

Land Deformation Prediction via Slope-Aware Graph Neural Networks

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.

1 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 $1.16 \times 10^9 \text{ m}^3$ (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 (Zhou, Lu, and Yang 2017; 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 learn-

ing techniques for landslide prediction includes Bayes networks (Shirzadi et al. 2017), logistic regression (LR) (Kalanter 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, Raghunathan, 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

84 identifying the object shapes without considering the unique
85 features of the terrain surface, e.g., orientations, slope, and
86 relative distances between points. The terrain surface is not
87 stable but gradually and continuously changes due to, for
88 example, sedimentation, erosion, and deposition. Moreover,
89 the displacement of a point may affect its surrounding locations
90 and vice versa, while the influence between points is
91 not fixed but is strongly related to the mutual distance, azimuth,
92 and gradient.

93 In this work, we propose an approach to address the challenges
94 in surface deformation monitoring using InSAR data.
95 Specifically, we present a method to preserve the manifold
96 structures of the surface points based on locally linear em-
97 bedding (LLE) and reconstruct the 3D surface with a neigh-
98 bor graph. Our SA-GNN model considers the intricate spatial
99 dependency between adjacent points and allows each
100 point in a graph to be aware of its nearby terrain deformation
101 by exploiting their inter-dependency among temporal me-
102 surements. Following are our main contributions:

- 103 • To our knowledge, we are among the first to present land-
104 slide monitoring and surface deformation prediction from
105 a graph-based perspective by adapting GNN to InSAR
106 point clouds. We believe this work can foster more en-
107 gagement in applying AI for securing humans, environ-
108 ments, and critical infrastructures.
- 109 • We propose a metric learning method to learn mutual spa-
110 tial dependencies in the point cloud and maintain the mani-
111 fold structure of terrains to handle implicit correlations
112 among local deformations. Our goal is to provide an al-
113 ternative view of learning complicated structures of point
114 clouds beyond the object shapes.
- 115 • We conduct experiments on real-world datasets collected
116 by monitoring activities spanning more than nine months.
117 We show that our method outperforms previous GNN-
118 based methods on predicting the temporal deformation,
119 which demonstrates its inherent capability of dynamically
120 and continuously warning the catastrophic slope failures.

121 2 Related Work

122 **Landslide Mapping & Predicting.** Timely and accurately
123 predicting landslides is a paramount in the geology com-
124 munity. Previous works have focused on generating proper
125 landslide susceptibility mapping (LSM) using optical satel-
126 lite images and aerial photographs due to increased accessi-
127 bility to high-resolution remote sensing data (Bozzano et al.
128 2011; Gao, Dai, and Chen 2020). The aim is to identify geo-
129 logical areas that are prone to catastrophic slope failures,
130 as well as terrain deformations susceptible to landslides. For
131 example, (Shirzadi et al. 2017) leverages naïve Bayes trees
132 for LSM, using random sub-space to generate subsets from
133 training data and then constructing a primary classifier with
134 tree structures. (Chen et al. 2017) trains a SVM with var-
135 ious kernels to calculate the susceptibility indexes for all
136 pixels in the study area. (Lei et al. 2019) employs a CNN
137 with pyramid pooling to learn image features for improving
138 LSM results and landslide recognition, where morpholog-
139 ical reconstruction and clustering are applied to distinguish
140 landslide areas.

Two experimental studies (Wang, Fang, and Hong 2019;
141 Ghorbanzadeh et al. 2019) evaluate CNNs and typical ma-
142 chine learning methods on susceptibility detection and in-
143 vestigate the impact of spectral and topographic factors on
144 LSM, indicating that CNNs are more practical for LSM and
145 landslide prevention than conventional methods. Recently,
146 (Hajimoradlou, Roberti, and Poole 2020) proposed locally
147 aligned CNN filters to capture the orientation of each pixel
148 at multiple resolutions for landslide identification. Previous
149 works either study landslide susceptibility using 2D geospa-
150 tial images (Wang, Fang, and Hong 2019; Ghorbanzadeh
151 et al. 2019; Hajimoradlou, Roberti, and Poole 2020) or con-
152 vert the InSAR data into a 2D bird’s view images and then
153 apply statistical methods for landslide prediction (Dai et al.
154 2016; Carlà et al. 2019). Such approaches ignore critical spa-
155 tial dependencies among monitored areas and induce predic-
156 tion errors due to the limited image resolution and the ab-
157 sence of significant information regarding the land surface
158 (e.g., gradient, azimuth, and mutual influence).

Graph Neural Networks. GNNs are powerful models for
160 learning rich relational information in graphs by aggregat-
161 ing features from adjacent nodes/edges, emerging as *de facto*
162 models for learning graph-structured data in a variety of do-
163 mains such as social networks, molecular/biological topo-
164 logy, and knowledge graphs (Wu et al. 2020). The general
165 GNNs seek to generalize knowledge among nodes/edges
166 and to learn better representations, but fail to capture the
167 node position within the context of the graph structure. Spa-
168 tial and temporal attributes associated with nodes have been
169 exploited for certain applications, e.g., traffic forecasting (Li
170 et al. 2018) and urban flow prediction (Wang et al. 2020).
171 Existing spatio-temporal GNNs (ST-GNN) model road sen-
172 sor networks or urban areas as 2D graphs and cannot be
173 directly applied for point cloud data. Point-GNN (Shi, Ra-
174 gunathan, and Rajkumar 2020), while predicting the cate-
175 gory and shape of objects described by point clouds, aims at
176 object detection through discriminating the bounding box a
177 node belongs to, and cannot handle the intra-nodes relations
178 beyond locations.

Our SA-GNN is specifically designed for landslide pre-
180 diction. Instead of simply converting a point cloud to a reg-
181 ular image or directly distinguishing single vertex, we use a
182 slope-aware locally linear embedding (LLE) module to pre-
183 serve the spatial characteristics of a point cloud. Unlike the
184 ST-GNNs that aggregate node attributes constrained by the
185 2D spatial locations, we embed the slope information and
186 interactive influence among nodes into the graph and con-
187 tinuously monitor and predict the surface deformation by it-
188 eratively updating local-level node representations.

189 3 Preliminaries

Dataset Our data is collected from the slopes on both sides
191 of a large-scale hydropower dam¹. We use InSAR technol-
192 ogy to monitor the surface displacement over time – from
193 Nov 30, 2018 to Sep 8, 2019. During this period of time,
194 there were several slope failures, ranged from 600 to 10,000
195 m³. The most serious landslide occurred in Aug 17, 2019,
196

¹Name and location anonymized for double-blind review.

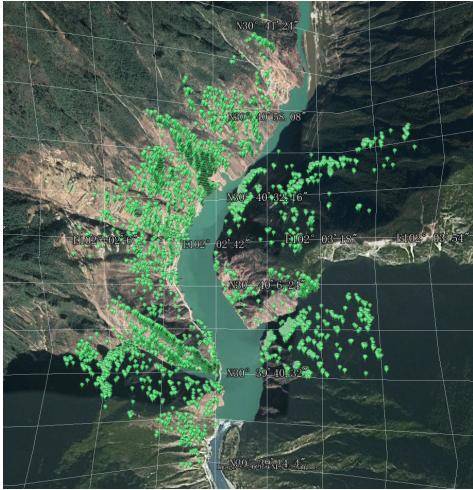


Figure 1: Bird’s-eye view of the studied reservoir areas above a large dam monitored for 9 months.

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.

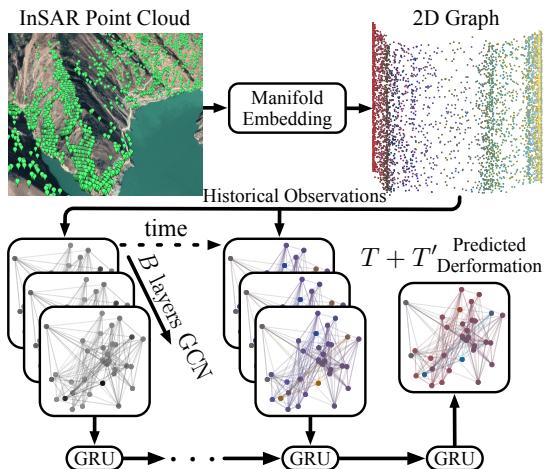


Figure 2: The framework of the proposed SA-GNN.

$$F(\mathbf{M}; \mathcal{G}) = \widehat{\mathbf{M}}' = (\hat{\mathbf{m}}^{T+1}, \dots, \hat{\mathbf{m}}^{T+t'}, \dots, \hat{\mathbf{m}}^{T+T'})$$

4 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.

4.1 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 :

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

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

254 be found by solving a least-square problem using Lagrange
255 multiplier method.

256 LLE and its variants (Donoho and Grimes 2003; Zhang
257 and Zha 2004; Zhang and Wang 2007), discriminate K near-
258 est neighbors via weights w_{ij} but may not be able to distin-
259 guish their relative positions when projecting into a lower
260 dimensional space. In landslide prediction, a node's azimuth
261 can strongly affect the prediction accuracy of surrounding
262 areas – e.g., a landslide (or dramatic deformations) of the
263 monitored areas above a point has a higher probability for
264 causing the displacement of that point. To capture such cor-
265 relations, WLLE accounts for the distance and slope be-
266 tween each node l_i and its neighbors.

267 Specifically, for all K neighbors of l_i , we have $\mathbf{g}_i = \mathbf{g}_j +$
268 $(\mathbf{g}_i - \mathbf{g}_j)$, which means we can use an extra vector $(\mathbf{g}_i - \mathbf{g}_j)$
269 to reconstruct \mathbf{g}_i . Given a point l_i , we measure the relevance
270 between l_i and its neighbors by learning the coefficient w_{ij} :

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

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

276 In addition, we would like to enrich the weight w_{ij} with
277 relative positions between two nodes. Let $d_{ij} = \|\mathbf{g}_i - \mathbf{g}_j\|_2$
278 and s_{ij} be the straight-line distance and slope between two
279 points, respectively. We assume a linear dependency $m'_{ij} =$
280 $\alpha s_{ij} + (1 - \alpha)/d_{ij}$, where α is a weight balancing the two
281 factors and m'_{ij} is normalized by $m_{ij} = m'_{ij}/\sum_j m'_{ij}$ so
282 that $\sum_j m_{ij} = 1$. The basic idea is to upweight the node
283 pairs that are more relevant by compensating w_{ij} with m_{ij} .
284 The main objective of WLLE can now be summarized as:

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

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

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

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

292 for which the solution is presented in Appendix B.

4.2 Graph Convolution and Prediction

293 **Graph Build.** With embedding matrix \mathbf{T} in hand, we calculate
294 the Euclidean distance $dist(\mathbf{t}_i - \mathbf{t}_j)$ between two point
295 vectors in the 2D space, which can be readily used to con-
296 struct the neighbor graph \mathcal{G} by setting a threshold ε , i.e.,
297 there is an edge between point l_i and l_j if $dist(\mathbf{t}_i - \mathbf{t}_j) \leq \varepsilon$.
298 **Spatial Dependency Modeling.** Since the influence of sur-
299 rounding points is critical in slope deformation prediction, it
300 is straightforward to learn such dependencies using GNNs,
301 which generalize convolution operations in the graph by
302 flexibly and aggressively capturing features of neighbors in
303 the non-Euclidean space. Given an adjacency matrix \mathbf{A} and
304 a feature matrix \mathbf{M} , we use GCN (Kipf and Welling 2017)
305 to perform local feature aggregations:
306

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

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

315 **Temporal Dependency Modeling.** To capture the temporal
316 dependency of deformation, we adopt the GRU (Chung et al.
317 2014) to process the sequence information of each node in-
318 dividually, while sharing the parameters of the GRUs with
319 each other for all the nodes.

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

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

323 where L_2 regularization is used to avoid overfitting.

5 Empirical Evaluation

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

326 **Baselines.** We compare our proposed model to the follow-
327 ing baselines for predicting the time series of displacement:
328 (1) *Historical Average (HA)*: models historical data average
329 of a certain time period T to predict the deformation in the
330 next time step. We set $T = 2$. (2) *ARIMA*: is a widely used
331 time-series model that combines auto-regressive and mov-
332 ing average for prediction. (3) *SVR*: is a typical time-series
333 model predicting the value at a future time step with linear
334 support vector regression. (4) *LSTM* and *GRU*: are variants
335 of RNN that captures the information of long-short term de-
336 pendency, and have been widely used for time-series fore-
337 casting. (5) *STGCN*: spatio-temporal GCNs model the time-
338 series with an external graph structure which describes the
339 relationships between nodes' geographical relations, which
340 has been widely used for graph-based time-series prediction
341 such as traffic prediction and action recognition (Wu et al.
342 2020). Here we use GCN to capture the 2D spatial depen-
343 dencies among nodes and use GRU to model the historical

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, R² and EVS.

345 observations and to make predictions. (6) *Point-GNN* (Shi,
346 Ragunathan, and Rajkumar 2020): is a point cloud graph
347 convolution method that is originally proposed for object
348 prediction. It directly constructs a graph based on the Eu-
349 clidean distance between nodes on the point cloud. We adapt
350 it to displacement prediction by using GRU to model the
351 temporal dependencies.

352 **Experimental settings.** Since the time span of the dataset
353 is around 9 months (280 days), we used the first 140 days
354 (~50%) data for training and the remaining for validation
355 (84 days, ~30%) and testing (56 days, ~20%). All deep
356 learning models, including ours, are tuned to the best per-
357 formance with early stopping when validation loss has not
358 declined for 40 consecutive epochs. We use Adam optimizer
359 with an initial learning rate of $3e^{-4}$, which decays with the
360 rate of 0.9 every 100 epochs. Unless otherwise specified, for
361 all manifold learning methods, the parameter K is tuned as
362 450 and 700 for west and east side, respectively. For our
363 WLLE, the default value of α is 0.7 and γ is 100. Lastly, the
364 graphs were built using threshold distance $\varepsilon = 100$ for west
365 side and $\varepsilon = 150$ for east side.

366 **Evaluation protocols.** We use five evaluation methods to
367 measure the prediction performance (defined in detail in Ap-
368 pendix C): root mean squared error (RMSE), mean absolute
369 error (MAE), accuracy (ACC), coefficient of determination
370 (R²), and explained variance score (EVS). RMSE, MAE,
371 and ACC are widely used metrics for evaluating time-series
372 models. R² measures the amount of variation explained by
373 the (least-squares) linear regression. EVS reports the total
374 variance explained by factors that are actually present rather
375 than due to an error variance.

376 **Overall performance comparison.** Table 2 summarizes the
377 performance of different methods on predicting the surface
378 displacement of both sides, from which we have the fol-
379 lowing observations: (1) SA-GNN consistently outperforms
380 all baselines on both land sides, demonstrating its effective-
381 ness of learning surface structures for terrain deformation
382 forecasting. (2) Traditional time-series prediction methods
383 such as HA, SVR, and ARIMA are not comparable due to
384 lack of graph-structure modeling capability, which is also
385 shown with GNN-based methods generally outperforming
386 them solely based on single area monitoring. These results
387 also demonstrate that the deformation of a specific moni-

tored location is strongly affected by surrounding areas. (3)
388 The improvements of SA-GNN over STGCN indicate that
389 simply modeling point cloud data with 2D spatial graph is
390 not enough for surface displacement prediction, largely be-
391 cause it overlooks the complex structures of the terrain sur-
392 face. Similarly, point-GNN, which encodes the point cloud
393 as near-neighbor graph using fixed cut-off distance on the
394 manifold, does not show competitive results, which demon-
395 strated the effectiveness of encoding relative distance and
396 slope between adjacent nodes in SA-GNN. Point-GNN per-
397 forms even poorer than STGCN due to the biases introduced
398 when constructing the graph. In other words, simply build-
399 ing graphs on point cloud does not preserve well the intricate
400 relations and positions of the points. (4) Finally, we found
401 that all models, including our SA-GNN, generally perform
402 better on the west side hill. This phenomenon indicates that
403 the more monitored points, the better the prediction perfor-
404 mance – recall that east slope is sparser in points (cf. Fig-
405 ure 1).

406 **Ablation Study.** We now verify two important motivations
407 of this work, i.e., relative position calculation on the point
408 cloud is biased and graph building requires a careful relative
409 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.

411 – *Effect of relative position embedding.* We first investigate
412 the effect of the relative position embeddings used in SA-
413 GNN. Towards that, we use two alternative metrics to con-
414 struct the graph. In Euclidean and Slope, we respectively use
415 the Euclidean distance and slope between two points to con-
416 struct the graph. Note that we directly calculate the distance
417 and slope on the point cloud without embedding the mani-

fold 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 building 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	R²	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

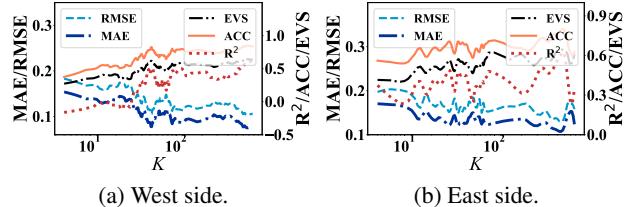


Figure 3: The influence of K on prediction performance.

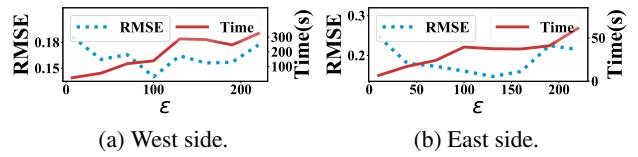


Figure 4: The influence of ε on computation time and prediction performance.

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.

– *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 pro-

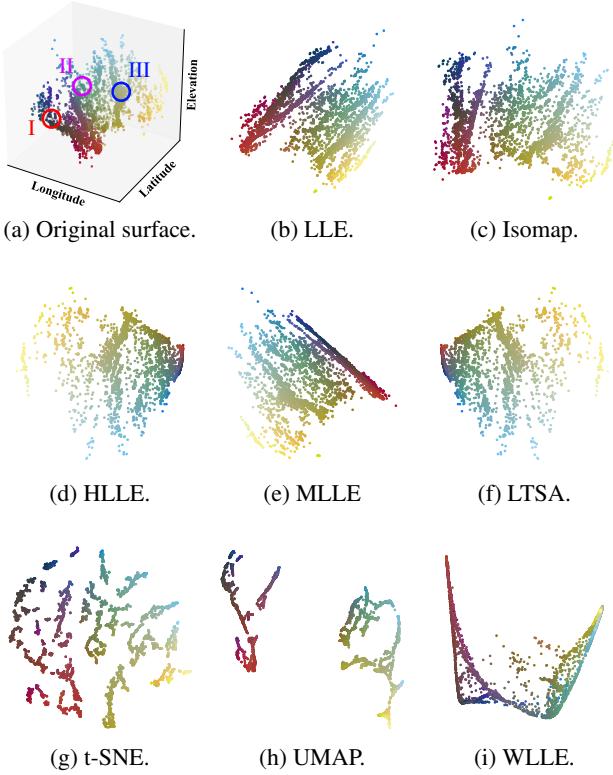


Figure 5: 2D visualization of the learned embeddings.

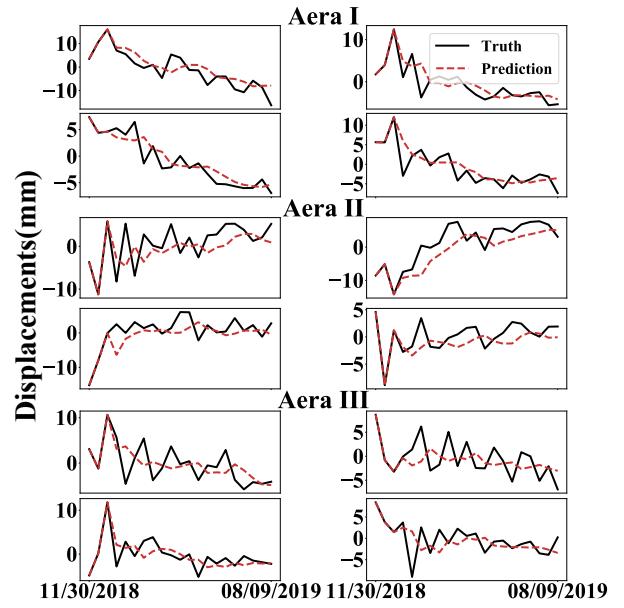


Figure 6: Prediction vs. the ground-truth in the areas marked in Figure 5a.

6 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.

jected 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 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.

515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547

References

- 548
549 Bai, Z.; Demmel, J.; Dongarra, J.; Ruhe, A.; and van der Vorst, H. 2000. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. SIAM.
550
551 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.
552
553 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.
554
555 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.
556
557 Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555* .
558
559 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.
560
561 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.
562
563 Donoho, D. L.; and Grimes, C. 2003. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS* 100(10): 5591–5596.
564
565 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.
566
567 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.
568
569 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.
570
571 Hajimoradlou, A.; Roberti, G.; and Poole, D. 2020. Predicting Landslides Using Locally Aligned Convolutional Neural Networks. In *IJCAI*, 3342–3348.
572
573 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.
574
575 Horn, R. A.; and Johnson, C. R. 2012. *Matrix analysis*. Cambridge university press.
576
577 Huang, R.; and Fan, X. 2013. The landslide story. *Nature Geoscience* 6(5): 325–326.
578
579 Kalantar, B.; Pradhan, B.; Naghibi, S. A.; Motavalli, 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.
580
581 Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
582
583 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.
584
585 Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.
586
587 Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR* 9(Nov): 2579–2605.
588
589 McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426* .
590
591 Roweis, S. T.; and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500): 2323–2326.
592
593 Shi, W.; Ragunathan; and Rajkumar. 2020. Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. In *CVPR*.
594
595 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.
596
597 Tenenbaum, J. B.; De Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500): 2319–2323.
598
599 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.
600
601 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.
602
603 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.
604
605 Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *TNNLS* .
606
607 Zhang, Z.; and Wang, J. 2007. MLLE: Modified locally linear embedding using multiple weights. In *NIPS*, 1593–1600.
608
609 Zhang, Z.; and Zha, H. 2004. Principal manifolds and non-linear dimensionality reduction via tangent space alignment. *SIAM journal on scientific computing* 26(1): 313–338.
610
611

- 656 Zhao, L.; and Akoglu, L. 2020. PairNorm: Tackling Over-
657 smoothing in GNNs. In *ICLR*.
- 658 Zhou, J.-w.; Lu, P.-y.; and Yang, Y.-c. 2017. Reservoir land-
659 slides and its hazard effects for the hydropower station: a
660 case study. In *World Landslide Forum*, 699–706.

661 **Ethical Impact**

662 The methods and findings described in this paper can be har-
663 nessed to improve the effectiveness of landslide prediction,
664 thereby helping to prevent secondary disasters and plan ef-
665 fective evacuations and post-remediations. Especially so for
666 geo-settings of high mountains and inaccessible areas that
667 are difficult for geological workers to reach and deploy other
668 sensing/monitoring equipment. In such settings, our work
669 can be potentially used to generate warnings that, from soci-
670 etal perspective, could help save property and people’s lives.
671 However, limited by the satellite resolution, methodologies
672 based on InSAR data sources are mostly suitable for larger
673 target-areas. While we worked under a full conviction that
674 our findings are beneficial for the society and made our best
675 effort to propose sound novel approaches, we are ethically
676 bound to make an explicit note that, whenever possible, our
677 SA-GNN system should be used as an “early warning”, in
678 conjunction with other domain expertise and tools.

699

Appendix

A Reconstruction Weight Calculation

681 By multiplying K , the main components of Eq.(3) can be
682 reformulated as:

$$\begin{aligned} & \|K\mathbf{g}_i - \sum_j (\mathbf{g}_j + (w_{ij} + m_{ij})(\mathbf{g}_i - \mathbf{g}_j))\|_2^2 \\ &= \|K\mathbf{g}_i - \sum_j \mathbf{g}_j - \sum_j w_{ij}(\mathbf{g}_i - \mathbf{g}_j) - \sum_j m_{ij}(\mathbf{g}_i - \mathbf{g}_j)\|_2^2 \\ &= \|(K-1)\mathbf{g}_i - \sum_j \mathbf{g}_j - \sum_j m_{ij}(\mathbf{g}_i - \mathbf{g}_j) + \sum_j w_{ij}\mathbf{g}_j\|_2^2 \\ &= \|\mathbf{p}_i + \sum_j w_{ij}\mathbf{g}_j\|_2^2 \\ &= \|\sum_j w_{ij}\mathbf{p}_i + \sum_j w_{ij}\mathbf{g}_j\|_2^2 \\ &= \|\sum_j w_{ij}(\mathbf{p}_i + \mathbf{g}_j)\|_2^2, \end{aligned} \quad (8)$$

683 where $\mathbf{p}_i = (K-1)\mathbf{g}_i - \sum_j \mathbf{g}_j - \sum_j m_{ij}(\mathbf{g}_i - \mathbf{g}_j)$ can
684 be numerically computed. For simplicity, we denote $\mathbf{C}_{js} =$
685 $(\mathbf{p}_i + \mathbf{g}_j) \cdot (\mathbf{p}_i + \mathbf{g}_s)$. Then, above equation is rewritten as:

$$\|\sum_j w_{ij}(\mathbf{p}_i + \mathbf{g}_j)\|_2^2 = \sum_{js} w_{ij}w_{is}\mathbf{C}_{js}, \quad (9)$$

686 which can be minimized via Lagrange multiplier method by
687 enforcing $\sum_j w_{ij} = 1$:

$$\mathcal{L} = \sum_{js} w_{ij}w_{is}\mathbf{C}_{js} + \gamma\|\mathbf{w}_i\|_2 + \lambda(\sum_j w_{ij} - 1), \quad (10)$$

688 Through solving $\partial\mathcal{L}/\partial w_{ij} = 0$, we can obtain the closed
689 form of the optimal weights:

$$(\sum_s \mathbf{C}_{js} + \gamma)w'_{ij} = \frac{-\lambda}{2}, \quad (11)$$

$$w_{ij} = \frac{w'_{ij}}{\sum_j w'_{ij}}. \quad (12)$$

B Computing Embedding Matrix

690 The analytic form of Eq.(5) can be derived as:

$$\begin{aligned} & \sum_i \|K\mathbf{t}_i - \sum_j (\mathbf{t}_j + h_{ij}(\mathbf{t}_i - \mathbf{t}_j))\|_2^2 \\ &= \sum_i \|(K-2)\mathbf{t}_i - \sum_j \mathbf{t}_j + \sum_j h_{ij}\mathbf{t}_j\|_2^2 \\ &= \sum_i \|(K-2)(\mathbf{t}_i - \sum_j \frac{1-h_{ij}}{K-2}\mathbf{t}_j)\|_2^2, \end{aligned} \quad (13)$$

692 which equals to minimize following

$$\Phi(\mathbf{T}) = \sum_i \|\mathbf{t}_i - \sum_j h'_{ij}\mathbf{t}_j\|_2^2. \quad (14)$$

where $h'_{ij} = \frac{1-h_{ij}}{K-2}$, and omit the constant coefficient $K-2$.
The cost essentially defines a quadratic form:
693
694

$$\Phi(\mathbf{T}) = \sum_{ij} \mathbf{Q}_{ij}(\mathbf{t}_i \mathbf{t}_j), \quad (15)$$

which involves inner products of the embedding vectors and
the $N \times N$ matrix \mathbf{Q} :
695
696

$$\mathbf{Q}_{ij} = \delta_{ij} - h'_{ij} - h'_{ji} + \sum_s h'_{si}h'_{sj}, \quad (16)$$

where δ_{ij} is 1 if $i = j$ and 0 otherwise. The optimal embedding
is found by computing the bottom $d+1$ (in our case,
 $d=2$) eigenvectors of the matrix \mathbf{Q} .
697
698
699

We can calculate \mathbf{Q} following (Bai et al. 2000) without
700 performing a full matrix diagonalization:
701

$$\begin{aligned} \Phi(\mathbf{T}) &= \sum_i \|\mathbf{t}_i - \sum_j h'_{ij}\mathbf{t}_j\|_2^2 \\ &= \sum_i \|\mathbf{T}\mathbf{I}_i - \mathbf{T}\mathbf{h}'_i\|_2^2 \\ &= \text{Tr}(\mathbf{T}(\mathbf{I} - \mathbf{H}'^\top)(\mathbf{I} - \mathbf{H}'^\top)^\top \mathbf{T}^\top) \\ &= \text{Tr}(\mathbf{T}\mathbf{Q}\mathbf{T}^\top), \end{aligned} \quad (17)$$

where $\mathbf{Q} = (\mathbf{I} - \mathbf{H}'^\top)(\mathbf{I} - \mathbf{H}'^\top)^\top$, and \mathbf{I} is the identity matrix.
702
The optimal embedding is found by computing the eigen-
703 vectors corresponding to the the smallest $d+1$ eigenvalues
704 of the matrix \mathbf{Q} , which is well defined in the Rayleitz-Ritz
705 theorem (Horn and Johnson 2012). And we discard the bot-
706 tom eigenvector because it is usually the unit vector with all
707 equal components. The eigenvectors of shape $N \times d$ form the
708 embedding vectors $\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N\}$ in d -dimensional
709 space.
710

C Evaluation Protocols

711 We use following metrics to measure the prediction perfor-
712 mance:
713

1. Root Mean Squared Error (RMSE):
714

$$RMSE = \sqrt{\frac{1}{T'N} \sum_{j=1}^{T'} \sum_{i=1}^N (\mathbf{m}_i^{T+j} - \hat{\mathbf{m}}_i^{T+j})^2}$$

2. Mean Absolute Error (MAE):
715

$$MAE = \frac{1}{T'N} \sum_{j=1}^{T'} \sum_{i=1}^N |\mathbf{m}_i^{T+j} - \hat{\mathbf{m}}_i^{T+j}|$$

3. Accuracy:
716

$$Accuracy = 1 - \frac{\|\mathbf{M} - \hat{\mathbf{M}}'\|_F}{\|\mathbf{M}\|_F}$$

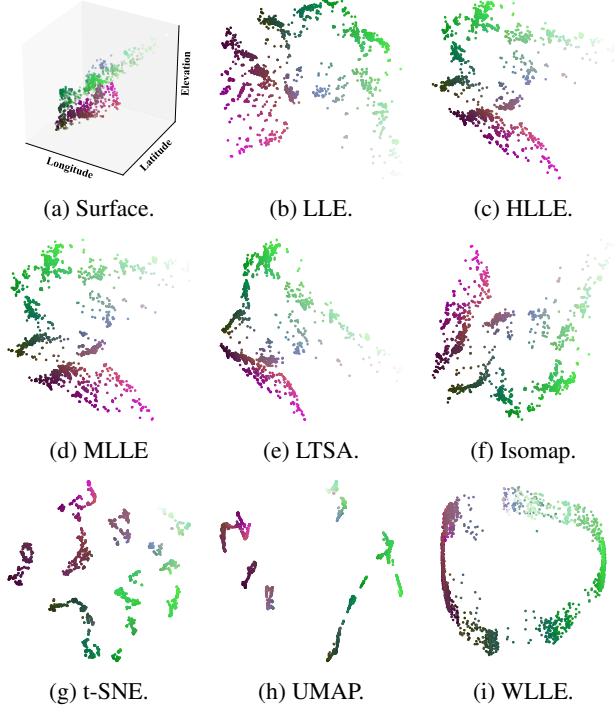


Figure 7: The embedded space of all methods on east side.

717 4. Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{j=1}^{T'} \sum_{i=1}^N \left(\mathbf{m}_i^{T+j} - \hat{\mathbf{m}}_i^{T+j} \right)^2}{\sum_{j=1}^{T'} \sum_{i=1}^N \left(\mathbf{m}_i^{T+j} - \bar{\mathbf{M}} \right)^2}$$

718 5. Explained Variance Score (var):

$$var = 1 - \frac{var\{\mathbf{M} - \hat{\mathbf{M}}'\}}{Var\{\mathbf{M}\}}$$

719 where \mathbf{m}_i^{T+j} and $\hat{\mathbf{m}}_i^{T+j}$ represent the true and predicted
 720 measurement at time $T + j$. The range of predictions is from
 721 T to $T + T'$ on all N nodes. \mathbf{M} , $\hat{\mathbf{M}}$ and $\bar{\mathbf{M}}$ are the real de-
 722 formations, predicted deformations and the average de-
 723 formations, respectively.

724 D Additional Visualizations

725 Figure 7 shows the embedding results of all methods on
 726 the east side. Similar to the results on west side, manifold
 727 learning methods (e.g., LLE, HLLE, IsoMap) are very close
 728 on the final node embeddings, while dimension reduction
 729 methods t-SNE and UMAP obtain intra-clustered but inter-
 730 distanced embeddings. Our WLLE, which is specifically de-
 731 signed for learning relative positions, preserves both the dis-
 732 tance and the slope information in the latent space.