Visual Informatics 6 (2022) 50–61

Contents lists available at ScienceDirect

Visual Informatics

journal homepage: www.elsevier.com/locate/visinf

A machine learning approach for predicting human shortest path task performance

Shijun Cai [a,*], Seok-Hee Hong [a], Xiaobo Xia [a], Tongliang Liu [a], Weidong Huang [b]

[a] University of Sydney, Australia
[b] University of Technology Sydney, Australia

ARTICLE INFO

Article history:
Available online 6 April 2022

Keywords:
Graph drawing
Machine learning
Shortest path task
Quality metrics

ABSTRACT

Finding a shortest path for a given pair of vertices in a graph drawing is one of the fundamental tasks for qualitative evaluation of graph drawings. In this paper, we present the first machine learning approach to predict human shortest path task performance, including accuracy, response time, and mental effort.

To predict the shortest path task performance, we utilize correlated quality metrics and the ground truth data from the shortest path experiments. Specifically, we introduce *path faithfulness metrics* and show strong correlations with the shortest path task performance. Moreover, to mitigate the problem of insufficient ground truth training data, we use the transfer learning method to pre-train our deep model, exploiting the correlated quality metrics.

Experimental results using the ground truth human shortest path experiment data show that our models can successfully predict the shortest path task performance. In particular, model MSP achieves an MSE (i.e., test mean square error) of 0.7243 (i.e., data range from −17.27 to 1.81) for prediction.

1. Introduction

Evaluation of graph drawings has been established as an important research area in graph drawing. Quality metrics (or aesthetic criteria), such as edge crossings, bends, area, total edge lengths, angular resolution and stress, have been proposed for *quantitative evaluation* of graph drawings (Di Battista et al., 1999). Subsequently, various graph drawing algorithms have been developed to optimize these metrics.

Moreover, *qualitative evaluation* on graph drawings is well studied using HCI (Human Computer Interaction) evaluation methods such as controlled human experiments. In particular, finding a *shortest path* for a given pair of vertices in a graph drawing is one of the fundamental tasks for qualitative evaluation (Huang et al., 2008; Purchase, 1997; Ware et al., 2002). Namely, a drawing D_1 of a graph is better than a drawing D_2, if human spend less time finding the shortest path with fewer errors.

A number of studies have established the correlation between quality metrics, such as edge crossings and crossing angles, and the shortest path task performance (i.e., time and accuracy). Specifically, Huang et al. (2016) defined the *performance-based efficiency E* of the shortest path task performance based on the accuracy, response time, and mental effort.

Recently, machine learning approaches have been used to address research problems in graph visualization, mainly focusing on *quantitative evaluation* (i.e., *quality metrics*) (Haleem et al., 2019; Kwon and Ma, 2019). More recently, a machine learning approach has been proposed to address *qualitative evaluation*, specifically predicting *human preference* task performance (Cai et al., 2021).

In this paper, we present the first machine learning approach to predict the human shortest path task performance. Specifically, we propose three machine learning models using correlated quality metrics and graph drawing images with the highlighted shortest path from the ground truth human experiment data (Huang et al., 2016; Fletcher et al., 2019; Huang et al., 2014). The main contributions of this paper are summarized as follows:

1. We introduce new *path faithfulness metrics* and show that they are strongly correlated with the efficiency E of the shortest path task, using the ground truth data from shortest path experiments (Huang et al., 2016; Fletcher et al., 2019; Huang et al., 2014) (See Section 3).
Moreover, we perform correlation analysis and the feature importance test using a variety of quality metrics and graph

[*] Corresponding author.
E-mail addresses: scai5619@uni.sydney.edu.au (S. Cai), seokhee.hong@sydney.edu.au (S.-H. Hong), xxia5420@uni.sydney.edu.au (X. Xia), tongliang.liu@sydney.edu.au (T. Liu), weidong.huang@uts.edu.au (W. Huang).

https://doi.org/10.1016/j.visinf.2022.04.001

properties, to find the most correlated metrics or properties to the efficiency E.

2. We present the first machine learning approach for predicting shortest path task performance, with the following three models (see Section 4):

    - Model M (Metrics) is based on regression and classification models, and trained using the most correlated quality metrics and the *task performance labels* (see Section 4.1).
    - Model SP (Shortest path) is a CNN-based (convolutional neural network) deep model, which reads graph drawing images with the highlighted shortest path from the ground truth shortest path experiments (Huang et al., 2016; Fletcher et al., 2019; Huang et al., 2014) and converts them into feature vectors. The deep model is trained to mimic the ground truth efficiency for the shortest path task by fitting the training data, consisting of graph drawing images and task performance labels.
    - Model MSP (M+SP) employs the *transfer learning* method to mitigate insufficient ground truth training data from the shortest path experiments. Specifically, we first pre-train the model using graph drawing images with highlighted paths and the *metric-based label* (i.e., the most correlated quality metrics or properties, see Section 4.4). Then, we fine-tune the model using graph drawing images with highlighted paths and task performance labels.

3. Experiments using the ground truth shortest path experiment data (Huang et al., 2016; Fletcher et al., 2019; Huang et al., 2014) show that all three models successfully predict the shortest path task performance. Overall, MSP performs the best, demonstrating the importance of employing path quality metrics for transfer learning, achieving an MSE of 0.7243 (i.e., data range from −17.27 to 1.81) for predicting efficiency E (see Section 5).

The rest of this paper is organized as follows. Section 2 describes the background, and Section 3 presents the new path faithfulness metrics and correlation analysis. Section 4 presents our machine learning models in detail, and Section 5 describes experimental results and discussions. Section 6 concludes with future work.

2. Background

2.1. Quality metrics for graph drawing

Readability Metrics. Various quality metrics, called *aesthetic criteria*, are available for quantitative evaluation of graph drawings (Di Battista et al., 1999). Consequently, many graph drawing algorithms have been designed to optimize these quality metrics (Di Battista et al., 1999).

Traditional *readability* metrics, such as edge crossings, bends, area, total edge lengths, angular resolution, crossing angles and overlap between the vertices and edges, measure how human understand graph drawings. However, most readability metrics tend to focus on *small* graphs.

Faithfulness Metrics. Recently, *faithfulness* metrics have been developed for the evaluation of *large* graph drawings, which measure how faithfully drawings show the ground truth structure of graphs. For example, *stress* (Di Battista et al., 1999) is a *distance faithful* metric, which compares the difference between the graph-theoretic distance of vertices and the Euclidean distance of vertices in a drawing. The *shape-based metrics* compare the similarity between a graph and a proximity graph, such as the Gabriel graph and the Relative Neighborhood graph, computed from a drawing (Eades et al., 2015).

The *cluster faithfulness* metrics (Meidiana et al., 2019) compare the similarity between the ground truth clustering of a graph and the geometric clustering computed from a drawing. The *symmetry faithfulness* metrics (Meidiana et al., 2020b) measure how the ground truth *automorphisms* of a graph are displayed as symmetries in a drawing. The *change faithfulness* metrics (Meidiana et al., 2020a) measure how the ground truth changes in dynamic graphs are proportionally displayed as a geometric change in drawings.

2.2. Qualitative evaluation using shortest path experiments

Qualitative evaluation of graph drawings has been investigated by conducting HCI experiments with task performance, measuring time and accuracy. Finding the shortest path for a given pair of vertices in a graph drawing is one of the most popular tasks for qualitative evaluation of graph drawing.

For example, Ware et al. (2002) found that fewer edge crossings and *path continuity* (i.e., less path bendiness) are significantly correlated with the shortest path task performance. Moreover, Huang et al. (2008, 2009) found a correlation between large *crossing angles* and the shortest path task performance, as well as the *geodesic-path tendency* (i.e., edges toward the target vertex is more likely to be searched first) in finding a shortest path.

Recently, a series of shortest path experiments have been conducted (Huang et al., 2016; Fletcher et al., 2019; Huang et al., 2014) using the system shown in Fig. 1. Each experiment recruited participants from different organizations, and participants could practice the system before the experiment started. The experiments used the Rome graphs[1] (i.e., small and sparse graphs) drawn with a force-directed algorithm in Batagelj and Mrvar (2004).

Each trial of the experiment began with the first screen (see Fig. 1(1)), showing two highlighted vertices, which were randomly selected with the following conditions: the shortest path was unique and the path length was between 3 and 6.

The second screen (see Fig. 1(2)) presented a node-link diagram, and participants were instructed to find the shortest path between the two highlighted vertices, as quickly and accurately as possible. The time to complete the task was recorded as the *response time*. The third screen (see Fig. 1(3)) asked participants to answer the length of the found path, and then rate the *mental effort* from 1 to 9 on a Likert scale.

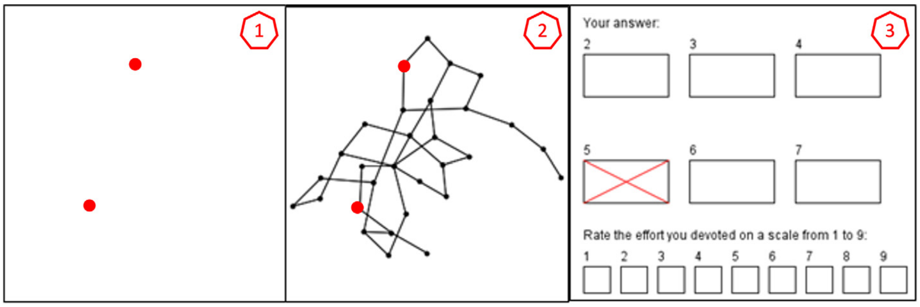
The *accuracy* (True (1) or False (0)) was computed based on the ground truth shortest path length, i.e., the accuracy is True if a participant correctly answers the length of the shortest path. Huang et al. (2016) defined the *performance-based efficiency E* for shortest path task performance as follows:

$$E = \frac{Z_{accuracy} - Z_{mental\ effort} - Z_{response\ time}}{\sqrt{3}}$$

Roughly speaking, the efficiency E is defined as the difference between the cognitive gain (i.e., accuracy) and the cognitive cost (i.e., mental effort and response time). Specifically, a drawing is of high efficiency, if high accuracy is achieved with low mental effort and less response time, and vice versa.

Note that the accuracy, mental effort and response time have been normalized into z scores to be on the same scale and become addable, e.g., $Z = \frac{\tau-\mu}{\sigma}$, where μ is the mean and σ is the standard deviation of data entries among all drawings and participants, and τ is the value of the accuracy (resp., mental effort or response time) of each data entry.

[1] http://www.graphdrawing.org/data.html.



Your answer:

2 □ 3 □ 4 □

5 ☒ 6 □ 7 □

Rate the effort you devoted on a scale from 1 to 9:

1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7 □ 8 □ 9 □

Fig. 1. Examples of three screens shown to participants for the shortest path task.

2.3. Machine learning approaches for graph visualization

Recently, machine learning approaches have been popular for addressing problems in visualization. For example, see a survey (Wang et al., 2020) on machine learning techniques to improve design, development, and evaluation of visualization.

Specifically, Kwon and Ma (2019) designed a GNN(graph neural network)-based encoder–decoder neural network to generate a new good layout from the pairwise distance matrix of vertex positions of a given layout. Giovannangeli et al. (2020) used deep convolutional networks to predict the length of the shortest path between two highlighted vertices in the images of node-link diagrams and adjacency matrices.

Machine Learning Approaches using Quality Metrics. A number of researchers employed machine learning approaches to solve problems in graph drawing, mainly focusing on *quantitative evaluation* (i.e., quality metrics) (Haleem et al., 2019; Kwon et al., 2017; Klammler et al., 2018). For example, Haleem et al. (2019) used a CNN with graph layout images to predict the readability metrics, such as vertex spread, minimum angle, edge length variation, group overlap and edge crossings.

Kwon et al. (2017) used a support vector regression model to estimate the quality metrics of a given drawing D_1 of a graph G_1 using a drawing D_2 of a graph G_2, where G_1 and G_2 have similar topological structures and D_1 and D_2 are computed by the same layout algorithm. Klammler et al. (2018) used the Siamese neural network with quality metrics to compare a graph drawing D with its deformed drawing D'.

Machine Learning Approaches for Qualitative Evaluation. Recently, a machine learning approach has been proposed to predict qualitative evaluation for graph drawing (Cai et al., 2021). Specifically, a CNN-Siamese-based model was presented to predict *human preference* between a pair of different layouts of the same graph. They employed a *transfer learning* method to overcome the insufficiency of ground truth human preference experiment data for training the deep model, i.e., pretraining the model using quality metrics correlated to human preference, and then fine-tuning the model using the ground truth human preference experiment data.

3. Faithfulness metrics and correlation analysis

Previous work (Ware et al., 2002; Huang et al., 2016; Fletcher et al., 2019; Huang et al., 2014, 2009) established the correlation between the *readability* metrics (see Table 1) and the *shortest path* task performance (i.e., efficiency, response time, accuracy and mental effort). More recently, researchers established the correlation between the *faithfulness* metrics and the *human preference* task performance (Eades et al., 2015; Chimani et al., 2014).

Motivated by these results, in this paper, we investigate the correlation between the faithfulness metrics and the shortest path task performance. More specifically, we introduce new *path faithfulness metrics* for measuring the quality of a drawing of the shortest path.

3.1. Path faithfulness metrics

Path faithfulness metrics are defined for a shortest path P between a pair of vertices and a drawing D_P of P.

Path Shape-based Metrics. We define the path shape-based metrics using the mean Jaccard Similarity between a path $P = (V, E)$ and a proximity graph $P' = (V, E')$, computed from a drawing D_P, as follows:

$$MJS(P, P') = \frac{1}{|V|} \sum_{v \in V} \frac{|N(v) \cap N'(v)|}{|N(v) \cup N'(v)|}$$

where $N(v)$ is the set of neighbors of v in P.

Specifically, we present several variations of path shape-based metrics, *pShape_GG*, *pShape_RNG*, *pShape_EMST*, and *pShape_KNN*, based on the types of proximity graphs (Toussaint, 2014) defined as follows. For a given point set Q in the plane,

- The Gabriel graph (GG) has an edge between two points $p, q \in Q$ if the closed disk which has the line segment pq as a diameter contains no other elements of Q.
- The relative neighborhood graph (RNG) has an edge between two points $p, q \in Q$ if there is no point $r \in Q$ such that $d(p, r) \leq d(p, q)$ and $d(q, r) \leq d(p, q)$.
- A Euclidean minimum spanning tree (EMST) is a minimum spanning tree of Q where the weight of the edge is the Euclidean distance.
- The k-nearest neighbor graph (KNN) has a (directed) edge from $p \in Q$ to $q \in Q$ if the number of points $r \in Q$ with $d(p, r) < d(p, q)$ is at most $k - 1$.

Path Stress Metrics. We define the path stress metrics based on the difference between the graph-theoretic distance and the Euclidean distance of two vertices i, j of a path in the drawing. Specifically, we define three variations based on the scaling as follows:

$$pRegularStress(D_P) = \sum_{i,j \in P} w_{ij}(\|x_i - x_j\|/d_{ij} - 1)^2$$

$$pAvgScaledStress(D_P) = \sum_{i,j \in P} w_{ij}(\|x_i - x_j\|/(d_{ij} \cdot l_{avg}))^2$$

$$pAvgStress(D_P) = \sum_{i,j \in P} w_{ij}(\|x_i - x_j\|/d_{ij} - l_{avg})^2$$

where

- x_i is the position of a vertex i of P in D_P,
- $\|x_i - x_j\|$ is the Euclidean distance between x_i and x_j in D_P,
- d_{ij} is the graph-theoretic distance between i and j in P,
- $w_{ij} = d_{ij}^{-2}$ is the weight factor,
- l_{avg} is the average edge length of D_P.

Table 1 shows the complete list of quality metrics (including readability and faithfulness metrics) as well as properties of a graph G and a path P.

Table 1
List of quality metrics and properties: readability metrics (Ware et al., 2002; Huang et al., 2016; Fletcher et al., 2019; Huang et al., 2014, 2009), and faithfulness metrics for graph and path.

	Graph
Property	$\lvert V\rvert$ (number of vertices) $\lvert E\rvert$ (number of edges)
Readability metrics	crossing (number of edge crossings) angleM (mean of all crossing angles) angleD (standard deviation of all crossing angles) minAng (minimum crossing angle) edgeM (mean of all edge lengths) edgeD (standard deviation of all edge lengths) minVertx (minimum vertex angular resolution) finVertx (mean of the differences between 2π/degree and minimum angle for all vertices) minPtEdg (minimum distance from a vertex to an edge)
Faithfulness metrics	Shape_GG (shape-based metric using Gabriel graph) Shape_RNG (shape-based metric using Relative Neighborhood graph) Shape_EMST (shape-based metric using Euclidean Minimum Spanning Tree) Shape_KNN (shape-based metric using k-Nearest Neighbor graph) RegularStress (regular stress) AvgScaledStress (average scaled stress) AvgStress (average stress)
	Path
Property	PathLength (number of edges of the path)
Readability metrics	pCrossNo (number of edge crossings of the path) pAngMean (mean of all crossing angles of the path) pMinAng (minimum crossing angles of the path) pContinu (path continuity: sum of angular deviations at all vertices of the path) pGeode (geodesic path continuity)
New faithfulness metrics	**pShape_RNG** (path shape-based metric using Relative Neighborhood graph) **pShape_EMST** (path shape-based metric using Euclidean Minimum Spanning Tree) **pShape_KNN** (path shape-based metric using k-Nearest Neighbor) **pRegularStress** (regular stress of the path) **pAvgScaledStress** (average scaled stress of the path) **pAvgStress** (average stress of the path)

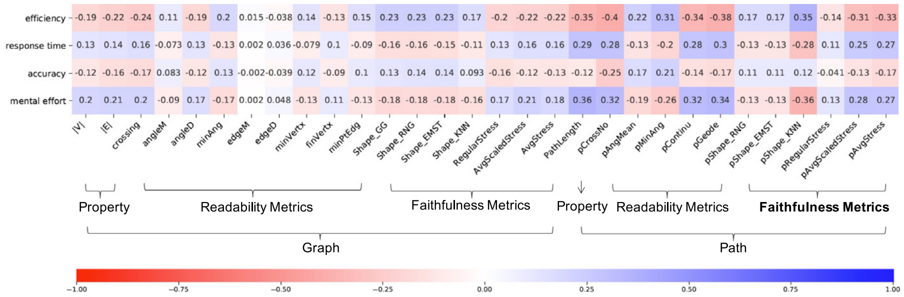


Fig. 2. Pearson correlation test between quality metrics, graph (resp., path) properties, and task performance (i.e., efficiency, response time, accuracy and mental effort). Red (resp., blue) color represents negative (resp., positive) correlation, and darker color represents a stronger correlation.

3.2. Correlation and feature importance

To find the most correlated metrics and properties for predicting shortest path task performance, we employ the Pearson correlation test (see Fig. 2) and the feature importance test (see Table 2).

Correlation between Faithfulness Metrics and Efficiency. The Pearson correlation test (Benesty et al., 2009) is executed by the default setting of DataFrame.corr function and the feature importance test is done by sklearn.feature_selection.SelectKBest function. Specifically, we use $F - value$ for regression (i.e., efficiency and response time), and Chi^2 (i.e., Chi-squared statistics) for classification (i.e., accuracy and mental effort).

Fig. 2 clearly shows that faithfulness metrics are correlated with efficiency E. Specifically, efficiency E is positively correlated with the shape-based metrics and negatively correlated with stress. Among the variations in the shape-based (resp., stress) metrics, $pShape\_KNN$ (resp., $pAvgStress$) shows the strongest correlation.

Path Quality Metrics vs. Graph Quality Metrics. Fig. 2 shows that the *path* quality metrics have a stronger correlation than the *graph* quality metrics. For example, $pShape\_KNN = 0.35$ (resp., $pCrossNo = -0.4$) shows a much stronger positive (resp., negative) correlation than corresponding $Shape\_KNN = 0.17$ (resp., $crossing = -0.24$).

Readability Metrics vs. Faithfulness Metrics. For *path* quality metrics, both readability metrics ($pCrossNo = -0.4$, $pGeode = -0.38$, $pContinu = -0.34$) and faithfulness metrics ($pShape\_KNN = 0.35$, $pAvgStress = -0.33$, $pAvgScaledStress = -0.31$) show correlations.

Table 2
Feature importance test among quality metrics, graph (resp., path) properties in descending order of the efficiency. A larger $F - value$ (resp., Chi^2) represents a more important metric or property.

	Efficiency ($F - value$)	Response time ($F - value$)	Accuracy (Chi^2)	Mental effort (Chi^2)
pCrossNo	**1823.52**	1105.77	95.03	160.04
pGeode	**1665.46**	1215.85	30.26	131.78
pShape_KNN	**1491.28**	1112.49	31.19	299.45
PathLength	**1478.81**	1140.70	54.82	535.77
pContinu	**1413.28**	1036.81	18.86	94.16
pAvgStress	**1259.86**	950.80	42.14	102.60
pAvgScaledStress	1072.76	846.31	23.04	101.16
pMinAng	**994.70**	536.09	68.64	101.95
Crossing	626.63	337.92	55.87	82.18
Shape_GG	567.24	348.87	16.68	34.91
Shape_EMST	557.28	310.74	23.05	39.94
Shape_RNG	549.94	324.72	19.81	37.31
AvgStress	513.93	347.01	24.38	48.69
AvgScaledStress	491.99	344.65	16.40	51.57
pAngMean	490.34	230.94	13.44	19.47
$\|E\|$	480.78	237.26	19.10	34.52
RegularStress	417.49	216.38	28.42	36.27
minAng	385.82	222.08	23.98	41.47
angleD	373.82	203.68	11.18	23.29
$\|V\|$	367.00	206.50	12.62	36.08
pShape_RNG	321.00	212.04	23.32	36.83
pShape_EMST	321.00	212.04	23.32	36.83
Shape_KNN	274.13	155.07	16.46	57.92
minPtEdg	240.45	103.27	23.29	70.81
pRegularStress	197.92	163.87	2.47	27.01
minVertx	194.74	79.92	17.05	29.30
finVertx	190.35	129.48	2.87	4.49
angleM	114.78	67.11	0.67	0.95
edgeD	13.58	16.12	2.93	10.05
edgeM	2.59	0.07	0.01	10.18

For *graph* quality metrics, the faithfulness metrics (e.g., $Shape\_GG = 0.23$ and $AvgStress = -0.22$) show stronger correlations than the readability metrics (e.g., $angleM = 0.11$ and $minVertx = 0.14$), except $crossing = -0.24$ and $minAng = 0.2$.

Most Correlated Metrics and Properties.

Table 2 shows the feature importance test results. Based on the results, we select the following 7 most correlated metrics and properties with strong correlation (i.e., $Pearson\ coefficient > 0.3$ and $F - value > 900$), including *pCrossNo*, *pGeode*, *pShape_KNN*, *PathLength*, *pContinu*, *pAvgStress* and *pMinAng*, for our machine learning models M and MSP for predicting shortest path task performance in Section 4.

Note that we choose *pMinAng* instead of *pAvgScaledStress*, although the $F - value$ of *pAvgScaledStress* is larger than *pMinAng*, since *pAvgStress* shows a stronger correlation than *pAvgScaledStress*.

4. Machine learning models

This section describes our machine learning approach for predicting the efficiency, response time, accuracy and mental effort of human finding the shortest path in a graph drawing.

4.1. Shortest path task performance labels

Let $D_k(s, t)$ denote a drawing of a graph G_k with two prespecified end-vertices s and t. For each instance $D_k(s, t)$, we compute *task performance labels*, $L_{efficiency}$, L_{time}, $L_{accuracy}$ and L_{effort}, using the ground truth data from the shortest path experiments (Huang et al., 2016; Fletcher et al., 2019; Huang et al., 2014). Since human task performance can be subjective, different participants may have different efficiencies (resp., response time, accuracy and mental effort) for the same instance. To solve this conflict, we use the following method to reach a consensus.

For an instance $D_k(s, t)$, let $T_{efficiency}$ (resp., T_{time}, $T_{accuracy}$ and T_{effort}) denote the z scores of efficiency (resp., response time), accuracy and mental effort from the ground truth shortest path experiment data. Specifically, we compute each task performance label $L_{efficiency}$, L_{time}, $L_{accuracy}$ and L_{effort} for $D_k(s, t)$ using the average value of $T_{efficiency}$, T_{time}, $T_{accuracy}$ and T_{effort} of each participant as follows:

1. For each instance $D_k(s, t)$, let l be the number of occurrences in the ground truth shortest path experiment data.
2. For each instance $D_k(s, t)$, compute task performance labels as follows:
   - $L_{efficiency} = \sum_{i=1}^{l} T_{efficiency}/l$
   - $L_{time} = \sum_{i=1}^{l} T_{time}/l$
   - $L_{accuracy} = \lfloor \sum_{i=1}^{l} T_{accuracy}/l \rceil$
   - $L_{effort} = \lfloor \sum_{i=1}^{l} T_{effort}/l - 1 \rceil$.

4.2. Model M using quality metrics

Fig. 3 shows the pipeline of the machine learning model M to predict task performance labels, including (a) Model input: the most correlated path metrics or properties (i.e., *pShape_KNN*, *pAvgStress*, *pCrossNo*, *pMinAng*, *pContinu*, *pGeode* and *PathLength*) from Section 3.2; (b) Selected regression and classification models; and (c) Output prediction. To improve the learning performance, each metric is scaled to the range [0, 1], using preprocessing.MinMaxScaler function in sklearn library.

(a) Model Input. In the training phase, the input includes the most correlated path metrics or properties (i.e., *pShape_KNN*, *pAvgStress*, *pCrossNo*, *pMinAng*, *pContinu*, *pGeode* and *PathLength*), and task performance labels of the training data. In the testing phase, input includes the most correlated path metrics or properties of the test data to predict the task performance labels.

(b) Model Selection for Regression and Classification. We use regression models (resp., classification models) for $L_{efficiency}$ and

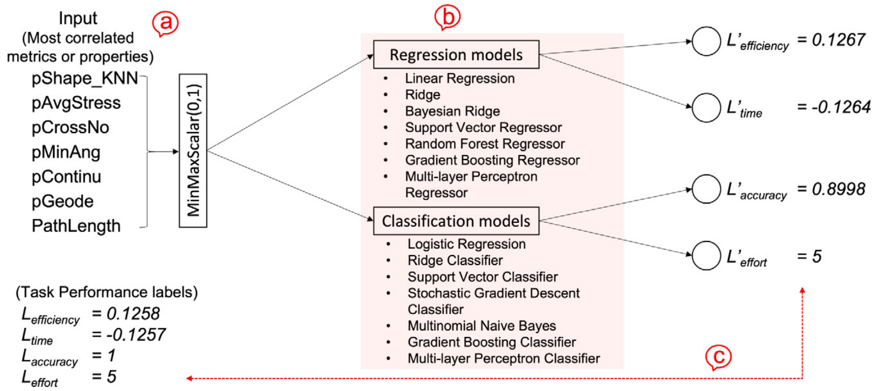


Fig. 3. Model M: (a) model input, (b) selected regression and classification models, (c) Output prediction.

L_{time} (resp., $L_{accuracy}$ and L_{effort}) from sklearn library (Pedregosa et al., 2011). Specifically, the selected regression (resp., classification) models are shown in Fig. 3(b).

Regression models include linear regression, ridge, Bayesian ridge, support vector regression, random forest regressor, gradient boosting regressor, and multi-layer perceptron regressor. Classification models include logistic regression, ridge classifier, support vector classifier, stochastic gradient descent classifier, multinomial naive Bayes, gradient boosting classifier, and multi-layer perceptron classifier.

Note that such models are selected based on smaller validation mean square errors (MSE) for regression and larger validation accuracy (ACC) for classification among machine learning models in the sklearn library (Pedregosa et al., 2011). More specifically, MSE means the mean squared error regression loss between the ground truth target values and the estimated target values. ACC means the accuracy classification score between the ground truth labels and predicted labels.

(c) Output prediction. In the training phase, we need to measure and minimize the difference between the output prediction (e.g., $L'_{efficiency}$) and task performance labels (e.g., $L_{efficiency}$).

In the testing phase, we use the MSE and ACC to measure the difference between the output prediction and the task performance labels of the test data set, and eventually evaluate the model performance.

4.3. CNN-based model SP

We also present a CNN-based model SP that can predict task performance labels from graph drawing images. The notable advantage of CNNs is that they excel at extracting features from image inputs.

Fig. 4 shows the pipeline of model SP, including: (a) Model input, i.e., a graph drawing with a highlighted shortest path in red color and the task performance labels; (b) A CNN-based (i.e., ResNet-18 He et al., 2016) image feature extractor and fully connected layers that convert the model output to match task performance labels; and (c) Output prediction.

(a) Model Input. In the training phase, input includes graph drawing images of size 320 × 320 with highlighted shortest path, and task performance labels of the training data. In the testing phase, input includes graph drawing images with highlighted shortest path of the test data to predict the task performance labels.

(b) A CNN-based Image Feature Extractor. Our CNN-based image feature extractor is built on ResNet-18, an 18-layer-deep residual network, which shows the best performance among other deep models in our preliminary experiment.

The last *fully connected layer* converts the output of the semantic feature vectors and maps them on task performance labels. Fig. 4(b) shows the design of our CNN-based feature extractor.

(c) Output Prediction. In the training phase, we aim to train the proposed deep model to have outputs aligning with task performance labels. Specifically, we optimize the parameters of the deep model by minimizing the difference between the model output and task performance labels.

More specifically, the output feature size of the last *fully connected layer* is one (resp., two and nine) with a sigmoid function (resp., Softmax function) for predicting $L_{efficiency}$ and L_{time} (resp., $L_{accuracy}$ and L_{effort}). For $L_{accuracy}$ (resp., L_{effort}), we convert the 2-dimensional array (resp., 9-dimensional array) to a single value using the numpy.argmax function, which returns the indices of the maximum values of an array.

Fig. 4(c) shows an example of task performance labels (e.g., $L_{accuracy} = 1$) and the model outputs (i.e., $L'_{accuracy} = [0.0021, 0.8755]$ and the maximum value is 0.8755 at index 1, which matches $L_{accuracy} = 1$), similar to the prediction of mental effort.

4.4. Model MSP using transfer learning

Transfer Learning. To train a deep model to better understand human performing the shortest path task, we need a large amount of labeled data. However, running the human experiment is usually time-consuming and expensive, therefore we address this issue by employing the transfer learning method (Pan and Yang, 2009).

Transfer learning extracts knowledge from a source task (different but related task) to improve learning performance in a target task, where the source task and target task share some similar information. Typically, if the target task has limited training data, by employing transfer learning, we can use a source task that has sufficient training data.

Metric-based Label. To pre-train the deep model MSP, we define a *metric-based label* (see Fig. 5(A)), using the following seven metrics and properties (i.e., *pShape_KNN*, *pAvgStress*, *pCrossNo*, *pMinAng*, *pContinu*, *pGeode* and *PathLength*), which are most correlated to shortest path task performance as shown in Section 3.2. To improve the learning performance, each metric is scaled to the range [0, 1]. Then we use the target task data (e.g., *task performance labels*) on the training data set to fine-tune the deep model MSP.

Model MSP. Fig. 5 shows the pipeline of model MSP with two phases:

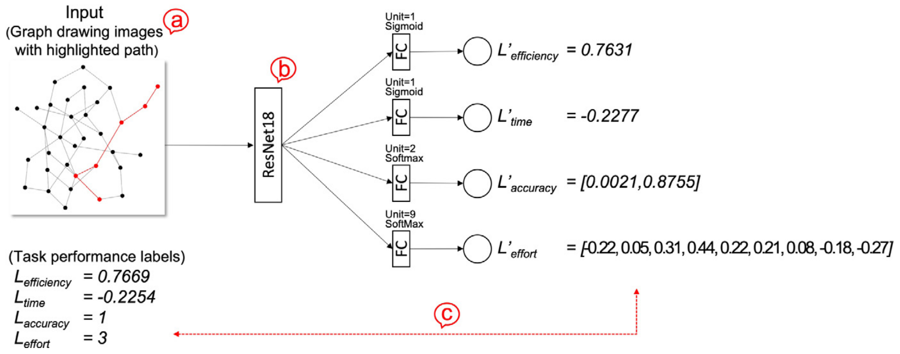


Fig. 4. Model SP: (a) Model input, (b) A CNN-based (i.e., ResNet-18) image feature extractor, (c) Output prediction.

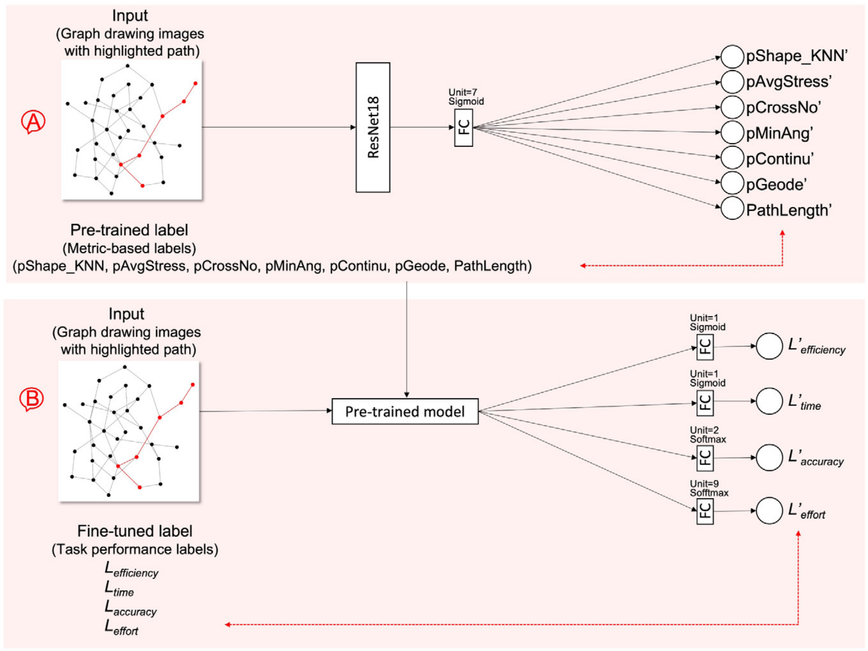


Fig. 5. Model MSP: (A) pre-training, (B) fine-tuning.

(A) Pre-training: In the pre-training phase, input includes graph drawing images with highlighted shortest path, and the metric-based label of the training data.
The model output is converted by a fully connected layer with a Sigmoid function. When the total validation loss of the metric-based label reaches a minimum value, we stop training and save the optimized model for fine-tuning.

(B) Fine-tuning: In the fine-tuning phase, input includes graph drawing images with highlighted shortest path, and the task performance labels of the training data, where we fine-tune the pre-trained model in phase (A) using the input.
In the testing phase, input includes graph drawing images with highlighted shortest path to predict the task performance labels using the fine-tuned deep model in phase (B).

5. Experiments

This section presents the details of our experiment, including data set, model design and implementation, model training, prediction results, and discussion.

5.1. Data set

We use the ground truth data from the shortest path experiments (Huang et al., 2016; Fletcher et al., 2019; Huang et al., 2014). The graphs are small and sparse Rome graphs (13–50 vertices and 12–71 edges) drawn by a force-directed layout. The length of the shortest path varies from 3 to 6.

Specifically, the data sets contain 230 graph drawings with various pre-specified vertices, resulting in 5542 instances of graph drawings with highlighted shortest paths. For each instance, we compute the task performance labels, as described in Section 4, where $L_{efficiency} \in [-17.27, 1.81]$, $L_{time} \in [-0.92, 31.34]$, $L_{accuracy}$ is 0 or 1, and L_{effort} is an integer from [0, 8].

5.2. Model design and implementation

To validate the importance of using path faithfulness metrics, we compare our models with a *baseline model B*, which uses the same selected regression and classification models shown in 3(b), however, trained with different inputs. Specifically, in the training phase, input includes all path *readability* metrics and task performance labels of the training data. In the testing phase, input includes all path readability metrics of the test data to predict the task performance labels.

Table 3
MSE and ACC of the prediction results of four trained models: All the models succeed in predicting task performance labels (i.e., efficiency, response time and accuracy). Note that MSP performs the best, M performs better than B, and SP performs better than M. A smaller MSE (resp., larger ACC) represents a better prediction result.

	B	*M*	*SP*	*MSP*
MSE: efficiency	0.7919 ± 0.0088	0.7740 ± 0.0078	0.7389 ± 0.0071	**0.7243 ± 0.0069**
MSE: response time	0.3757 ± 0.0065	0.3664 ± 0.0071	0.3627 ± 0.0059	**0.3555 ± 0.0104**
ACC: accuracy	(69.25 ± 0.83)%	(69.44 ± 0.91)%	(71.15 ± 0.76)%	**(71.69 ± 0.73)%**
ACC: mental effort	(30.40 ± 1.36)%	(32.17 ± 1.03)%	(32.69 ± 0.83)%	(33.37 ± 0.68)%

Therefore, we compare the following four models in our experiments:

1. *B*: a baseline regression (resp., classification) model trained on only path *readability* metrics (i.e., *pCrossNo*, *pAngMean*, *pMinAng*, *pContinu* and *pGeode*) and task performance labels.
2. *M*: a regression (resp., classification) model trained on the 7 most correlated path *readability* and *faithfulness* metrics (i.e., *pShape_KNN*, *pAvgStress*, *pCrossNo*, *pMinAng*, *pContinu*, *pGeode* and *PathLength*) and task performance labels.
3. *SP*: a deep model trained on graph drawing images with highlighted shortest path and task performance labels.
4. *MSP*: a deep model pre-trained on graph drawing images with highlighted shortest path and the *metric-based labels*, and then fine-tuned on graph drawing images with highlighted shortest path and task performance labels.

We implement the ResNet-18 for models SP and MSP by PyTorch on Google Colab Pro. The model parameters include an Adam optimizer, 5×10^{-4} weight decay, 128 batch size, 5×10^{-4} learning rate and 200 epochs.

5.3. Model training

For a machine learning algorithm to predict task performance labels, we need training data sets to train the model parameters. Furthermore, we need to select a model controlled by hyper-parameters with good performance.

To avoid overfitting in deep models, we exploit the data augmentation, including horizontal random flips and random rotations (i.e., 25 degrees). All experiments are repeated five times based on random data splitting, using random_state of model_selection.train_test_split function in sklearn library.

Specifically, we randomly split the input data into training and test data sets with a ratio of 7 : 3, therefore, training data are not used for the test data. For the training set, we further randomly selected 70% for training and 30% for validation. We repeat the random splitting five times to avoid overfitting.

More specifically, for model M (resp., B), we randomly split the most correlated path readability and faithfulness metrics (resp., path readability metrics) and their task performance labels for training and test data sets. Similarly, for model SP, we randomly split the graph drawings with highlighted shortest path and their task performance labels for training and test data sets. For model MSP, we randomly split graph drawings with highlighted shortest path and their metric-based (resp., task performance) labels for pre-training (resp., fine-tuning and test data set).

5.4. Prediction results

To compare the models, we use the MSE (test mean square error) and ACC (test accuracy) of their prediction results. Specifically, the MSE (resp., ACC) for models M and B is the minimum (resp., maximum) MSE (resp., ACC) of the prediction results among the seven regression (resp., classification) models described in Section 4.2. For models MSP and SP, the MSE (resp., ACC) is of the prediction results for the model. We compute the average MSE and ACC of prediction results from the five times of random splitting, with a standard deviation for four trained models.

Table 4
The *p*-values of the Wilcoxon signed-rank tests for comparing the pairwise difference between two models, showing that the comparison between the models is statistically significant (e.g., a *p*-value $<$ 0.05 means that the first model is statistically significantly better than the second model).

	M vs. B	*SP vs. M*	*MSP vs. M*	*MSP vs. SP*
Efficiency	0.0313	0.0313	**0.0313**	**0.0313**
Response time	0.0313	0.1563	**0.0313**	0.0938
Accuracy	0.0721	0.0313	**0.0313**	**0.0313**
Mental effort	0.0313	0.0938	**0.0313**	**0.0313**

Table 3 shows the MSE and ACC of the prediction results for four trained models. A smaller MSE (resp., larger ACC) represents a better prediction result. In summary, the results show that all the models successfully predict human shortest path task performance labels (i.e., efficiency, response time and accuracy). Specifically, MSP performs the best, demonstrating the success of the transfer learning, i.e., the importance of pre-training on graph drawing images with highlighted shortest path and the metric-based labels, and fine-tuning on graph drawing images with highlighted shortest path and task performance labels.

Note that M performs better than B, demonstrating the importance of the new path faithfulness metrics in predicting human shortest path task performance. Similarly, SP performs better than M, demonstrating the importance of graph drawing images with highlighted shortest path for predicting human shortest path task performance.

To validate whether the comparison between the models in Table 3 is statistically significant, we perform the Wilcoxon signed-rank test (Wilcoxon, 1992), a non-parametric statistical hypothesis test method to compare the pairwise models using scipy.stats.wilcoxon function with MSE and ACC values.

Table 4 shows the *p*-values of the Wilcoxon signed-rank tests for comparing the pairwise models. The *p*-value depends on the median MSE (resp., ACC) of the first model that is larger (resp., smaller) than the median MSE (resp., ACC) of the second model. Generally, a *p*-value $<$ 0.05 means that the first model is statistically significantly better than the second model.

Note that MSP performs better than M, where the difference for predicting all task performance labels is significant, which demonstrates the importance of fine-tuning on graph drawing images with highlighted shortest path and task performance labels. Furthermore, MSP performs better than SP, where the difference for predicting efficiency, accuracy and mental effort is significant, which demonstrates the importance of pre-training on the metric-based labels.

Overall, M performs better than B, where the difference is significant except for the accuracy, which demonstrates the importance of the new path faithfulness metrics in predicting human shortest path task performance. Similarly, SP performs better than M, where the difference is significant for the efficiency and accuracy, which demonstrates the importance of graph drawing images with highlighted shortest path for predicting human shortest path task performance.

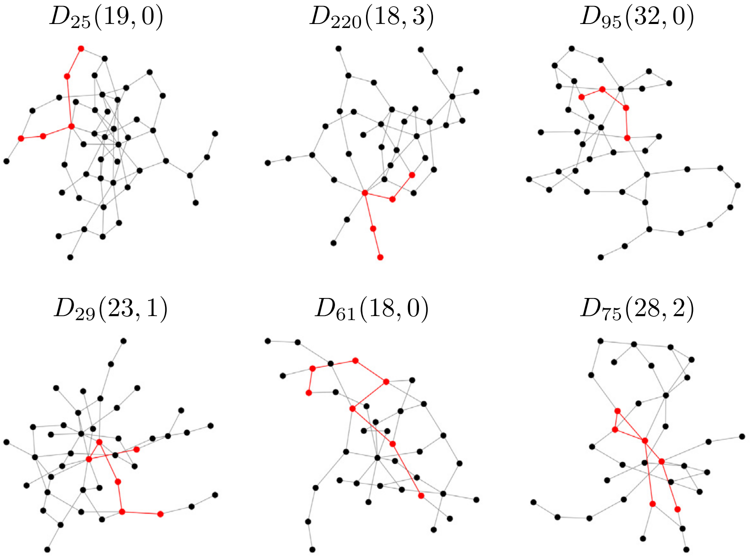


Fig. 6. Examples of short paths with good quality drawing (e.g., $D_{25}(19, 0)$, $D_{220}(18, 3)$ and $D_{95}(32, 0)$) and long paths with poor quality drawing (e.g., $D_{29}(23, 1)$, $D_{61}(18, 0)$ and $D_{75}(28, 2)$), where all three models succeed in predicting accuracy (resp., for efficiency and response time, the difference between the ground truth/predicted label is within ±0.6; for mental effort, the difference between actual/predicted value is within ±2).

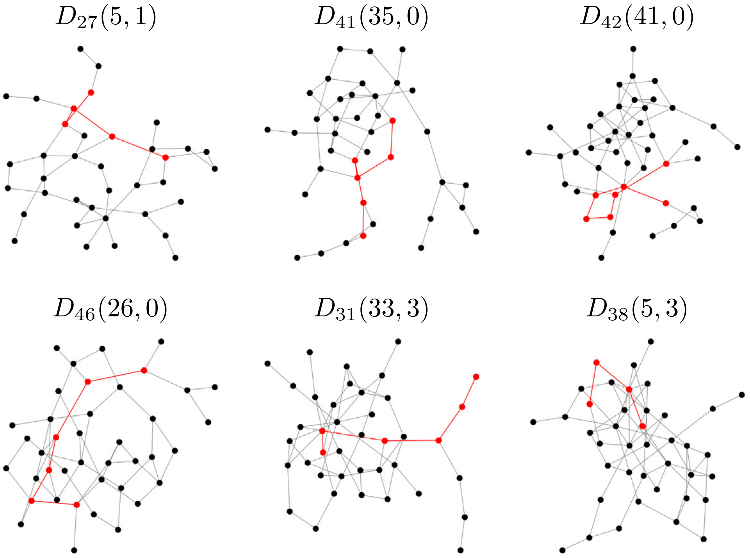


Fig. 7. Examples of path drawing with small angular resolution (e.g., $D_{27}(5, 1)$, $D_{41}(35, 0)$ and $D_{42}(41, 0)$) and high degree vertices around the path drawing (e.g., $D_{46}(26, 0)$, $D_{31}(33, 3)$ and $D_{38}(5, 3)$), where all three models fail to predict accuracy (resp., efficiency, response time and mental effort).

5.5. Discussion and summary

In summary, our extensive experiments show that the shortest path task performance can be predicted by a machine. In general, all the models succeed in predicting shortest path task performance labels (i.e., efficiency, response time and accuracy). In particular, MSP performs the best, demonstrating the importance of the transfer learning. M performs better than B, and SP performs better than M.

In particular, our models show good prediction results for the following instances:

- *Short path with good quality drawing:* when the path length is small (i.e., *PathLength* $\leq$ 4) and the quality of the path drawing is good, all three models successfully predict the accuracy.
  Fig. 6 shows examples (i.e., $D_{25}(19, 0)$, $D_{220}(18, 3)$ and D_{95} $(32, 0)$) of path drawings with large *pShape_KNN*, small *pAvgStress* and a few crossing (see Table 5), where all three models succeed to predict task performance labels (see Table 6).
- *Long path with poor quality drawing:* When the path length is long (i.e., *PathLength* $\geq$ 5), and the quality of graph drawing and path drawing are both bad, three models successfully predict the accuracy.
  Fig. 6 shows examples (i.e., $D_{29}(23, 1)$, $D_{61}(18, 0)$ and D_{75} $(28, 2)$) of graph drawings with small *Shape_GG* and many crossings, and path drawings with small *pShape_KNN*, large *pContinu* (zigzag path) and large *pGeode* (zigzag geodesic path) (see Table 5), where all three models succeed in predicting task performance labels (see Table 6).

Nevertheless, there are some difficult cases for prediction, although the quality metrics of the drawing are good:

Table 5
Quality metrics for the graph drawing (i.e., *Shape_GG*, *AvgStress* and *Crossing*); property and quality metrics for the path drawing (i.e., *PathLength*, *pShape_KNN*, *pAvgStress*, *pCrossNo*, *pMinAng*, *pContinu* and *pGeode*) for graph drawings shown in Fig. 6.

D_k	Shape_GG	AvgStress	Crossing
D_{25}	0.40	0.20	40
D_{220}	0.49	0.11	11
D_{95}	0.57	0.09	13
D_{29}	0.25	0.32	32
D_{61}	0.19	0.34	25
D_{75}	0.35	0.26	16

$D_k(s,t)$	PathLength	pShape_KNN	pAvgStress	pCrossNo	pMinAng	pContinu	pGeode
$D_{25}(19,0)$	4	0.58	0.06	1	0.96	0.33	0.25
$D_{220}(18,3)$	4	0.58	0.07	1	0.7	0.29	0.36
$D_{95}(32,0)$	3	1	0.02	1	0.72	0.28	0.25
$D_{29}(23,1)$	5	0.25	0.26	8	0.10	0.50	0.67
$D_{61}(18,0)$	6	0	0.37	11	0.23	0.73	0.58
$D_{75}(28,2)$	5	0.25	0.19	12	0.28	0.53	0.54

Table 6
The ground truth label and the predicted labels of model M, SP and MSP for graph drawings shown in Fig. 6.

$D_k(s,t)$	$L_{efficiency}$				L_{time}			
	Ground truth	*M*	*SP*	*MSP*	Ground truth	*M*	*SP*	*MSP*
$D_{25}(19,0)$	0.52	0.45	0.52	0.51	−0.17	−0.35	−0.20	−0.25
$D_{220}(18,3)$	0.46	0.30	0.37	0.47	−0.18	−0.19	−0.08	−0.23
$D_{95}(32,0)$	0.48	0.65	0.32	0.49	−0.35	−0.37	−0.27	−0.36
$D_{29}(23,1)$	−1.24	−1.23	−0.70	−1.02	1.27	0.76	0.41	0.60
$D_{61}(18,0)$	−1.38	−1.39	−1.48	−1.68	0.43	1.24	0.80	0.96
$D_{75}(28,2)$	−1.11	−1.38	−1.21	−1.45	0.49	0.93	0.72	0.83

$D_k(s,t)$	$L_{accuracy}$				L_{effort}			
	Ground truth	*M*	*SP*	*MSP*	Ground truth	*M*	*SP*	*MSP*
$D_{25}(19,0)$	1	1	1	1	2	1	3	3
$D_{220}(18,3)$	1	1	1	1	3	2	3	3
$D_{95}(32,0)$	1	1	1	1	3	1	3	1
$D_{29}(23,1)$	0	0	0	0	4	4	3	3
$D_{61}(18,0)$	0	0	0	0	3	4	5	5
$D_{75}(28,2)$	0	0	0	0	3	4	3	3

Table 7
Quality metrics for the graph drawing (i.e., *Shape_GG*, *AvgStress* and *Crossing*); property and quality metrics for the path drawing (i.e., *PathLength*, *pShape_KNN*, *pAvgStress*, *pCrossNo*, *pMinAng*, *pContinu* and *pGeode*) for graph drawings shown in Fig. 7.

D_k	Shape_GG	AvgStress	Crossing
D_{27}	0.44	0.15	9
D_{41}	0.36	0.28	18
D_{42}	0.28	0.34	37
D_{46}	0.27	0.42	32
D_{31}	0.38	0.37	47
D_{38}	0.18	0.34	75

$D_k(s,t)$	PathLength	pShape_KNN	pAvgStress	pCrossNo	pSPinAng	pContinu	pGeode
$D_{27}(5,1)$	4	0.58	0.16	3	0.94	0.25	0.36
$D_{41}(35,0)$	5	0.25	0.20	3	0.61	0.35	0.29
$D_{42}(41,0)$	6	0	0.32	4	0.34	0.71	0.66
$D_{46}(26,0)$	5	0.25	0.48	6	0.48	0.35	0.51
$D_{31}(33,3)$	5	0.25	0.44	7	0.24	0.37	0.34
$D_{38}(5,3)$	3	1.00	0.04	6	0.33	0.25	0.29

- *Path drawing with small angular resolution:* When a path drawing has a small angular resolution, even with a few crossings and small stress, it was difficult for a machine to predict shortest path task performance.
  Fig. 7 shows examples (i.e., $D_{27}(5,1)$, $D_{41}(35,0)$ and $D_{42}(41,0)$) of path drawings with small angular resolution (see the related metrics in Table 7 and predicted labels in Table 8).
- *High degree vertices around the path drawing:* When there are high degree vertices or overlap between the vertices and edges near the shortest path, even with small crossings, it was difficult for a machine to predict shortest path task performance.
  Fig. 7 shows examples (i.e., $D_{46}(26,0)$, $D_{31}(33,3)$ and $D_{38}(5,3)$) of graph drawings with high degree vertices around the path or vertices too close to the path (see the related metrics in Table 7 and predicted labels in Table 8).

The first case is due to the fact that the angular resolution of a path drawing was not included in the quality metrics for model training. Therefore, to improve the prediction results, we may

Table 8
The ground truth label and the predicted labels of model M, SP and MSP for graph drawings shown in Fig. 7.

$D_k(s,t)$	$L_{efficiency}$				L_{time}			
	Ground truth	M	SP	MSP	Groundtruth	M	SP	MSP
$D_{27}(5,1)$	−3.51	0.17	−0.12	0.14	3.10	−0.11	0.05	0.02
$D_{41}(35,0)$	−2.43	−0.10	−0.07	−0.02	2.30	0.12	0.09	0.08
$D_{42}(41,0)$	−4.37	−0.94	−1.08	−1.52	3.24	0.82	1.13	0.47
$D_{46}(26,0)$	−3.62	−0.77	−0.62	−0.83	1.92	0.59	0.54	0.39
$D_{31}(33,3)$	−2.98	−0.85	−0.23	−0.40	1.64	0.58	0.18	0.23
$D_{38}(5,3)$	−3.16	0.09	−0.18	−0.12	1.40	−0.13	−0.21	0.04

$D_k(s,t)$	$L_{accuracy}$				L_{effort}			
	Ground truth	M	SP	MSP	Groundtruth	M	SP	MSP
$D_{27}(5,1)$	0	1	1	1	6	2	3	3
$D_{41}(35,0)$	0	1	1	1	5	3	3	3
$D_{42}(41,0)$	0	1	1	1	8	4	3	3
$D_{46}(26,0)$	0	1	1	1	8	3	3	3
$D_{31}(33,3)$	0	1	1	1	7	3	3	3
$D_{38}(5,3)$	0	1	1	1	8	2	3	3

need to include the angular resolution metrics of a path drawing for model training.

Similarly, the second case is due to the fact that structural graph properties such as high degree vertices, as well as the quality of drawings of subgraphs near the path drawing, were not considered for model training. Therefore, to improve the prediction results, we may need to consider high degree vertices and the quality metrics of drawings near the path drawing for model training.

5.6. Implication, potential application and limitation

In this paper, we show that a machine can successfully predict the *qualitative* evaluation in graph drawings, by considering the most fundamental task, i.e., the shortest path task, advancing the known results in the literature for predicting *quantitative* evaluation (such as quality metrics) in graph drawings.

In contrast to quantitative evaluation, where quality metrics can be easily computed (e.g., in $O(nlogn)$ time for edge crossings), qualitative evaluation requires significant time and effort to conduct controlled human experiments, including the ethics approval. Therefore, a potential practical application of our model is to save time and effort for qualitative evaluation of a graph drawing, by predicting human shortest path task performance without conducting a controlled human experiment.

Although our model can successfully predict the human shortest task performance, there are some limitations. Our model is based on the ground truth human experiment data, which uses a force-directed layout of small and sparse Rome graphs. Therefore, it may not generalize well for different types of graphs (e.g., large and complex graphs) and different graph layouts.

6. Conclusion and future work

We present the first machine learning approach to predict human shortest path task performance, including efficiency, accuracy, response time, and mental effort, utilizing correlated quality metrics, the ground truth shortest path experiments data, and transfer learning.

Specifically, we introduce *path faithfulness metrics* and show strong correlations with the shortest path task performance. Moreover, we use the transfer learning method to pre-train our deep model, exploiting the most correlated quality metrics (i.e., *pCrossNo*, *pGeode*, *pShape_KNN*, *PathLength*, *pContinu*, *pAvgStress* and *pMinAng*) to mitigate the problem of insufficient ground truth training data.

Experimental results show that our models can successfully predict the shortest path task performance. In particular, MSP performs the best, achieving an MSE of 0.7243 (i.e., data range from −17.27 to 1.81) for prediction, demonstrating the success of transfer learning using the correlated metrics.

While, in general, our trained models show good prediction performance, there are some difficult cases for prediction, e.g., path drawings with small angular resolution and high degree vertices around the path drawing. Therefore, our future work is to design new quality metrics to better measure the quality of path drawings to improve prediction for such cases.

Moreover, we plan to conduct a new human experiment using various graph types and graph layouts, to generate new ground truth shortest path performance data, which will be used to improve our model and overcome current limitations.

CRediT authorship contribution statement

Shijun Cai: Investigation, Writing – original draft, Software, Validation, Visualization. **Seok-Hee Hong:** Conceptualization, Supervision, Methodology, Investigation, Validation, Writing – review & editing, Funding acquisition. **Xiaobo Xia:** Software. **Tongliang Liu:** Methodology, Supervision, Validation. **Weidong Huang:** Data curation, Resources, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Research supported by ARC Linkage Project, Australia (LP160100935) with Oracle Research lab.

Ethical approval

All data used in the study are taken from public databases that were published in the past.

References

Batagelj, V., Mrvar, A., 2004. Pajek—analysis and visualization of large networks. In: Graph Drawing Software. Springer, pp. 77–103.

Benesty, J., Chen, J., Huang, Y., Cohen, I., 2009. Pearson correlation coefficient. In: Noise Reduction in Speech Processing. Springer, pp. 1–4.

Cai, S., Hong, S., Shen, J., Liu, T., 2021. A machine learning approach for predicting human preference for graph layouts. In: PacificVis. pp. 6–10.

Chimani, M., Eades, P., Eades, P., Hong, S., Huang, W., Klein, K., Marner, M., Smith, R.T., Thomas, B.H., 2014. People prefer less stress and fewer crossings. In: Proceedings of International Symposium on Graph Drawing. Springer, pp. 523–524.

Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G., 1999. Graph Drawing, Vol. 357.

Eades, P., Hong, S., Klein, K., Nguyen, A., 2015. Shape-based quality metrics for large graph visualization. In: International Symposium on Graph Drawing. pp. 502–514.

Fletcher, C., Huang, W., Arness, D., Nguyen, Q.V., 2019. The role of working memory capacity in graph reading performance. In: PacificVis. IEEE, pp. 77–81.

Giovannangeli, L., Bourqui, R., Giot, R., Auber, D., 2020. Toward automatic comparison of visualization techniques: Application to graph visualization. Vis. Inform. 4 (2), 86–98.

Haleem, H., Wang, Y., Puri, A., Wadhwa, S., Qu, H., 2019. Evaluating the readability of force directed graph layouts: A deep learning approach. IEEE Comput. Graph. Appl. 39 (4), 40–53.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Huang, W., Eades, P., Hong, S., 2009. A graph reading behavior: Geodesic-path tendency. In: PacificVis. IEEE, pp. 137–144.

Huang, W., Eades, P., Hong, S., 2014. Larger crossing angles make graphs easier to read. J. Vis. Lang. Comput. 25 (4), 452–465.

Huang, W., Hong, S., Eades, P., 2008. Effects of crossing angles. In: PacificVis. pp. 41–46.

Huang, W., Huang, M.L., Lin, C., 2016. Evaluating overall quality of graph visualizations based on aesthetics aggregation. Inform. Sci. 330, 444–454.

Klammler, M., Mchedlidze, T., Pak, A., 2018. Aesthetic discrimination of graph layouts. In: International Symposium on Graph Drawing and Network Visualization. pp. 169–184.

Kwon, O., Crnovrsanin, T., Ma, K., 2017. What would a graph look like in this layout? a machine learning approach to large graph visualization. TVCG 24 (1), 478–488.

Kwon, O., Ma, K., 2019. A deep generative model for graph layout. TVCG 26 (1), 665–675.

Meidiana, A., Hong, S., Eades, P., 2020a. New quality metrics for dynamic graph drawing. In: International Symposium on Graph Drawing and Network Visualization. pp. 450–465.

Meidiana, A., Hong, S., Eades, P., Keim, D., 2019. A quality metric for visualization of clusters in graphs. In: International Symposium on Graph Drawing and Network Visualization. pp. 125–138.

Meidiana, A., Hong, S., Eades, P., Keim, D., 2020b. Quality metrics for symmetric graph drawings. In: PacificVis. pp. 11–15.

Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Purchase, H., 1997. Which aesthetic has the greatest effect on human understanding. In: International Symposium on Graph Drawing, Vol. 1353. p. 248.

Toussaint, G.T., 2014. Computational Morphology: A Computational Geometric Approach to the Analysis of Form. Elsevier.

Wang, Q., Chen, Z., Wang, Y., Qu, H., 2020. Applying machine learning advances to data visualization: A survey on ML4VIS. arXiv preprint arXiv:2012.00467.

Ware, C., Purchase, H., Colpoys, L., McGill, M., 2002. Cognitive measurements of graph aesthetics. Inf. Vis. 1 (2), 103–110.

Wilcoxon, F., 1992. Individual comparisons by ranking methods. In: Breakthroughs in Statistics. Springer, pp. 196–202.