–940, Jun. 2013.
[38] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang "A tutorial on energy-based learning," *Predicting Struct. Data*, vol. 1, pp. 1–59, Aug. 2006.

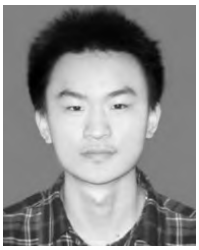


GUANGLUAN XU received the B.Sc. degree from Beijing Information Science and Technology University, Beijing, China, in 2000, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2005. He is currently a Professor with the Institute of Electronics, Chinese Academy of Sciences. His research interests include computer vision and remote sensing image understanding.



ZHENZHOU SONG received the bachelor's degree from Jilin University, China, in 2017. He is currently pursuing the master's degree with the Institute of Electronics, University of Chinese Academy of Sciences, with a focus on dimension reduction and graph layout.

YANG WANG received the B.E. degree from Beihang University, Beijing, China, and the Ph.D. degree from Peking University, Beijing. She is currently an Assistant Professor with the Institute of Electronics, Chinese Academy of Sciences. Her research interests include geospatial data organization and visualization.



DAOYU LIN received the bachelor's degree from the Beijing University of Posts and Telecommunications. He is currently pursuing the degree with the Institute of Electronics, University of Chinese Academy of Sciences, with a focus on computer vision and image processing.

JING CHEN received the bachelor's degree from the Beijing Institute of Technology, China, in 2016. She is currently pursuing the master's degree with the Institute of Electronics, University of Chinese Academy of Sciences, with a focus on visualization analysis and network data mining.

TINGYUN MAO received the bachelor's degree from Jilin University, China, in 2016. He is currently pursuing the Ph.D. degree with the Institution of Electronics, Chinese Academy of Sciences. His research interests include information visualization and visual analytics.

WENJIA XU received the bachelor's degree (Hons.) from the Beijing Institute of Technology. She is currently pursuing the Ph.D. degree with the Institute of Electronics, University of Chinese Academy of Sciences, with a focus on machine learning and computer vision.