Land Deformation Prediction via Slope-Aware Graph Neural Networks

Fan Zhou[1*], **Rongfan Li**[1], **Goce Trajcevski**[2], **Kunpeng Zhang**[3]
[1] University of Electronic Science and Technology of China
[2] Iowa State University
[3] University of Maryland, College Park
fan.zhou@uestc.edu.cn, rongfanli1998@gmail.com, gocet25@iastate.edu, kpzhang@umd.edu

Abstract

We introduce a slope-aware graph neural network (SA-GNN) to leverage continuously monitored data and predict the land displacement. Unlike general GNNs tackling tasks in the plain graphs, our method is capable of generalizing 3D spatial knowledge from InSAR point clouds. Specifically, we structure of the land surface, while preserving the spatial correlations among adjacent points. The point cloud can then be efficiently converted to a near-neighbor graph where general GNN methods can be applied to predict the displacement of the slope surface. We conducted experiments on real-world datasets and the results demonstrate that SA-GNN outperforms existing 3D CNN and point GNN methods.

Introduction

Landslides are geological hazards that can result in significant fatalities and economic losses. They occur because of the gradual moving of soils, debris, and rocks on hills, caused by various factors, such as water fluctuation, heavy rainfall, soil erosion, and earthquakes. For example, the 2008 Wenchuan earthquake has induced more than 60,000 landslides, among which the Daguangbao landslide is one of the most massive in the world, with a volume of displaced mass exceeding 1.16×10^9 m^3 (Huang and Fan 2013). Landslides and mudslides are also significant threats for infrastructures and residents near hydropower stations. Therefore, monitoring and preventing such disasters have received considerable attention from both industry and academia (Bozzano et al. 2011; Gao, Dai, and Chen 2020; Hajimoradlou, Roberti, and Poole 2020).

The existing approaches for landslide prediction fall into three main categories: expert-based, monitoring-based, and machine learning-based ones. Expert-based methods (Gao, Dai, and Chen 2020) rely on domain knowledge of experts, requiring case by case judgement, and is often post-explained. Monitoring-based approaches (Gan, Yang, and Zhou 2019) conduct field observations and use real-time monitoring data (e.g., soil, rocks, and rainfall) to study the deformation characteristics and monitor specific events that can potentially trigger landslides. The line of works employing machine learning techniques for landslide prediction includes Bayes networks (Shirzadi et al. 2017), logistic regression (LR) (Kalantar et al. 2018), decision trees and random forest (Chen et al. 2017), support vector machines (SVM) (Hong et al. 2016) and neural networks (Ghorbanzadeh et al. 2019; Lei et al. 2019).

Convolutional neural networks (CNNs) are used in (Hajimoradlou, Roberti, and Poole 2020) to generate landslide susceptibility maps, while considering the orientation of each pixel at multiple scales, incorporating the slope and uphill/downhill directions for learning hidden features. However, it applies 2D CNNs on maps, which may make it unable to fully capture the spatial correlations (e.g., distance and elevation) among monitored points, due to the limited image resolution. The development of satellite Interferometric Synthetic Aperture Radar (InSAR) allows to generate point cloud maps of slope surfaces and identify precursors to catastrophic landslides (Carlà et al. 2019; Dong et al. 2019). The importance of satellite InSAR on enhancing the predictive ability of slope failures was highlighted in (Carlà et al. 2019), exploring three major slope failures – an open-pit mine slope, a natural rock slope in alpine terrain, and a breakdown of a tailings dam embankment – focusing on quantitative analysis of the InSAR point cloud data, without particular predictive model. In contrast (Dong et al. 2019) proposed two complementary approaches to correct the stratified tropospheric delays for time series InSAR data, that may introduce seasonal oscillation biases into slope monitoring. Despite the considerable efforts in prior studies, little research has looked into the *prediction* of continuous slope deformation, which is the objective of our study.

The satellite InSAR point cloud data contains rich spatial-temporal information associated with the measured points as well as accurate surface deformation (i.e., millimetric measurement accuracy). This characteristic suggests using graph neural network (GNN) (Wu et al. 2020) as a compact representation of a point cloud and iteratively aggregating the point features from the measurements of adjacent points. Recent studies (Wang et al. 2019; Shi, Ragunathan, and Rajkumar 2020) in computer vision have represented point clouds as graphs, employing GNNs for 3D object detection, classification, and semantic segmentation of point clouds. However, these methods cannot be directly applied for deformation prediction because they emphasize identifying the object shapes without considering the unique

*Corresponding author

features of the terrain surface, e.g., orientations, slope, and relative distances between points. The terrain surface is not stable but gradually and continuously changes due to, for example, sedimentation, erosion, and deposition. Moreover, the displacement of a point may affect its surrounding locations and vice versa, while the influence between points is not fixed but is strongly related to the mutual distance, azimuth, and gradient.

In this work, we propose an approach to address the challenges in surface deformation monitoring using InSAR data. Specifically, we present a method to preserve the manifold structures of the surface points based on locally linear embedding (LLE) and reconstruct the 3D surface with a neighbor graph. Our SA-GNN model considers the intricate spatial dependency between adjacent points and allows each point in a graph to be aware of its nearby terrain deformation by exploiting their inter-dependency among temporal measurements. Following are our main contributions:

- To our knowledge, we are among the first to present landslide monitoring and surface deformation prediction from a graph-based perspective by adapting GNN to InSAR point clouds. We believe this work can foster more engagement in applying AI for securing humans, environments, and critical infrastructures.
- We propose a metric learning method to learn mutual spatial dependencies in the point cloud and maintain the manifold structure of terrains to handle implicit correlations among local deformations. Our goal is to provide an alternative view of learning complicated structures of point clouds beyond the object shapes.
- We conduct experiments on real-world datasets collected by monitoring activities spanning more than nine months. We show that our method outperforms previous GNN-based methods on predicting the temporal deformation, which demonstrates its inherent capability of dynamically and continuously warning the catastrophic slope failures.

Related Work

Landslide Mapping & Predicting. Timely and accurately predicting landslides is a paramount in the geology community. Previous works have focused on generating proper landslide susceptibility mapping (LSM) using optical satellite images and aerial photographs due to increased accessibility to high-resolution remote sensing data (Bozzano et al. 2011; Gao, Dai, and Chen 2020). The aim is to identify geological areas that are prone to catastrophic slope failures, as well as terrain deformations susceptible to landslides. For example, (Shirzadi et al. 2017) leverages naïve Bayes trees for LSM, using random sub-space to generate subsets from training data and then constructing a primary classifier with tree structures. (Chen et al. 2017) trains a SVM with various kernels to calculate the susceptibility indexes for all pixels in the study area. (Lei et al. 2019) employs a CNN with pyramid pooling to learn image features for improving LSM results and landslide recognition, where morphological reconstruction and clustering are applied to distinguish landslide areas.

Two experimental studies (Wang, Fang, and Hong 2019; Ghorbanzadeh et al. 2019) evaluate CNNs and typical machine learning methods on susceptibility detection and investigate the impact of spectral and topographic factors on LSM, indicating that CNNs are more practical for LSM and landslide prevention than conventional methods. Recently, (Hajimoradlou, Roberti, and Poole 2020) proposed locally aligned CNN filters to capture the orientation of each pixel at multiple resolutions for landslide identification. Previous works either study landslide susceptibility using 2D geospatial images (Wang, Fang, and Hong 2019; Ghorbanzadeh et al. 2019; Hajimoradlou, Roberti, and Poole 2020) or convert the InSAR data into a 2D bird's view images and then apply statistical methods for landslide prediction (Dai et al. 2016; Carlà et al. 2019). Such approaches ignore critical spatial dependencies among monitored areas and induce prediction errors due to the limited image resolution and the absence of significant information regarding the land surface (e.g., gradient, azimuth, and mutual influence).

Graph Neural Networks. GNNs are powerful models for learning rich relational information in graphs by aggregating features from adjacent nodes/edges, emerging as *de facto* models for learning graph-structured data in a variety of domains such as social networks, molecular/biological topology, and knowledge graphs (Wu et al. 2020). The general GNNs seek to generalize knowledge among nodes/edges and to learn better representations, but fail to capture the node position within the context of the graph structure. Spatial and temporal attributes associated with nodes have been exploited for certain applications, e.g., traffic forecasting (Li et al. 2018) and urban flow prediction (Wang et al. 2020). Existing spatio-temporal GNNs (ST-GNN) model road sensor networks or urban areas as 2D graphs and cannot be directly applied for point cloud data. Point-GNN (Shi, Ragunathan, and Rajkumar 2020), while predicting the category and shape of objects described by point clouds, aims at object detection through discriminating the bounding box a node belongs to, and cannot handle the intra-nodes relations beyond locations.

Our SA-GNN is specifically designed for landslide prediction. Instead of simply converting a point cloud to a regular image or directly distinguishing single vertex, we use a slope-aware locally linear embedding (LLE) module to preserve the spatial characteristics of a point cloud. Unlike the ST-GNNs that aggregate node attributes constrained by the 2D spatial locations, we embed the slope information and interactive influence among nodes into the graph and continuously monitor and predict the surface deformation by iteratively updating local-level node representations.

Preliminaries

Dataset Our data is collected from the slopes on both sides of a large-scale hydropower dam. We use InSAR technology to monitor the surface displacement over time – from Nov 30, 2018 to Sep 8, 2019. During this period of time, there were several slope failures, ranged from 600 to 10,000 m^3. The most serious landslide occurred in Aug 17, 2019, which caused around 10,000 m^3 volume collapse and 3-8m road sank. The displacement of the monitored points – which

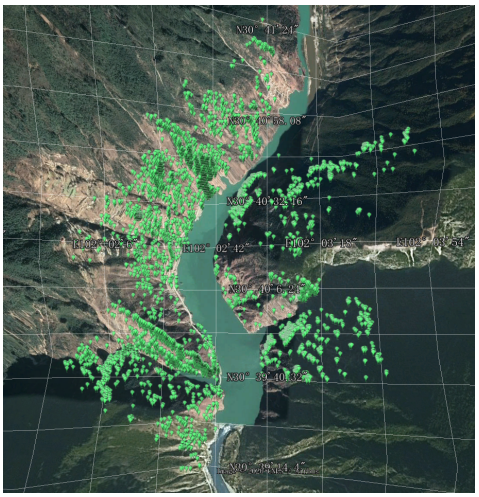


Figure 1: Bird's-eye view of the studied reservoir areas above a large dam monitored for 9 months.

Dataset	West side	East side
# nodes	4,569	2,164
Longitude (E102°)	[1'50", 3'2"]	[2'35", 3'46"]
Latitude (N30°)	[39'12", 41'25"]	[39'38", 40'48"]
Elevation	[1671.2, 2527.4]	[1470.2, 2899.6]
Displacement	[-27.58, 28.03]	[-29.06, 30.50]

Table 1: Statistics of datasets, the surface displacement information is collected from Nov 30, 2018 to Sep 8, 2019.

can be either vertical or horizontal – ranges in [-27.58mm, 28.03mm] and [-29.06mm, 30.5mm] on west and east side, respectively. Table 1 shows the statistics of the dataset. Figure 1 plots the studied areas, where dots alongside the river denote the monitored locations.

Problem Settings. We now discuss the context and formally define the prediction problem.

Point cloud: The InSAR point cloud consists of N monitored locations $L = \{l_1, \ldots, l_N\}$, where $l_i = (\mathbf{g}_i, \mathbf{m}_i)$ is a location with a 3D geographical coordinates $\mathbf{g}_i \in \mathbb{R}^3$ (longitude, latitude, elevation), and a sequence of displacement measurements $\mathbf{m}_i = \{m_i^1, \ldots, m_i^T\}$.

Near-neighbor graph: We construct a graph $\mathcal{G} = (L, E)$ using L as vertices and connecting a point to its neighbors within a fixed distance ε i.e., there is an edge e_{ij} between vertices l_i and l_j if their distance is within ε.

Displacement prediction: For a point cloud graph $\mathcal{G}$ and historical displacement observations $\mathbf{M} = (\boldsymbol{m}^1, \ldots, \boldsymbol{m}^t, \ldots, \boldsymbol{m}^T)$ – where $\boldsymbol{m}^t$ is the measurement of all nodes at time t, the objective of displacement prediction is to learn a model $\mathrm{F}(\cdot)$ that outputs the future displacement of all N locations over the next T' time steps $\widehat{\mathbf{M}}' = (\hat{\boldsymbol{m}}^{T+1}, \ldots, \hat{\boldsymbol{m}}^{T+T'})$ – i.e., we learn a nonlinear probabilistic function to predict the future displacements as:

$$\mathrm{F}(\mathbf{M}; \mathcal{G}) = \widehat{\mathbf{M}}' = \left(\hat{\boldsymbol{m}}^{T+1}, \ldots, \hat{\boldsymbol{m}}^{T+t'}, \ldots, \hat{\boldsymbol{m}}^{T+T'}\right).$$

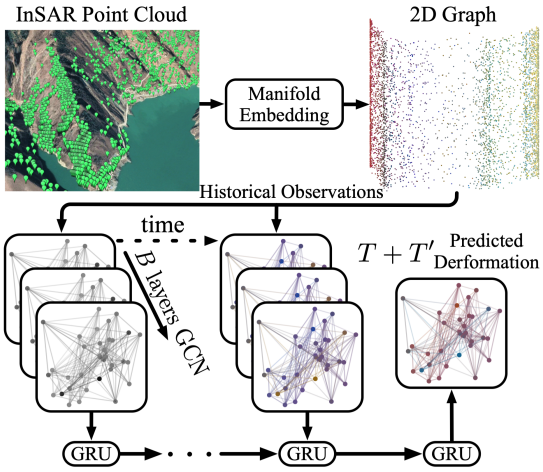


Figure 2: The framework of the proposed SA-GNN.

Methodology: SA-GNN

We present our proposed approach SA-GNN to predict landslide susceptibility from an InSAR point cloud. SA-GNN consists of three main components (cf. Figure 2): (**A**) a weighted metric learning, employed to encode the manifold structure and local spatial correlations of terrain surface; (**B**) a near-neighbor graph that is built by reconstructing the point cloud structure with slope information; (**C**) a GNN, employed to iteratively capture the nodes' interactions and predict the deformation of the surface.

Manifold Structure Embedding

A direct solution is to convert the point cloud into a 2D image and then apply CNNs to learn spatial features of the studied areas. This, however, is limited by the 2D image resolution and may introduce quantization errors in the projection procedure. Recently, (Shi, Ragunathan, and Rajkumar 2020) constructed graphs based on the near-neighbors within a cut-off distance, but still ignored intra-node correlations such as azimuth and slope. To overcome this issue, we propose to use metric learning to encode the manifold structure of the terrain surface and preserve the mutual influence among the monitored locations.

Here we present a weighted locally linear embedding (WLLE) to project the manifold structure of point cloud into a 2D space. It extends locally linear embedding (LLE) (Roweis and Saul 2000), which reconstruct each node l_i on the manifold using its K nearest neighbors $\mathcal{N}(l_i)(j \in \mathcal{N}(l_i)$ if l_j is a neighbor of $l_i)$:

$$\min_{w_{ij} \in \mathbf{W}} \sum_i \|\mathbf{g}_i - \sum_{j \in \mathcal{N}(l_i)} w_{ij} \mathbf{g}_j\|_2^2 \qquad (1)$$

i.e., it adds up the squared distances between all locations and their neighborhood reconstructions. Generally, we normalize w_{ij} by constraining $\sum_j w_{ij} = 1$. The optimal $\mathbf{W}$ can be found by solving a least-square problem using Lagrange multiplier method.

LLE and its variants (Donoho and Grimes 2003; Zhang and Zha 2004; Zhang and Wang 2007), discriminate K near-

est neighbors via weights w_{ij} but may not be able to distinguish their relative positions when projecting into a lower dimensional space. In landslide prediction, a node's azimuth can strongly affect the prediction accuracy of surrounding areas – e.g., a landslide (or dramatic deformations) of the monitored areas above a point has a higher probability for causing the displacement of that point. To capture such correlations, WLLE accounts for the distance and slope between each node l_i and its neighbors.

Specifically, for all K neighbors of l_i, we have $\mathbf{g}_i = \mathbf{g}_j + (\mathbf{g}_i - \mathbf{g}_j)$, which means we can use an extra vector $(\mathbf{g}_i - \mathbf{g}_j)$ to reconstruct $\mathbf{g}_i$. Given a point l_i, we measure the relevance between l_i and its neighbors by learning the coefficient w_{ij}:

$$\min_{w_{ij} \in \mathbf{W}} \|\mathbf{g}_i - \frac{1}{K} \sum_{j \in \mathcal{N}(l_i)} \big(\mathbf{g}_j + w_{ij}(\mathbf{g}_i - \mathbf{g}_j)\big)\|_2^2, \quad (2)$$

which is subjected to $\sum_j w_{ij} = 1$. When minimizing this function, it tends to allocate more weights to those nodes with larger values of $(\mathbf{g}_i - \mathbf{g}_j)$. Therefore, we add a L_2 regularization (with a weight factor γ) to avoid excessive imbalance.

In addition, we would like to enrich the weight w_{ij} with relative positions between two nodes. Let $d_{ij} = \|\mathbf{g}_i - \mathbf{g}_j\|_2$ and s_{ij} be the straight-line distance and slope between two points, respectively. We assume a linear dependency $m'_{ij} = \alpha s_{ij} + (1-\alpha)/d_{ij}$, where α is a weight balancing the two factors and m'_{ij} is normalized by $m_{ij} = m'_{ij}/\sum_j m'_{ij}$ so that $\sum_j m_{ij} = 1$. The basic idea is to upweight the node pairs that are more relevant by compensating w_{ij} with m_{ij}. The main objective of WLLE can now be summarized as:

$$\min_{w_{ij} \in \mathbf{W}} \sum_i \|\mathbf{g}_i - \frac{1}{K} \sum_{j \in \mathcal{N}(l_i)} \big(\mathbf{g}_j + (w_{ij} + m_{ij})(\mathbf{g}_i - \mathbf{g}_j)\big)\|_2^2 + \gamma \|\mathbf{w}_i\|_2, \quad (3)$$

$$\text{s.t.} \sum_{j \in \mathcal{N}(l_i)} w_{ij} = 1. \quad (4)$$

Solving Eq. (3) can be done using Lagrange multiplier method. After obtaining w_{ij}, we can analytically compute the sum $h_{ij} = w_{ij} + m_{ij}$, which has a closed form as m_{ij} is fixed between any two points. Note that $\sum_j h_{ij} = 2$ since both $\sum_j w_{ij} = 1$ and $\sum_j m_{ij} = 1$. To obtain the embedded vector $\mathbf{t}_i \in \mathbf{T}$ for each point l_i, we optimize the following equation:

$$\min_{\mathbf{T}} \quad \sum_i \|K\mathbf{t}_i - \sum_j \big(\mathbf{t}_j + h_{ij}(\mathbf{t}_i - \mathbf{t}_j)\big)\|_2^2 \quad \text{s.t.} \quad \mathbf{T}\mathbf{T}^\intercal = \mathbf{I} \quad (5)$$

Graph Convolution and Prediction

Graph Build. With embedding matrix $\mathbf{T}$ in hand, we calculate the Euclidean distance $dist(\mathbf{t}_i - \mathbf{t}_j)$ between two point vectors in the 2D space, which can be readily used to construct the neighbor graph $\mathcal{G}$ by setting a threshold ε, i.e., there is an edge between point l_i and l_j if $dist(\mathbf{g}_i - \mathbf{g}_j) \leq \varepsilon$.

Spatial Dependency Modeling. Since the influence of surrounding points is critical in slope deformation prediction, it is straightforward to learn such dependencies using GNNs, which generalize convolution operations in the graph by flexibly and aggressively capturing features of neighbors in the non-Euclidean space. Given an adjacency matrix $\mathbf{A}$ and a feature matrix $\mathbf{M}$, we use GCN (Kipf and Welling 2017) to perform local feature aggregations:

$$\mathbf{X}^{(b+1)} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{X}^{(b)}\theta^{(b)}\right), \quad (6)$$

where $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ is the diagonal matrix of node degrees, $\mathbf{X}^{(b)}$ is the input of b-th layer with trainable parameters $\theta^{(b)}$, and we have B layers in total. We initialize $\mathbf{X}^{(0)} = \mathbf{M}$, and use ReLU as the activation function $\sigma(\cdot)$. Note that our method does not rely on a particular GNN model. Although we employed GCN for nodes' feature aggregation, any other GNN models such as GAT, GraphSAGE, and GIN can be easily used to replace the GCN in our framework.

Temporal Dependency Modeling. To capture the temporal dependency of deformation, we adopt the GRU (Chung et al. 2014) to process the sequence information of each node individually, while sharing the parameters of the GRUs with each other for all the nodes.

Objective. The objective of our model is to minimize the loss between the ground-truth ($\mathbf{M}'$) and the predicted ($\widehat{\mathbf{M}}'$) deformation values of each point:

$$\mathcal{L} = \|\mathbf{M}' - \widehat{\mathbf{M}}'\|_2 + L_2, \quad (7)$$

where L_2 regularization is used to avoid overfitting.

Empirical Evaluation

We now report the experimental observations comparing SA-GNN with baselines and analyzing its benefits.

Baselines. We compare our proposed model to the following baselines for predicting the time series of displacement: (1) *Historical Average (HA)*: models historical data average of a certain time period T to predict the deformation in the next time step. We set T = 2. (2) *ARIMA*: is a widely used time-series model that combines auto-regressive and moving average for prediction. (3) *SVR*: is a typical time-series model predicting the value at a future time step with linear support vector regression. (4) *LSTM* and *GRU*: are variants of RNN that captures the information of long-short term dependency, and have been widely used for time-series forecasting. (5) *STGCN*: spatio-temporal GCNs model the time-series with an external graph structure which describes the relationships between nodes' geographical relations, which has been widely used for graph-based time-series prediction such as traffic prediction and action recognition (Wu et al. 2020). Here we use GCN to capture the 2D spatial dependencies among nodes and use GRU to model the historical observations and to make predictions. (6) *Point-GNN* (Shi, Ragunathan, and Rajkumar 2020): is a point cloud graph convolution method that is originally proposed for object prediction. It directly constructs a graph based on the Euclidean distance between nodes on the point cloud. We adapt

Method	West Side					East Side				
	RMSE	**MAE**	**ACC**	**R^2**	**EVS**	**RMSE**	**MAE**	**ACC**	**R^2**	**EVS**
HA	3.144	2.454	0.047	0.134	0.262	3.858	2.870	0.046	0.224	0.288
SVR	6.872	5.528	0.018	0.036	0.025	8.735	6.749	0.016	0.021	0.017
ARIMA	4.764	3.947	0.041	0.072	0.157	8.326	6.865	0.021	0.052	0.185
LSTM	0.254	0.218	0.490	0.038	0.094	0.254	0.210	0.518	0.077	0.086
GRU	0.254	0.217	0.491	0.040	0.095	0.250	0.207	0.526	0.078	0.092
STGCN	0.151	0.124	0.834	0.254	0.526	0.155	0.120	0.834	0.373	0.465
Point-GNN	0.177	0.148	0.725	0.121	0.409	0.174	0.142	0.749	0.098	0.324
SA-GNN	**0.094**	**0.067**	**0.954**	**0.668**	**0.669**	**0.115**	**0.081**	**0.911**	**0.657**	**0.661**

Table 2: Performance comparison of displacement prediction on both sides of hills. For RMSE and MAE, the lower the value, the better the performance. Conversely, higher values are desirable for ACC, R^2 and EVS.

it to displacement prediction by using GRU to model the temporal dependencies.

Experimental settings. Since the time span of the dataset is around 9 months (280 days), we used the first 140 days ($\sim$50%) data for training and the remaining for validation (84 days, $\sim$30%) and testing (56 days, $\sim$20%). All deep learning models, including ours, are tuned to the best performance with early stopping when validation loss has not declined for 40 consecutive epochs. We use Adam optimizer with an initial learning rate of $3e^{-4}$, which decays with the rate of 0.9 every 100 epochs. Unless otherwise specified, for all manifold learning methods, the parameter K is tuned as 450 and 700 for west and east side, respectively. For our WLLE, the default value of α is 0.7 and γ is 100. Lastly, the graphs were built using threshold distance $\varepsilon = 100$ for west side and $\varepsilon = 150$ for east side.

Evaluation protocols. We use five evaluation methods to measure the prediction performance: root mean squared error (RMSE), mean absolute error (MAE), accuracy (ACC), coefficient of determination (R^2), and explained variance score (EVS). RMSE, MAE, and ACC are widely used metrics for evaluating time-series models. R^2 measures the amount of variation explained by the (least-squares) linear regression. EVS reports the total variance explained by factors that are actually present rather than due to an error variance.

Overall performance comparison. Table 2 summarizes the performance of different methods on predicting the surface displacement of both sides, from which we have the following observations: (1) SA-GNN consistently outperforms all baselines on both land sides, demonstrating its effectiveness of learning surface structures for terrain deformation forecasting. (2) Traditional time-series prediction methods such as HA, SVR, and ARIMA are not comparable due to lack of graph-structure modeling capability, which is also shown with GNN-based methods generally outperforming them solely based on single area monitoring. These results also demonstrate that the deformation of a specific monitored location is strongly affected by surrounding areas. (3) The improvements of SA-GNN over STGCN indicate that simply modeling point cloud data with 2D spatial graph is not enough for surface displacement prediction, largely because it overlooks the complex structures of the terrain surface. Similarly, point-GNN, which encodes the point cloud as near-neighbor graph using fixed cut-off distance on the manifold, does not show competitive results, which demonstrated the effectiveness of encoding relative distance and slope between adjacent nodes in SA-GNN. Point-GNN performs even poorer than STGCN due to the biases introduced when constructing the graph. In other words, simply building graphs on point cloud does not preserve well the intricate relations and positions of the points. (4) Finally, we found that all models, including our SA-GNN, generally perform better on the west side hill. This phenomenon indicates that the more monitored points, the better the prediction performance – recall that east slope is sparser in points (cf. Figure 1).

Ablation Study. We now verify two important motivations of this work, i.e., relative position calculation on the point cloud is biased and graph building requires a careful relative position embedding.

	Model	RMSE	MAE	ACC	R^2	EVS
West	Eucli.	0.229	0.182	0.621	0.011	0.046
	Slope	0.175	0.136	0.762	0.107	0.155
	WLLE	**0.094**	**0.067**	**0.954**	**0.668**	**0.669**
East	Eucli.	0.232	0.187	0.621	0.085	0.144
	Slope	0.173	0.128	0.810	0.221	0.287
	WLLE	**0.115**	**0.081**	**0.911**	**0.657**	**0.661**

Table 3: Comparison among different graph constructions.

– *Effect of relative position embedding.* We first investigate the effect of the relative position embeddings used in SA-GNN. Towards that, we use two alternative metrics to construct the graph. In Euclidean and Slope, we respectively use the Euclidean distance and slope between two points to construct the graph. Note that we directly calculate the distance and slope on the point cloud without embedding the manifold to 2D space. The results are shown in Table 3, where we can observe that computing the Euclidean distance or slope on the point cloud is not effective, which justifies the main motivation of this work, i.e., relative positions calculated directly on the point cloud would introduce biases when build-

ing the graph and therefore leads to poor prediction performance. The results also suggest that slope is more important than distance on deformation prediction which, one may argue that it is intuitive due to the influence of gravity.

	Model	RMSE	MAE	ACC	R^2	ESV
West Side	LLE	0.123	0.095	0.910	0.440	0.548
	MLLE	0.113	0.086	0.933	0.523	0.601
	HLLE	0.114	0.087	0.931	0.515	0.587
	LTSA	0.142	0.115	0.861	0.248	0.517
	Isomap	0.126	0.069	0.901	0.407	0.464
	t-SNE	0.151	0.111	0.849	0.155	0.291
	UMAP	0.121	0.087	0.913	0.457	0.460
	WLLE	**0.094**	**0.067**	**0.954**	**0.668**	**0.669**
East Side	LLE	0.142	0.109	0.870	0.474	0.558
	MLLE	0.128	0.096	0.894	0.574	0.620
	HLLE	0.135	0.103	0.886	0.528	0.605
	LTSA	0.121	0.088	0.906	0.621	0.645
	Isomap	0.146	0.112	0.863	0.448	0.549
	t-SNE	0.147	0.106	0.859	0.438	0.450
	UMAP	0.132	0.095	0.887	0.547	0.551
	WLLE	**0.115**	**0.081**	**0.911**	**0.657**	**0.661**

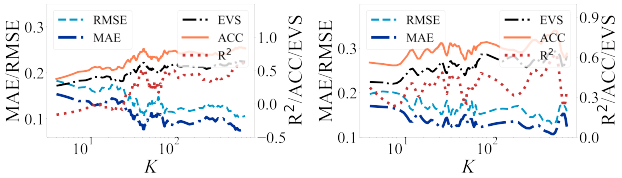
Table 4: Compare WLLE with several widely used manifold embedding and dimension reduction algorithms.

– *Effect of WLLE.* Next, we replace the WLLE in SA-GNN with several popular dimension reduction methods – LLE (Roweis and Saul 2000), MLLE (Zhang and Wang 2007), HLLE (Donoho and Grimes 2003), LTSA (Zhang and Zha 2004), Isomap (Tenenbaum, De Silva, and Langford 2000), t-SNE (Maaten and Hinton 2008), and UMAP (McInnes, Healy, and Melville 2018) – in order to examine the effectiveness of local relation embedding in our WLLE. As shown in Table 4, WLLE achieves the best performance compared to other methods, which proves its effectiveness in preserving relative positions when projecting to 2D space. Among the baselines, t-SNE and UMAP are general dimensional reduction methods without properly reconstruct the local structures, and may not fully preserve relations of surrounding areas. The performance of LLE family models and UMAP are very close, and usually outperform t-SNE that is specifically designed for data visualization. However, all these methods ignore the azimuths and do not distinguish the local distance, which have been explicitly modeled in our WLLE.

We also observe that the superiority of WLLE on the east side is not significant, in comparison to the west side. The reason behind this phenomenon is that the distribution of the monitored points on east side is quite sparse, which neutralizes the effect of relation embedding used in WLLE, as can be verified by the results shown in Table 3, which indicates that the performance of WLLE on east side declines as the relevance between two points decreases.

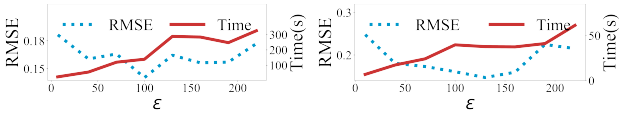
Parameter sensitivity.

– *Influence of* K. One of the most important parameters in our model is K, which specifies the number of nearest neighbors when projecting the manifold onto the 2D space. Figure 3 shows the result of SA-GNN on land deformation prediction with varied K. Clearly, we can see that the model requires larger value of K to obtain the best results on a sparser point cloud, e.g., around $K = 450$ on east side. This result further confirms our hypothesis that WLLE performs better on densely monitored surfaces. Unlike the manifold embedding methods such as LLE and Isomap that focus on preserving local structures, the goal of this work is to predict the land deformation from the point cloud data. Therefore, the value of K required in our model is significantly larger than in manifold embedding methods.



(a) West side. (b) East side.

Figure 3: The influence of K on prediction performance.



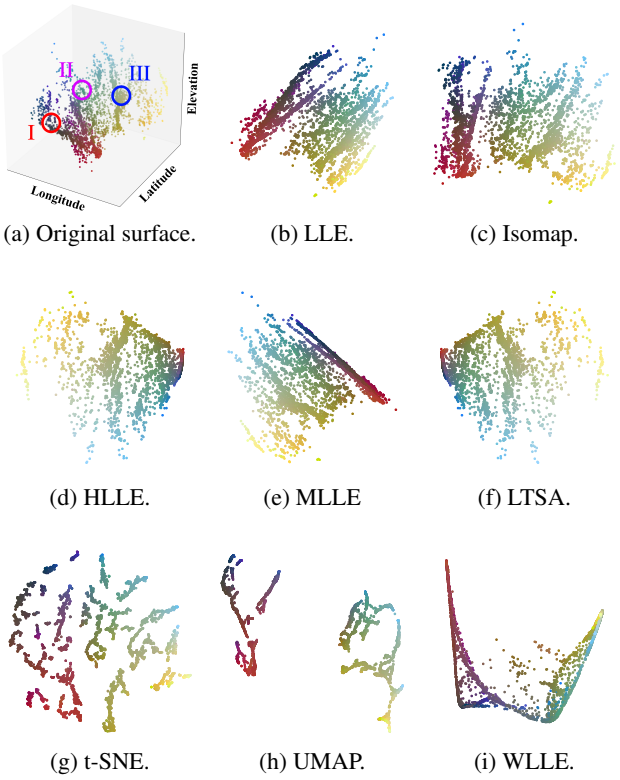
(a) West side. (b) East side.

Figure 4: The influence of ε on computation time and prediction performance.

– *Influence of* ε. Another important parameter of SA-GNN is how to build the connected graph based on the threshold distance ε. Intuitively, larger value of ε would obtain a denser graph, which, however, requires more computation overhead for feature aggregation in GNN. Figure 4 shows the influence of ε on both prediction accuracy and computation cost. We found that appropriate value of ε is required for our model to achieve best performance, e.g., 100m on west side, and 150m on east side. The difference is natural since the point cloud on the east side is sparser, needing more neighbors for deformation estimation. However, further increasing ε would not only incur more computation overhead, but also deteriorate the prediction performance. While the deformation of a point is more affected by surrounding areas, aggregating the deformation features from distanced points would weaken the GNN model. In another words, the closer the neighbor nodes, the higher effect they have on the deformation prediction.

Qualitative analysis. We have the following two kinds:

– *Visualization of node embedding.* Figure 5 plots the projected 2D embeddings of different algorithms for dimension reduction and metric learning. We can see that LLE-based methods and Isomap almost intactly map the manifold into the 2D space. However, they may fail to keep certain useful information that could be critical for land deformation



(a) Original surface. (b) LLE. (c) Isomap.

(d) HLLE. (e) MLLE (f) LTSA.

(g) t-SNE. (h) UMAP. (i) WLLE.

Figure 5: 2D visualization of the learned embeddings.

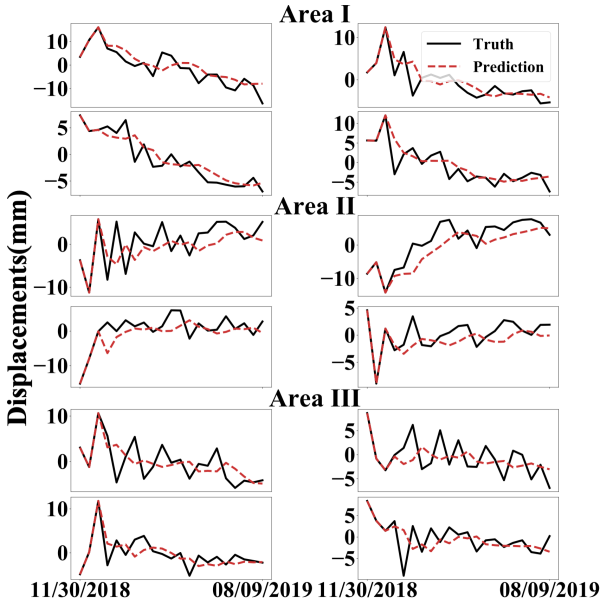


Figure 6: Prediction vs. the ground-truth in the areas marked in Figure 5a.

prediction, such as slope and local relative distance. t-SNE obtains very scattered node embedding, which also explains why it performs poorly on land deformation prediction – relatively little knowledge relevant for deformation can be aggregated from the neighboring areas. WLLE, in contrast, preserves not only the basic surface structures but also the relative positions after projection. An interesting observation is that WLLE can possibly better discriminate the outliers such as the sparsely monitored areas (e.g., the yellow dots), which are difficult to predict due to the lack of the data with respect to surrounding areas.

– *Predicted vs. Real deformation.* To qualitatively investigate the performance of WLLE, we randomly select a few areas (marked by I, II, and III in Figure 5a) and plot their real deformations and our predictions. Figure 6 plot the results. Clearly, the nearby areas have very similar deformation, since adjacent areas usually receive similar stress. This result also verifies the motivation of this work, i.e., learn and aggregate the features of surrounding areas through GNNs for land deformation prediction.

Discussion & Conclusion

We presented SA-GNN, a novel method for predicting land deformations (e.g., landslides) from point cloud data. Our experiments demonstrated that SA-GNN is superior to structure learning and ST-GNN approaches. Incorporating both relative positions and azimuth data renders SA-GNN is more effective than methods that directly build graphs on the point cloud and general manifold learning algorithms for landslides prediction. Intuitively, the improvements depend on the aggregated neighbor information using the graph neural networks, which is also supported by a closer look at the qualitative results.

There are several extensions that we plan for future works. First, SA-GNN can usually successfully predict the trend of the deformation rather than the exact value of the displacement (cf. Figure 6). In other words, the extreme deformation in some monitored areas is usually "smoothed" by the aggregation mechanism in GNN. This, however, is an open problem in the community, happening because the features of the connected nodes in the graph would converge to similar values due to the nature of Laplacian smoothing in graph convolution (Zhao and Akoglu 2020). Next, our metric learning model WLLE may, to an extent, distort the original manifold structures, despite its ability to embed the relative slope information that is desirable for land deformation prediction. This, however, raises the question how to maintain the local manifold structures while preserving the relative azimuth information. Besides, it is possible to dynamically update the embedding information of nodes, since the nodes' displacements are continuously changed. Last but not least, how to extend the current version of WLLE to model richer information (e.g., weather, precipitation, soil type), is a worthwhile challenge.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No.62072077 and No.61602097), and NSF grant CNS 1646107.

References

Bozzano, F.; Cipriani, I.; Mazzanti, P.; and Prestininzi, A. 2011. Displacement patterns of a landslide affected by human activities: insights from ground-based InSAR monitoring. *Natural hazards* 59(3): 1377–1396.

Carlà, T.; Intrieri, E.; Raspini, F.; Bardi, F.; Farina, P.; Ferretti, A.; Colombo, D.; Novali, F.; and Casagli, N. 2019. Perspectives on the prediction of catastrophic slope failures from satellite InSAR. *Scientific reports* 9(1): 1–9.

Chen, W.; Xie, X.; Peng, J.; Wang, J.; Duan, Z.; and Hong, H. 2017. GIS-based landslide susceptibility modelling: a comparative assessment of kernel logistic regression, Naïve-Bayes tree, and alternating decision tree models. *Geomatics, Natural Hazards and Risk* 8(2): 950–973.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555* .

Dai, K.; Li, Z.; Tomás, R.; Liu, G.; Yu, B.; Wang, X.; Cheng, H.; Chen, J.; and Stockamp, J. 2016. Monitoring activity at the Daguangbao mega-landslide (China) using Sentinel-1 TOPS time series interferometry. *Remote Sensing of Environment* 186: 501–513.

Dong, J.; Zhang, L.; Liao, M.; and Gong, J. 2019. Improved correction of seasonal tropospheric delay in InSAR observations for landslide deformation monitoring. *Remote Sensing of Environment* 233: 111370.

Donoho, D. L.; and Grimes, C. 2003. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS* 100(10): 5591–5596.

Gan, B.-R.; Yang, X.-G.; and Zhou, J.-W. 2019. GIS-based remote sensing analysis of the spatial-temporal evolution of landslides in a hydropower reservoir in southwest China. *Geomatics, Natural Hazards and Risk* 10(1): 2291–2312.

Gao, W.; Dai, S.; and Chen, X. 2020. Landslide prediction based on a combination intelligent method using the GM and ENN: two cases of landslides in the Three Gorges Reservoir, China. *Landslides* 17(1): 111–126.

Ghorbanzadeh, O.; Blaschke, T.; Gholamnia, K.; Meena, S. R.; Tiede, D.; and Aryal, J. 2019. Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection. *Remote Sensing* 11(2): 196.

Hajimoradlou, A.; Roberti, G.; and Poole, D. 2020. Predicting Landslides Using Locally Aligned Convolutional Neural Networks. In *IJCAI*, 3342–3348.

Hong, H.; Pradhan, B.; Jebur, M. N.; Bui, D. T.; Xu, C.; and Akgun, A. 2016. Spatial prediction of landslide hazard at the Luxi area (China) using support vector machines. *Environmental Earth Science* 75(1): 40.

Huang, R.; and Fan, X. 2013. The landslide story. *Nature Geoscience* 6(5): 325–326.

Kalantar, B.; Pradhan, B.; Naghibi, S. A.; Motevalli, A.; and Mansor, S. 2018. Assessment of the effects of training data selection on the landslide susceptibility mapping: a comparison between support vector machine (SVM), logistic regression (LR) and artificial neural networks (ANN). *Geomatics, Natural Hazards and Risk* 9(1): 49–69.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Lei, T.; Zhang, Y.; Lv, Z.; Li, S.; Liu, S.; and Nandi, A. K. 2019. Landslide inventory mapping from bitemporal images using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters* 16(6): 982–986.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR* 9(Nov): 2579–2605.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426* .

Roweis, S. T.; and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500): 2323–2326.

Shi, W.; Ragunathan; and Rajkumar. 2020. Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. In *CVPR*.

Shirzadi, A.; Bui, D. T.; Pham, B. T.; Solaimani, K.; Chapi, K.; Kavian, A.; Shahabi, H.; and Revhaug, I. 2017. Shallow landslide susceptibility assessment using a novel hybrid intelligence approach. *Environmental Earth Science* 76(2): 60.

Tenenbaum, J. B.; De Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500): 2319–2323.

Wang, X.; Ma, Y.; Wang, Y.; Jin, W.; Wang, X.; Tang, J.; Jia, C.; and Yu, J. 2020. Traffic Flow Prediction via Spatial Temporal Graph Neural Network. In *WWW*, 1082–1092.

Wang, Y.; Fang, Z.; and Hong, H. 2019. Comparison of convolutional neural networks for landslide susceptibility mapping in Yanshan County, China. *Science of the total environment* 666: 975–993.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *ACM TOG* 38(5): 1–12.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *TNNLS* .

Zhang, Z.; and Wang, J. 2007. MLLE: Modified locally linear embedding using multiple weights. In *NIPS*, 1593–1600.

Zhang, Z.; and Zha, H. 2004. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM journal on scientific computing* 26(1): 313–338.

Zhao, L.; and Akoglu, L. 2020. PairNorm: Tackling Oversmoothing in GNNs. In *ICLR*.