

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/359362034>

# Spatio-Temporal Graph Convolutional Networks for Earthquake Source Characterization

Preprint · March 2022

DOI: 10.1002/essoar.10510873.1

---

CITATION

1

READS

125

5 authors, including:



Xitong Zhang

Michigan State University

26 PUBLICATIONS 126 CITATIONS

[SEE PROFILE](#)



Miao Zhang

Dalhousie University

59 PUBLICATIONS 1,140 CITATIONS

[SEE PROFILE](#)



# JGR Solid Earth

## RESEARCH ARTICLE

10.1029/2022JB024401

Xitong Zhang and Will Reichard-Flynn contributed equally to this work.

### Key Points:

- We design a multistation-based graph neural network for earthquake source characterization
- Both geographic distance and waveform feature similarity are considered in a graph convolutional neural network for feature combination
- Our network is capable of automatically selecting and combining relevant seismic stations to characterize earthquake source parameters

### Supporting Information:

Supporting Information may be found in the online version of this article.

### Correspondence to:

Y. Lin,  
[ylin@lanl.gov](mailto:ylin@lanl.gov)

### Citation:

Zhang, X., Reichard-Flynn, W., Zhang, M., Hirn, M., & Lin, Y. (2022). Spatiotemporal graph convolutional networks for earthquake source characterization. *Journal of Geophysical Research: Solid Earth*, 127, e2022JB024401. <https://doi.org/10.1029/2022JB024401>

Received 17 MAR 2022

Accepted 23 OCT 2022

## Spatiotemporal Graph Convolutional Networks for Earthquake Source Characterization

Xitong Zhang<sup>1,2</sup>, Will Reichard-Flynn<sup>1</sup>, Miao Zhang<sup>3</sup> , Matthew Hirn<sup>2,4,5</sup>, and Youzuo Lin<sup>1</sup> 

<sup>1</sup>Geophysics Group, Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, USA,

<sup>2</sup>Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI, USA,

<sup>3</sup>Department of Earth and Environmental Sciences, Dalhousie University, Halifax, NS, Canada, <sup>4</sup>Department of Mathematics, Michigan State University, East Lansing, MI, USA, <sup>5</sup>Center for Quantum Computing, Science and Engineering, Michigan State University, East Lansing, MI, USA

**Abstract** Accurate earthquake location and magnitude estimation play critical roles in seismology. Recent deep learning frameworks have produced encouraging results on various seismological tasks (e.g., earthquake detection, phase picking, seismic classification, and earthquake early warning). Many existing machine learning earthquake location methods utilize waveform information from a single station. However, multiple stations contain more complete information for earthquake source characterization. Inspired by recent successes in applying graph neural networks (GNNs) in graph-structured data, we develop a Spatiotemporal Graph Neural Network (STGNN) for estimating earthquake locations and magnitudes. Our graph neural network leverages geographical and waveform information from multiple stations to construct graphs automatically and dynamically by adaptive message passing based on graphs' edges. Using a recent graph neural network and a fully convolutional neural network as baselines, we apply STGNN to earthquakes recorded by the Southern California Seismic Network from 2000 to 2019 and earthquakes collected in Oklahoma from 2014 to 2015. STGNN yields more accurate earthquake locations than those obtained by the baseline models and performs comparably in terms of depth and magnitude prediction, though the ability to predict depth and magnitude remains weak for all tested models. Our work demonstrates the potential of using GNNs and multiple stations for better automatic estimation of earthquake epicenters.

**Plain Language Summary** The location and magnitude of an earthquake can be best determined by relating the motion recorded at multiple stations in a network, and it would be beneficial to combine the waveforms from multiple seismic stations for source characterization. Machine learning-based approaches have recently become prevalent in seismological tasks such as earthquake source characterization. Because of the irregular spatial distribution of seismic stations, graph convolutional neural networks (a deep learning architecture which handles graph-structured data) have great potential in combining both spatial and temporal information from different seismic stations. In this work, waveforms recorded at multiple stations are passed through neural networks with connective links based on the similarity of waveform features and geographic locations. The model is tested on two data sets and compared to two published baselines (a graph convolutional neural network and a fully convolutional network). Compared with the baselines, STGNN achieves improved accuracy for epicenter estimation and comparable accuracy for depth and magnitude estimation.

## 1. Introduction

Earthquake source characterization plays a fundamental role in various seismic studies, including earthquake early-warning, hazard assessment, subsurface energy exploration, etc. (Li et al., 2020). Characterization of an earthquake source can be posed as a classical inverse problem. Its purpose is to infer the source information (location, magnitude, etc.) from seismic recordings. Various approaches have been developed to characterize earthquake sources, the most well-established being travel time-based inversion (Li & van der Baan, 2016; Lin et al., 2015; Zhang & Thurber, 2003; Zhang et al., 2017) and waveform-based inversion (Beskardes et al., 2018; Gajewski et al., 2007; Pesicek et al., 2014; Zhebel & Eisner, 2015). Travel time-based methods implement a multi-step process, in which the arrival times of P and S waves are determined through phase detection and associated to specific earthquakes; earthquake locations are estimated as an inversion process given arrival times, station locations, and a velocity model. Magnitudes are calculated based on waveform amplitudes and source-receiver distances. Though travel time-based methods are commonly used in seismic applications, they are susceptible to

© 2022. The Authors.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

noise-related errors, particularly when estimating low-magnitude events, and fail to utilize abundant phase and amplitude information in the complete waveform. In contrast, waveform-based inversion integrates all phase and amplitude information recorded in seismographs, resulting in high quality source characterization. However, waveform-based inversion is computationally expensive. Both methods require domain expertise to properly tune parameters in the inversion process. Deep learning for source characterization provides a data-driven alternative, where integrated location and magnitude predictions extract full-waveform features with less computational expense than waveform inversion.

Advances in algorithms and computing, and the availability of large, high-quality data sets have allowed machine learning techniques to attain spectacular success in seismological applications (Bergen et al., 2019; Kong et al., 2019) including phase picking (Zhu & Beroza, 2019), seismic discrimination (Li et al., 2018), waveform denoising (Zhu et al., 2019), phase association (Ross et al., 2019), earthquake location (Perol et al., 2018), and magnitude estimation (Mousavi & Beroza, 2020b). Although machine learning has long been applied to seismic event detection (Wang & Teng, 1995; Tiira, 1999), the first work to leverage recent advances in deep learning was developed by Perol et al. (2018), where convolutional neural networks (CNNs) were trained to detect earthquakes from single station recordings and predict the source locations from among six regions. Though successful in establishing foundational research in machine learning for earthquake location, the CNN model is restricted to waveforms from a single seismic station and can only classify earthquakes into broad geographic groups without providing specific location information. Since then, more advanced single-station approaches have been developed to improve location accuracy. Mousavi and Beroza (2020a) build Bayesian neural networks to learn epicenter distance, P-wave travel time, and associated uncertainty from single-station data.

Recently, multistation-based machine learning methods have shown promising results. For instance, Kriegerowski et al. (2019) develop a CNN structure that combines three-component waveforms from multiple stations to predict hypocenter locations, resulting in more accurate source parameters than single station methods. Zhang et al. (2020) developed an end to end fully convolutional network (FCN) to predict the probability distribution of earthquake location directly from input data recorded at multiple stations, which was extended to determine earthquake locations and magnitudes from continuous waveforms for earthquake early warning (Zhang et al., 2021). Shen and Shen (2021) also adopt a CNN framework, extracting the location, magnitude, and origin time from continuous waveforms collected across a seismic network.

Though multiple-station approaches improve upon single-station methods, the use of standard convolutional layers is limited in several ways: (a) CNNs are designed to function on evenly spaced grids (i.e., photographs) where information is exclusively shared between adjacent cells and (b) CNNs require the input of station locations to be static (i.e., recordings from station N must always be found at position N of the input file) in order to learn positional mapping. These assumptions are inappropriate for seismic networks, which are not regularly spaced and may record information related to nonadjacent stations. Additionally, station outages, the addition/removal of stations to seismic networks, and the ability to select a localized array for the detection of small-magnitude events makes dynamic station input highly desirable for source characterization.

To solve this problem, recently several graph-based machine learning methods have been developed. Münchmeyer et al. (2020) developed an attention-based transformer model for earthquake early warning, which was extended to predict hypocenters and magnitudes of events in Münchmeyer et al. (2021). While this model is successful in implementing a multistation approach that allows for dynamic inputs, high computational complexity restricts inputs to a relatively small number of stations. Another method for implementing flexible, multistation input that avoids high complexity for large networks is through graph convolution. This method is implemented by van den Ende and Ampuero (2020), who develop a multistation source characterization model. This model regards features as nodes on an edgeless graph, implementing single-station convolution and global pooling. However, global pooling may not sufficiently extract all useful information from multiple seismic stations, as the pooling layer is ideally applied after global features are obtained by feature fusion along the spatial dimension. Yano et al. (2021) introduce a multistation technique in which edges are selected and held fixed for all inputs. While this model allows for more meaningful features to be constructed than in global pooling, station inputs are required to be fixed during training and implementation, introducing the same limitation inherent to CNNs. McBrearty and Beroza (2022) propose a GNN framework using multiple predefined graphs constructed on both labels and station locations. The model allows for variation in the set of input stations, but the inputs are waveform amplitudes and phase arrival times rather than whole waveforms.

To harness the full functionality of graph neural networks (GNNs) while maintaining flexibility in the location and number of seismic stations, we design a data-driven framework, spatiotemporal graph neural network (STGNN), that creates edges automatically to combine waveform features and spatial information. In order to evaluate the performance of our approach, we compare STGNN to two baselines: the GNN model designed by van den Ende and Ampuero (2020) and the FCN designed by Zhang et al. (2020). We apply all three models to the two data sets upon which the baselines were originally tested and trained: (a) regional  $2.5 < M < 6$  earthquakes recorded by 185 seismic stations in Southern California from 2000 to 2019 and (b) local  $0 < M < 4$  earthquakes recorded by 30 seismic stations in Oklahoma from 2014 to 2015.

The layout of this article is as follows. In Section 2, we describe the fundamentals of graph-based CNN models and STGNN. In Section 3, we introduce the field data, training procedures, and experimental results. In Section 4, we discuss the mechanisms which enhance and inhibit the performance of STGNN in the context of previous work. Finally, in Section 5, we present concluding remarks and discuss future work.

## 2. Methodology

### 2.1. Overview

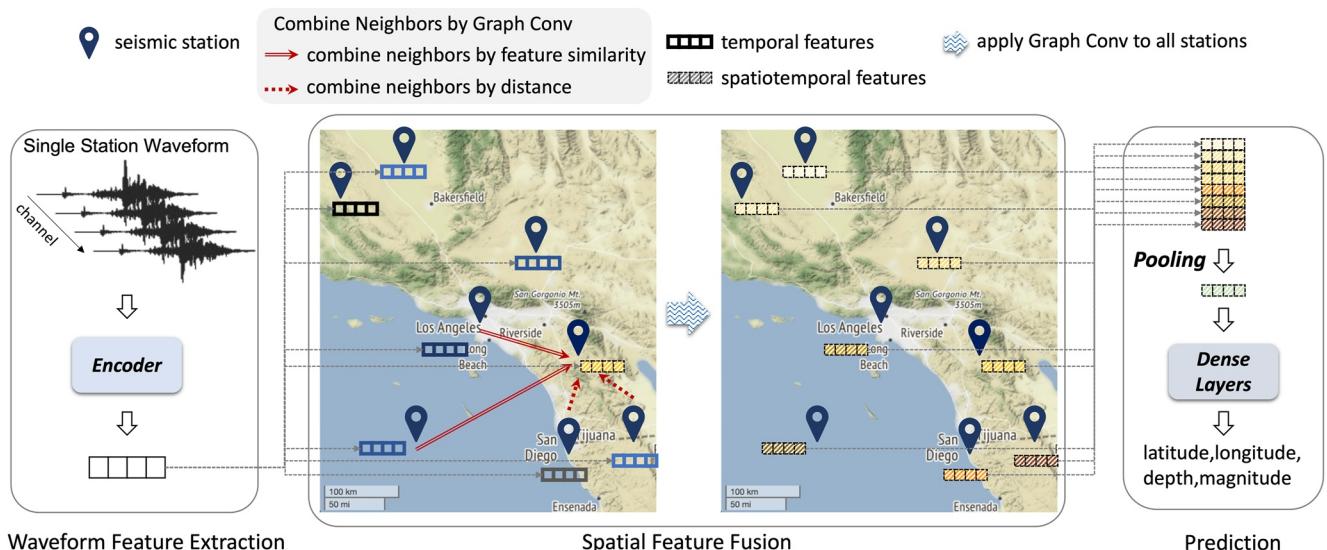
GNNs are designed to handle graphical data, or data that can be represented by vertices connected by edges. In GNNs, convolution and pooling operate along connecting edges. In CNNs, on the other hand, convolution and pooling operate on regions closest together on a Euclidean grid, meaning that input order directly impacts information-sharing and featurization. This is not the case for GNNs, in which edges are not restricted to Euclidean grids but may instead be constructed by any criteria. Two major advantages of GNN architectures are that they do not require a fixed input order and can handle graphs with different sets of vertices. These properties of GNNs are well-suited for seismic data analysis with inputs from multiple stations. It is common for stations in a seismic network to be added, removed, or repositioned, or for the recording quality of individual stations to fluctuate over time due to operation and/or equipment issues. It is therefore beneficial to dynamically select relevant seismic stations for source characterization. We therefore propose a dynamic GNN framework as the basis for STGNN, in which seismic stations act as nodes, connected by dynamically defined edges.

Inspired by Wang et al. (2019), our graph convolutions follow the design of EdgeConv layers to automatically generate edges between nodes. Instead of manually constructing fixed edges or implementing an edgeless graph, our framework learns to combine useful information from multiple stations implicitly during the training process. Our framework consists of three major components as shown in Figure 1:

1. Waveform feature extraction: We first extract temporal features from the waveform recorded at each seismic station using a CNN-based encoder. The three-channel seismic recordings are reduced to a low dimensional representation.
2. Spatial feature fusion: We then represent the seismic station network as a graph, in which each node (i.e., station) is connected to other nodes by automatically generated edges. Through iterative steps of edge generation and convolution, the perceptive field is gradually enlarged. The model integrates and fuses features from different stations to obtain a high-order spatiotemporal representation of the recorded wavefield over the seismic network. The graph convolutional architecture considers both geographic locations and waveform feature similarity among multiple seismic stations.
3. Prediction: The last component is the prediction module. A fully connected neural network outputs four normalized scalars corresponding to latitude, longitude, depth, and magnitude based on features learned from the previous steps.

### 2.2. Graph Convolutional Layers

The spatial feature fusion process consists of multiple graph convolution layers. The goal of each graph convolution layer is to enlarge the perceptive field by combining the extracted feature of each seismic stations and auto-selected neighbor stations. Each graph convolution layer can be broken down into two steps: edge generation and feature update.



**Figure 1.** The overview of Spatiotemporal Graph Neural Network (STGNN). There are three major components in STGNN: (1) Waveform feature extraction for obtaining time domain feature from each station independently. (2) Spatial feature fusion for time domain feature integration from different stations based on their geographic locations and extracted feature similarity. (3) Earthquake location and magnitude prediction given spatiotemporal features from the previous step. See Figure 2 for a more detailed summary of the network architecture.

**Edge generation.** Each station node is connected to several other station nodes which show maximum similarity to the node. Similarity measurements are based on two criteria:

1. *Geographic distance*: The geographic distance is the intuitive choice, since adjacent stations tend to record related signals due to similar wave paths. Additionally, events are more likely to be mutually recorded by stations in close proximity, especially in the case of small-magnitude events.
2. *Feature similarity*: As the same earthquake event can be recorded by distant stations in a large area, waveform similarity provides a complimentary perspective to geographic distance. We compare  $l_2$  distance of features from stations  $i$  and  $j$  directly by  $\|x_i - x_j\|^2$ , and thus we can combine waveform features from distant stations, where  $x_i$  and  $x_j$  are the extracted feature vectors.

In edge generation, we link every station with its K-nearest neighbors based on their similarity, where K is a tunable hyperparameter. In our framework, both geographic proximity and waveform feature similarity are considered.

By ignoring the feature channel and batch dimensions, we assume the feature for neighbors selection is  $X \in \mathbb{R}^{N \times d}$ , where  $N$  is the number of stations and  $d$  is the feature dimension.  $d$  equals to 2 when the criterion of generating edges is geographic distance (longitude and latitude). Let  $x_i = X(i)$  and  $x_j = X(j)$ , the process of selecting K-nearest neighbors can be explained with the following equations:

1. Compute the pair-wise distance matrix  $D \in \mathbb{R}^{N \times N}$ :

$$D(i, j) = \|x_i - x_j\|^2, 1 \leq i \leq N, 1 \leq j \leq N \quad (1)$$

2. Get K-nearest neighbors  $\mathcal{N} \in \mathbb{R}^{N \times K}$  by sorting each row of  $D$ :

$$\mathcal{N}(i) = \text{smallest-K}(D(i)), 1 \leq i \leq N \quad (2)$$

In practice, the similarity between waveforms can also be affected by other factors, such as wave path and signal to noise ratio. By training with a large amount of samples with different sets of seismic stations with distinct spatial distributions, the network will learn to embed these implicit and complex factors to low dimensional features automatically in order to minimize the misfit between labels and predictions.

**Feature update.** Given the edges, we update the features of each stations by

$$\tilde{x}_i = \maxpool_{j \in \mathcal{N}_{\text{distance}}(i)} (g(x_i - x_j) + f(x_i)) + \maxpool_{j' \in \mathcal{N}_{\text{similarity}}(i)} (g'(x_i - x_{j'}) + f'(x_i)), \quad (3)$$

where  $x_i$ ,  $x_j$ , and  $x_{j'}$  are features of station  $i$ ,  $j$ , and  $j'$ , respectively.  $j$  is a neighbor of  $i$  based on geographic distance and  $j'$  is a neighbor of  $i$  by measuring feature similarity from the previous edge generation step.  $g(\cdot)$ ,  $f(\cdot)$ ,  $g'(\cdot)$ , and  $f'(\cdot)$  are trainable fully connected neural networks.  $\tilde{x}_i$  is the updated feature of station  $i$ . Max pooling is conducted along the constructed edges to combine information from the K-nearest neighbors of  $i$ , so that each station is once more associated with a single feature vector. The update is asymmetric for stations  $i$  and  $j$  to encourage the update processes of  $i$  and  $j$  to be different, as it is possible that only one of the stations records the event.

### 2.3. Architecture

The model takes as input (a) a list of station coordinates and (b) waveforms recorded by each station. Each input can contain an arbitrary set of stations, limited by a trainable maximum. If the number of functioning stations is less than the maximum number of stations for which the model is trained, the input is padded with zeroed channels and the coordinates of the missing stations are set to  $(-1, -1)$ .

A graphical illustration of the architecture is presented in Figure 2. In Temporal Feature Extraction, time domain waveform features are extracted from each station independently using an encoder with 11 convolutional layers. These features are used to construct the initial inputs for Spatial Feature Fusion.

Two graphs are generated for each layer of graph convolution: one in which edges are generated based on geographic distance and one in which edges are generated based on waveform feature similarity. For graphs in which geographic distance dictates edges, two scalars containing station coordinates, normalized between  $-1$  and  $1$ , are concatenated to the station's feature vector for neighbor selection and graph convolution. Spatial Feature Fusion uses four layers of graph convolutional layers to obtain spatially hierarchical features.

The features from all four graph convolutions are concatenated together along the feature dimension for final source characterization regression. A fully connected layer transforms the features in each station, and the features are then compressed with adaptive max pooling along the station dimension. The compressed features are regressed to scalar predictions of latitude, longitude, depth, and magnitude using a fully connected neural network. The objective function is

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \frac{1}{4} \|y_i - \hat{y}_i\|, \quad (4)$$

where  $\hat{y}_i$  and  $y_i$  are the prediction and ground truth values of  $i$ th sample, respectively, represented as vectors of latitude, longitude, depth, and magnitude.

The maximum number of stations and number of edges are architectural parameters set during training. The model design in PyTorch allows retraining to new architectural parameters without fundamental alteration of the code.

## 3. Experiments and Results

In this section, the data, experiment settings, and results are discussed. We evaluate STGNN in two ways: (a) performance on two data sets compared to GNN and CNN baselines and (b) stability analysis of STGNN with various settings.

### 3.1. Data Description

The Southern California data set uses waveforms and catalog information collected by the Southern California Earthquake Data Center (SCEDC) (Hutton et al., 2010) and was used for training and testing in the GNN baseline (van den Ende & Ampuero, 2020). The selected data set contains events from January 2000 to June 2019 within a geographic subset from  $32^\circ$  to  $36^\circ$  latitude and  $-120^\circ$  to  $-116^\circ$  longitude, a depth range of  $0$ – $30$  km, and a magnitude range of  $2.5 < M < 6$ . The final data set contains 2,209 events recorded by 185 broadband seismic stations.

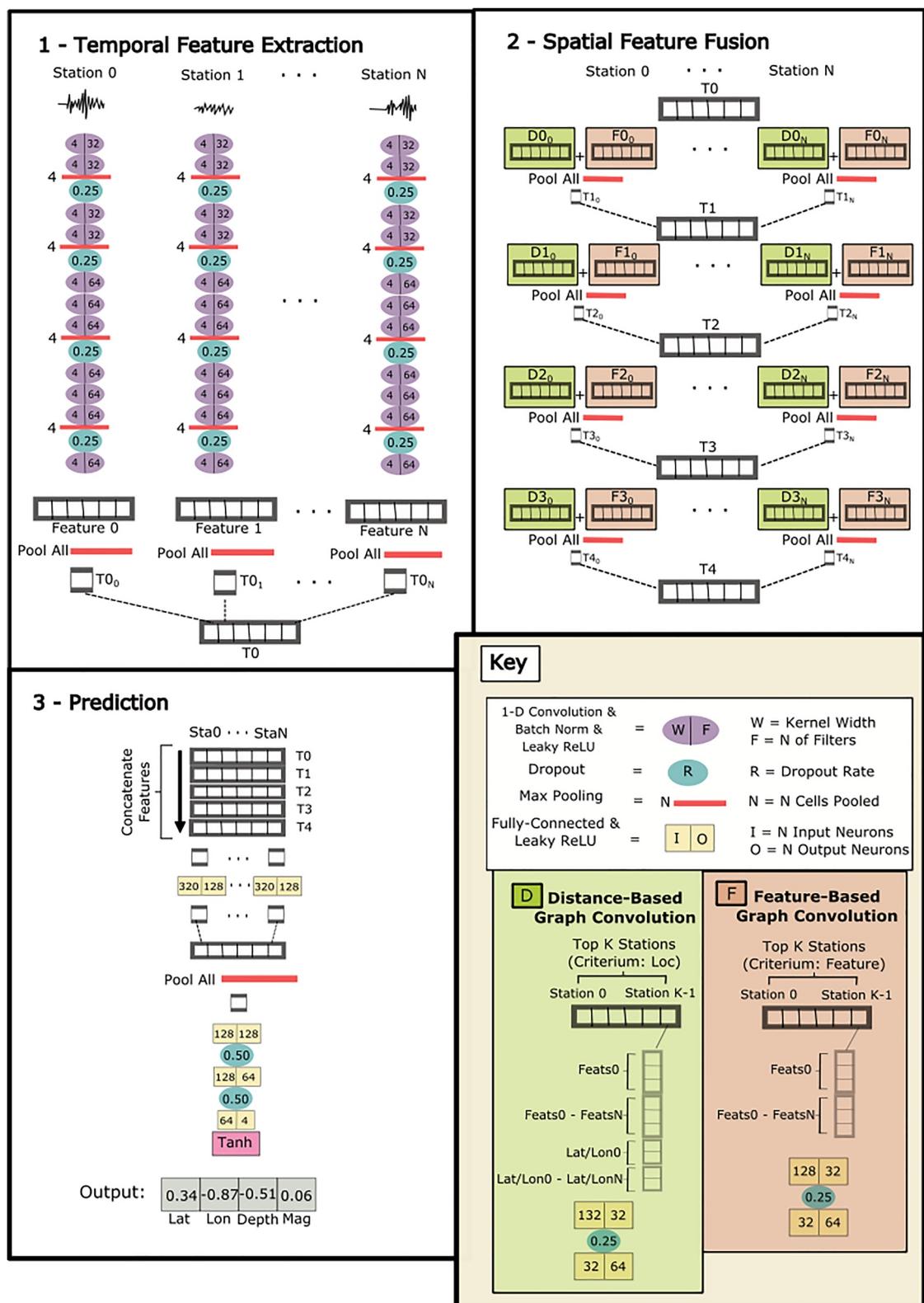
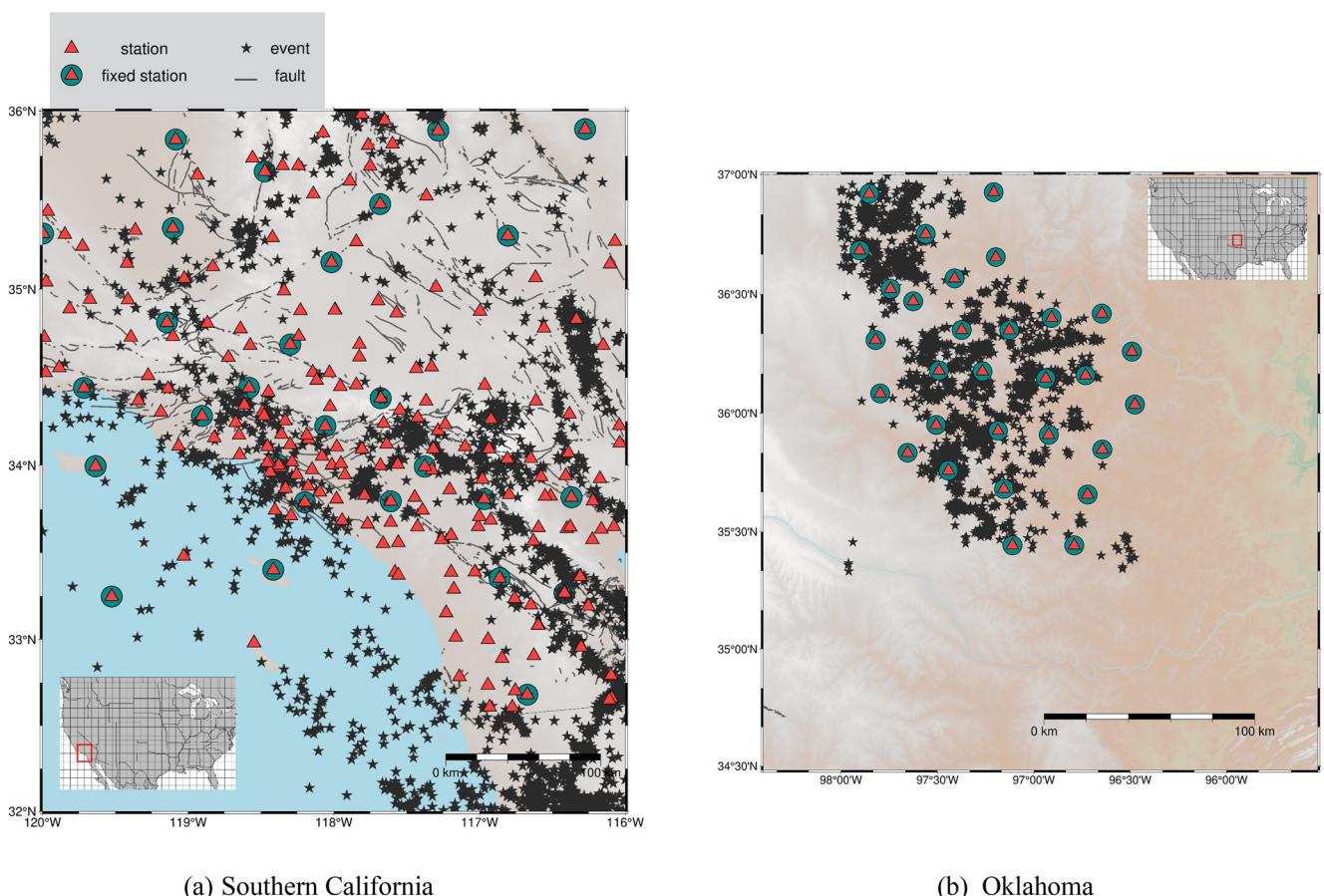


Figure 2.



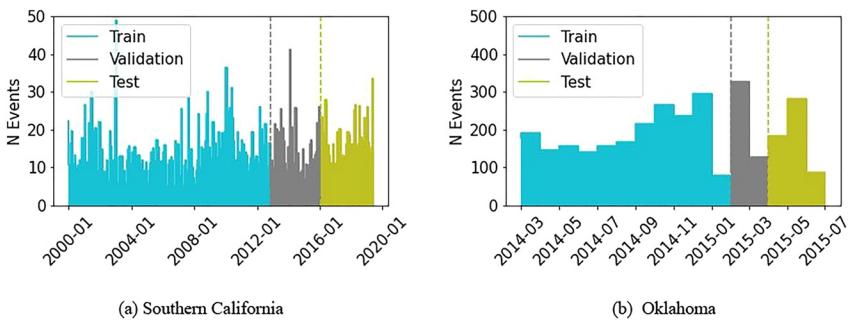
**Figure 3.** Maps of the two target regions used in this study: (a) Southern California and (b) Oklahoma. The distribution of all seismic stations (red triangles) and earthquakes (black stars) are shown. The 30 stations selected for fixed input testing are surrounded by a green circle. Black lines indicate seismic faults (United States Geological Survey and California Geological Survey, 2022).

The Oklahoma data set uses waveforms and catalog information collected by the Nanometrics Research Network and was used for training and testing in the FCN baseline (Zhang et al., 2020). The selected data set contains events from March 2014 to July 2015 (Nanometrics Seismological Instruments, 2013) within a geographic subset from  $34.482^{\circ}$  to  $37^{\circ}$  latitude and  $-98.405^{\circ}$  to  $-95.527^{\circ}$  longitude, a depth range of 0–12 km, and a magnitude range of  $1.5 < M < 4$ . The final data set contains 3,456 events recorded by 30 broadband seismic stations.

All waveforms and catalogs were accessed using ObsPy (Beyreuther et al., 2010). Each trace contains 200 s of seismic displacement collected by three orthogonal channels, which is interpolated into 4,096 evenly spaced samples, resulting in a sampling rate of approximately 20 Hz. For both data sets, the instrument response was removed, and waveforms were band-pass filtered from 1–8 Hz. As the recorded displacement amplitudes are very small, the waveforms are multiplied by a constant scaling factor of  $1e7$  to raise the input data to a numerically stable range close to  $[-1, 1]$  without eliminating magnitude information. A map of events and stations is shown in Figure 3.

One advantage of the graph neural network is its ability to make predictions using dynamic inputs (i.e., the selected stations and their order in the input file are not necessarily the same for each sample). To demonstrate this ability, we perform tests with STGNN and the GNN baselines using Southern California data with dynamic

**Figure 2.** Overview of Spatiotemporal Graph Neural Network, including three components outlined in Figure 1. (1) In Temporal Feature Extraction, standard convolution and maximum pooling along the time dimension reduce three-component waveforms to a feature vector of length 64. These feature vectors are concatenated to form T0. (2) In Spatial Feature Fusion, four layers of graph convolution are performed. In each layer, two graphs are constructed: Distance-Based (D) and Feature-Based (F), which are combined through element-wise summation and max pooled along the K-nearest neighbors dimension. (3) In Prediction, feature tensors T0...T4 are concatenated and regressed to normalized predictions of latitude, longitude, depth, and magnitude. The pooling is applied along the station dimension.



**Figure 4.** The monthly earthquake frequency distribution for (a) Southern California and (b) Oklahoma. The temporal boundaries between the training, validation, and testing data are indicated by color.

inputs, in which functioning stations are randomly selected for each event. However, the FCN baseline requires a fixed input, in which the same stations must occupy the same position for each sample. To make a fair comparison, we train STGNN as well as both baselines on 30 fixed stations to compare the performance of all methods. This results in three data sets: (a) Dynamic Southern California Data Set, in which 100 stations are randomly selected for each sample, (b) Fixed Southern California Data Set, and (c) Fixed Oklahoma Data Set, in which a static set of 30 stations are used for every sample. For all data sets, events are omitted where <25 stations are functioning.

### 3.2. Training Procedure

In the experiments, we use AdamW as the optimizer with a learning rate of  $3e - 4$ . The  $L_2$  regularization term  $\lambda$  is  $1e - 4$ . Models are trained for 400 epochs with early stopping after 50 epochs without validation error improvement, from which we select the model with the best validation performance. We use a 20–80 split to divide each data set into testing and training data, and reserve 20% of the training data for validation. The data sets are not randomly shuffled but rather separated by time in which training data precede testing data. This approach avoids potential information leakage (Kaufman et al., 2012), which might occur from spatially and temporally localized swarms. This method of splitting data also better simulates a real-use case, in which historic earthquakes would be used to train a model to detect more recent events on a network where station configuration and seismic characteristics may evolve over time. Figure 4 shows the monthly event frequency distribution in the training and testing data set.

We use a sliding window with a length of 100 s and a stride of 5 s to create 10 100 s samples from each 200 s recording. This augments the data by increasing the sample size and cropping different portions of the wave train, assuring that the model can be used without known origin and arrival times. Within each sample, the same time shift is applied to all stations (See Section S1 in Supporting Information S1 for visualization).

### 3.3. Performance Comparison

To evaluate our developed framework, we compare the performance of our model against two baselines: (a) van den Ende and Ampuero (2020) (referenced as GNN for graph neural network) and (b) Zhang et al. (2020) (referenced as FCN for fully convolutional network). The performance of each model is evaluated using the following metrics:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (5)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (6)$$

**Table 1**

Performance of the Spatiotemporal Graph Neural Network (STGNN) Model Proposed in This Paper and the Graph Neural Network (GNN) Baseline When Applied to the Southern California Data Set With Dynamic Inputs

	MAE (km)	MSE ( $10^2$ km)	$R^2$
Latitude			
STGNN	<b><math>7.548 \pm 9.841</math></b>	<b><math>1.538 \pm 9.255</math></b>	<b>0.980</b>
GNN	$10.201 \pm 11.791$	$2.431 \pm 12.438$	0.969
Longitude			
STGNN	<b><math>6.931 \pm 8.152</math></b>	<b><math>1.145 \pm 5.589</math></b>	<b>0.986</b>
GNN	$10.095 \pm 12.086$	$2.480 \pm 11.865$	0.970
Depth			
STGNN	<b><math>3.472 \pm 2.928</math></b>	<b><math>0.206 \pm 0.358</math></b>	<b>0.266</b>
GNN	$3.837 \pm 3.166$	$0.247 \pm 0.399$	0.120
MAE		MSE	$R^2$
Magnitude			
STGNN	<b><math>0.120 \pm 0.114</math></b>	<b><math>0.027 \pm 0.085</math></b>	<b>0.826</b>
GNN	$0.120 \pm 0.126$	$0.030 \pm 0.105$	0.807

Note. MAE refers to the mean absolute error (Equation 5) and MSE refers to the mean squared error (Equation 6), where a lower value indicates less error. The  $R^2$  value (Equation 7) is a measure of how strongly variation in the predicted values are related to variation in the ground truth value, where a value close to 1 is indicative of high accuracy. The values in bold are the ones with better performance.

$$R^2 = 1 - \sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{(y_i - \bar{y})^2}, \quad (7)$$

where  $\hat{y}_i$  is the model's prediction,  $y_i$  is the true value,  $\bar{y}$  is the average true value, and  $n$  is the total number of predictions.

Both STGNN and GNN make normalized predictions between  $-1$  and  $1$ . When calculating the above metrics, the values are first reverted from the normalized scalars to degrees of latitude and longitude, kilometers of depth, and magnitude values. Degrees of latitude and longitude are then converted to kilometers using conversions of  $110$  km/degree and  $92$  km/degree, respectively.

Testing is conducted across three data sets:

1. *Dynamic Southern California Data Set*: The performance is tested for the STGNN and GNN models. Five neighbors ( $K = 5$ ) were selected for feature update for STGNN. Results are detailed in Table 1.
2. *Fixed Southern California Data Set*: The performance is tested for the STGNN, GNN, and FCN models. Seven neighbors ( $K = 7$ ) were selected for feature update for STGNN. Results are detailed in Table 2.
3. *Fixed Oklahoma Data Set*: The performance is tested for the STGNN, GNN, and FCN models. Seven neighbors ( $K = 7$ ) were selected for feature update for STGNN. Results are detailed in Table 3.

The performance overview (Figure 5) demonstrates that our proposed model achieves a higher location accuracy than baselines for all data sets. STGNN makes predictions with an average of  $6.8$  km less location error, a  $40\%$  improvement across all tested data sets when compared to the FCN baseline. Across all data sets, STGNN makes predictions with an average of  $3.0$  km less location error than the GNN baseline, a  $22\%$  improvement. The improved location is primarily due to epicentral location accuracy. All tested models demonstrate low  $R^2$  values for depth prediction. STGNN and FCN achieve comparable magnitude prediction, and FCN does not support magnitude prediction. STGNN appears to incorporate a consistent bias, underpredicting magnitude values.

Figures 6–8 plot all predictions to give a richer understanding of model capacity beyond individual quality metrics. Observation of individual predictions makes it clear that while both models succeed in learning a meaningful

**Table 2**

Performance of Spatiotemporal Graph Neural Network (STGNN), Graph Neural Network (GNN), and Fully Convolutional Network (FCN) Baselines When Applied to the Southern California Data Set With Fixed Inputs

	MAE (km)	MSE ( $10^2$ km)	$R^2$
Latitude			
STGNN	<b><math>10.396 \pm 11.388</math></b>	$2.378 \pm 10.067$	<b>0.954</b>
GNN	$11.263 \pm 11.696$	$2.637 \pm 8.010$	0.949
FCN	$14.415 \pm 21.827$	$6.842 \pm 34.697$	0.869
Longitude			
STGNN	<b><math>9.663 \pm 13.048</math></b>	<b><math>2.636 \pm 15.761</math></b>	<b>0.962</b>
GNN	$11.485 \pm 12.199$	$2.807 \pm 10.252$	0.960
FCN	$16.369 \pm 24.872$	$8.865 \pm 47.323$	0.874
Depth			
STGNN	<b><math>4.030 \pm 3.396</math></b>	<b><math>0.278 \pm 0.405</math></b>	<b>-0.069</b>
GNN	$4.264 \pm 3.384$	$0.296 \pm 0.403$	-0.141
FCN	$4.105 \pm 3.324$	$0.279 \pm 0.431$	-0.074
	MAE	MSE	$R^2$
Magnitude			
STGNN	$0.216 \pm 0.151$	$0.069 \pm 0.083$	0.583
GNN	<b><math>0.120 \pm 0.118</math></b>	<b><math>0.028 \pm 0.088</math></b>	<b>0.830</b>

Note. MAE refers to the mean absolute error (Equation 5) and MSE refers to the mean squared error (Equation 6), where a lower value indicates less error. The  $R^2$  value (Equation 7) is a measure of how strongly variation in the predicted values are related to variation in the ground truth value, where a value close to 1 is indicative of high accuracy. The values in bold are the ones with better performance.

mapping to latitude and longitude predictions, depth predictions are highly scattered and are little better than predictions of the mean.

### 3.4. Stability Analysis

There are three critical hyperparameters in STGNN: the number of neighbors ( $K$ ) considered for edge generation, the maximum amount of observed stations, and the random selection of seismic stations when creating data sets. We use the Southern California data set to vary these hyperparameters in order to assess the stability of STGNN. The results of the parameter permutation are shown in Figure 9. When a parameter is not permuted, 100 stations, 5 edges, 4 graph convolutions, and a random seed of 0 is used.

For each prediction, a random subset of functional stations was selected. We permute the random seed during sample selection to alter the set of stations used for training. We find that the random subsets return similar results for all predictions except for magnitude, which shows a higher degree of variation. Prediction accuracy improves as more stations are used. Accuracy is moderately impacted by the number of edges selected, with magnitude predictions fluctuating most significantly. Overall, the model appears to be generally stable, with magnitude demonstrating the greatest sensitivity to hyperparameter tuning.

We also examine the impact of the number of GNN layers on the model's performance. A depth of four convolutions produces the best balance of location and magnitude accuracy and is used for this study.

## 4. Discussion

### 4.1. Architecture Strengths and Weaknesses

Our STGNN has several advantages over the FCN baseline model. One of the primary advantages is the ability to make predictions on a dynamic set of inputs, allowing the model to adapt to station outages, network alterations,

**Table 3**

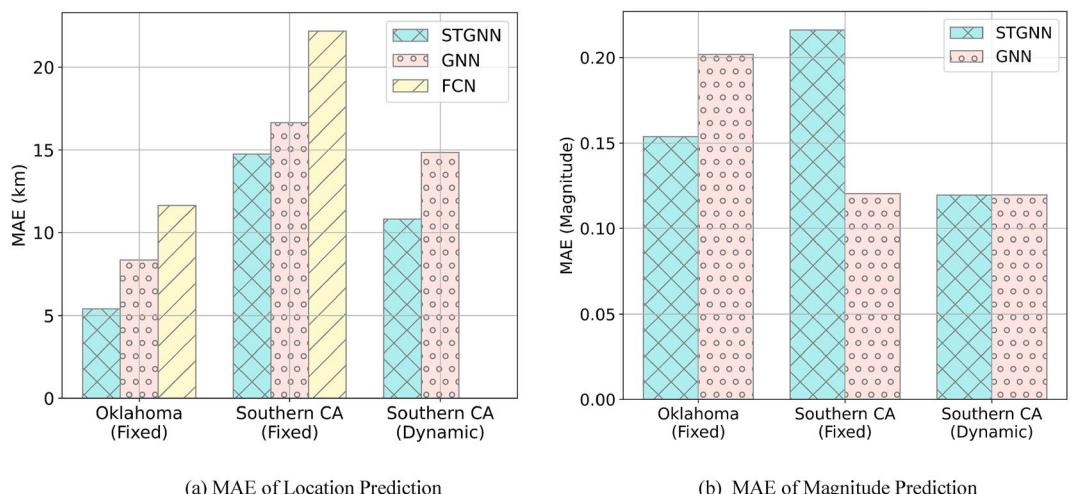
Performance of Spatiotemporal Graph Neural Network (STGNN), Graph Neural Network (GNN), and Fully Convolutional Network (FCN) Baselines When Applied to the Oklahoma Data Set With Fixed Inputs

	MAE (km)	MSE ( $10^2$ km)	$R^2$
<b>Latitude</b>			
STGNN	<b><math>3.574 \pm 5.755</math></b>	<b><math>0.459 \pm 3.665</math></b>	<b>0.975</b>
GNN	$7.166 \pm 12.414$	$2.055 \pm 14.820$	0.897
FCN	$9.219 \pm 16.418$	$3.545 \pm 23.070$	0.822
<b>Longitude</b>			
STGNN	<b><math>3.697 \pm 4.936</math></b>	<b><math>0.380 \pm 2.365</math></b>	<b>0.942</b>
GNN	$5.934 \pm 8.144$	$1.015 \pm 5.547$	0.904
FCN	$9.308 \pm 11.883$	$2.279 \pm 8.244$	0.785
<b>Depth</b>			
STGNN	<b><math>1.686 \pm 1.427</math></b>	<b><math>0.049 \pm 0.082</math></b>	0.036
GNN	$1.701 \pm 1.423$	$0.049 \pm 0.078$	<b>0.090</b>
FCN	$1.865 \pm 1.546$	$0.059 \pm 0.084$	-0.086
	MAE	MSE	$R^2$
<b>Magnitude</b>			
STGNN	<b><math>0.154 \pm 0.126</math></b>	<b><math>0.040 \pm 0.066</math></b>	<b>0.790</b>
GNN	$0.195 \pm 0.142$	$0.058 \pm 0.083$	0.681

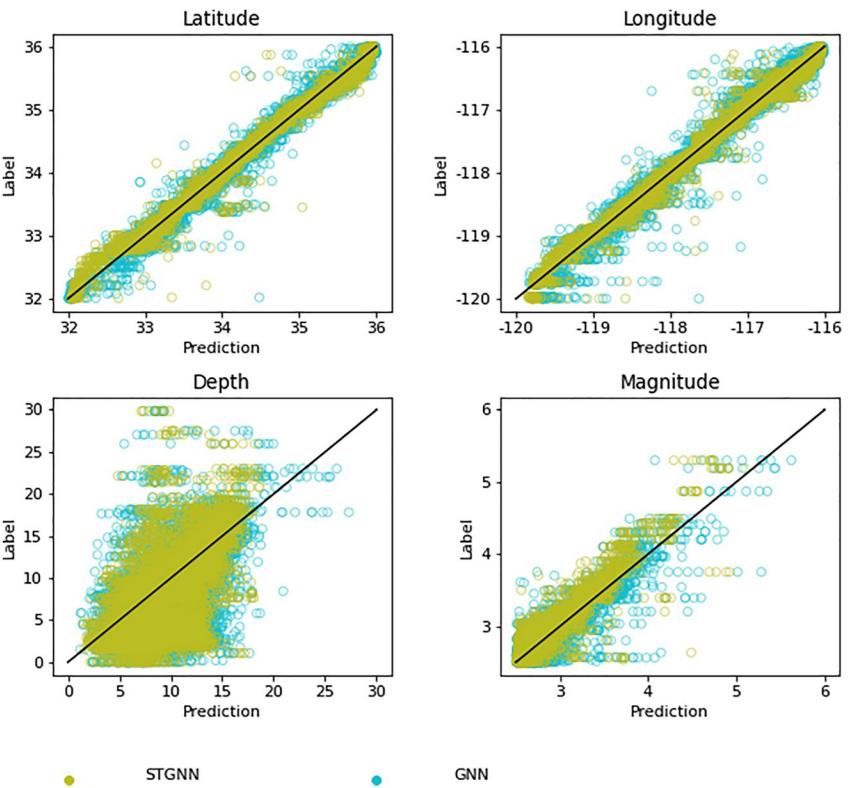
Note. MAE refers to the mean absolute error (Equation 5) and MSE refers to the mean squared error (Equation 6), where a lower value indicates less error. The  $R^2$  value (Equation 7) is a measure of how strongly variation in the predicted values are related to variation in the ground truth value, where a value close to 1 is indicative of high accuracy. The values in bold are the ones with better performance.

and station subsetting. As STGNN featurizes individual stations rather than an ordered network image, the model can be easily trained to predict using any number of stations without architectural alteration.

The FCN baseline uses an image-to-image strategy, outputting a probability volume in which the highest values correspond to the event location. This has the advantage of predicting a probability amplitude, which Zhang



**Figure 5.** (a) Mean absolute error (MAE) of each tested model where the location error refers to the Euclidean distance between the predicted location and the true event location. (b) MAE of the magnitude predictions from the graph convolutional neural networks when applied to the Oklahoma data set with 30 fixed stations, the Southern California data set with 30 fixed stations, and the Southern California data set with 100 dynamically selected stations.



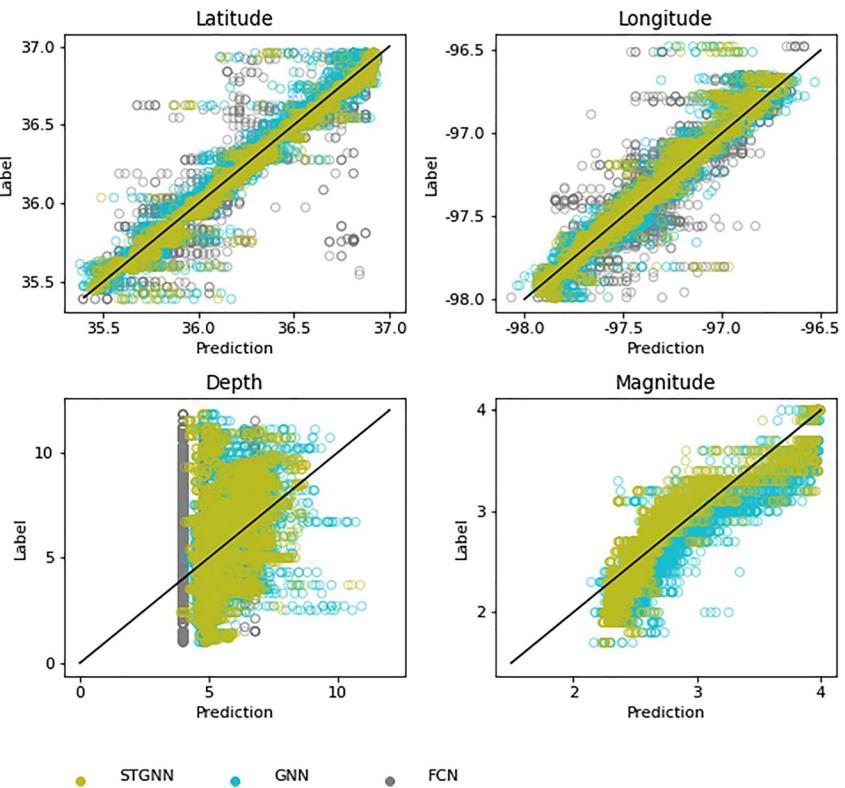
**Figure 6.** Testing comparison on 100 dynamically selected stations from the Southern California data set with five convolutional edges. “STGNN” and “GNN” denote the performance of our framework and van den Ende and Ampuero (2020), respectively. In the scatter plot, each point represents an event, and a position on the diagonal line corresponds to perfect agreement between the predicted value ( $x$ -axis) and the true value ( $y$ -axis). Latitude and longitude values are displayed in degrees and depth values are displayed in kilometers.

et al. (2020) demonstrate as a useful measure of prediction uncertainty, especially in cases where earthquakes occur outside the bounds of the modeled region. However, the volumetric output comes at the cost of resolution limitation due to discretization. The gridded, three-dimensional output also requires a high degree of model complexity. The FCN baseline consequently comprises approximately 27 million parameters while our STGNN with scalar predictions comprises fewer than 0.24 million parameters.

The baseline GNN (van den Ende & Ampuero, 2020) implements edgeless graph convolution (i.e., station-by-station convolutions with global pooling) while the STGNN model developed in this paper implements convolution and pooling over dynamically generated edges. Figure 10 gives insight into the edge generation process. For clear visualization, we select a case with 50 seismic stations with  $K = 5$ . In the edges generated by waveform similarity, stations that have recorded an event are generally connected to other recording stations, forming distinct clusters from edges generated by geographic proximity. This indicates that the model is able to successfully extract waveform information and associate stations in order to characterize an event. The graph generated based on the feature similarity is different from that created based on the location, showing that the feature similarity is a complement of location during aggregating features from different seismic stations.

A limitation that STGNN shares with the baselines is the ability to make predictions only within a certain range of area, depth, and magnitude. The model outputs normalized values between  $-1$  and  $1$ , which correspond to a range selected at the beginning of training. The spatial restrictions are similar to the bounds set in inversion-based methods and are arguably less limiting, as the predictions made by our model are continuous and therefore not bound by grid-spacing.

However, STGNN is more limited than nonmachine learning methods with regard to magnitude prediction. Magnitudes falling above or below the training range cannot be predicted by STGNN or the deep learning baselines. The limited range of predictions adversely impacts the usefulness of the deep learning methods for



**Figure 7.** Testing comparison on 30 fixed stations from the Oklahoma data set with seven convolutional edges. “STGNN,” “GNN,” and “FCN” denote the performance of our framework, van den Ende and Ampuero (2020) and Zhang et al. (2020). In the scatter plot, each point represents an event, and a position on the diagonal line corresponds to perfect agreement between the predicted value (x-axis) and the true value (y-axis). Latitude and longitude values are displayed in degrees and depth values are displayed in kilometers. Magnitude is omitted for the fully convolutional network (FCN), as this model makes only location predictions.

applications such as Earthquake Early Warning, where magnitude saturation must be avoided. The limitations posed by fixed prediction ranges are made less severe by STGNN’s ability to be tuned to new ranges with small amounts of training data (see Section 4.3). However, the fixed prediction ranges nonetheless represent a weakness in our framework.

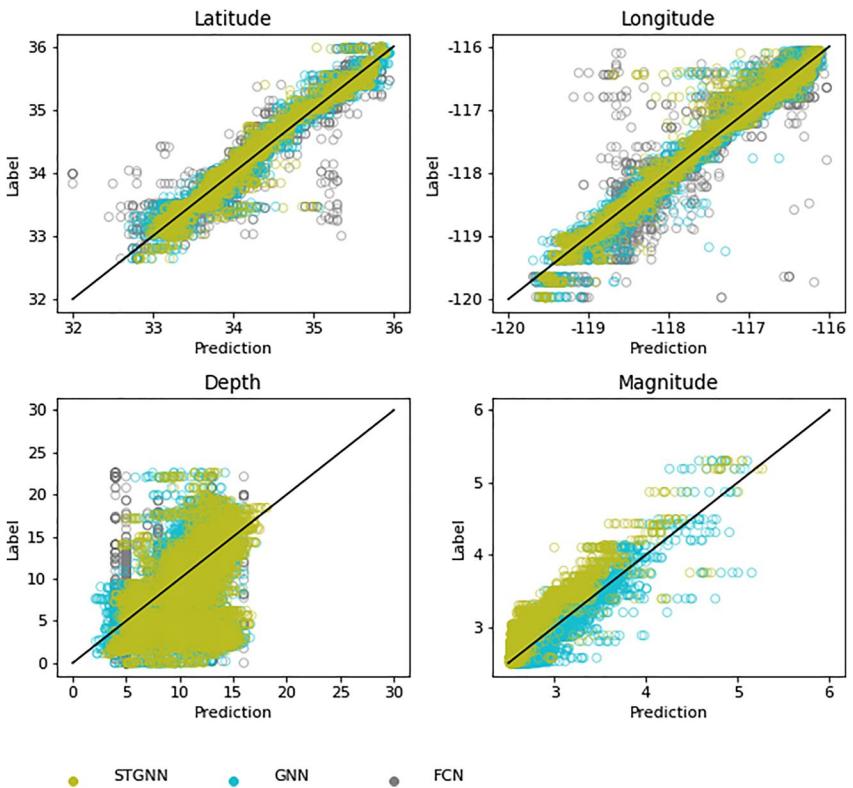
#### 4.2. Impacts on Location Prediction

Overall location error for the STGNN model is 5.41 km for the Fixed Oklahoma Data Set and 14.75 km for the Fixed Southern California Data Set. The higher loss for the Southern California data set may be attributable to the larger size of the region. As locations in both the smaller and larger regions are normalized to values between  $-1$  and  $1$ , errors in the initial prediction will result in larger errors when converted to kilometers in larger regions. In addition, larger regions may include a greater range of structural complexity that may be more challenging for the model to learn.

Location error for the Dynamic California Data Set was 3.93 km lower than that of the Fixed California Data Set. This supports the assumption that dynamic inputs improve not only the flexibility but also the performance of prediction models.

#### 4.3. Transferability

To test transferability, models trained on Oklahoma data were tested on Southern California data and vice versa (See Section S2 in Supporting Information S1). After training in one region, STGNN does not transfer well to other regions without retraining. This indicates that the models are encoding site-specific information such as

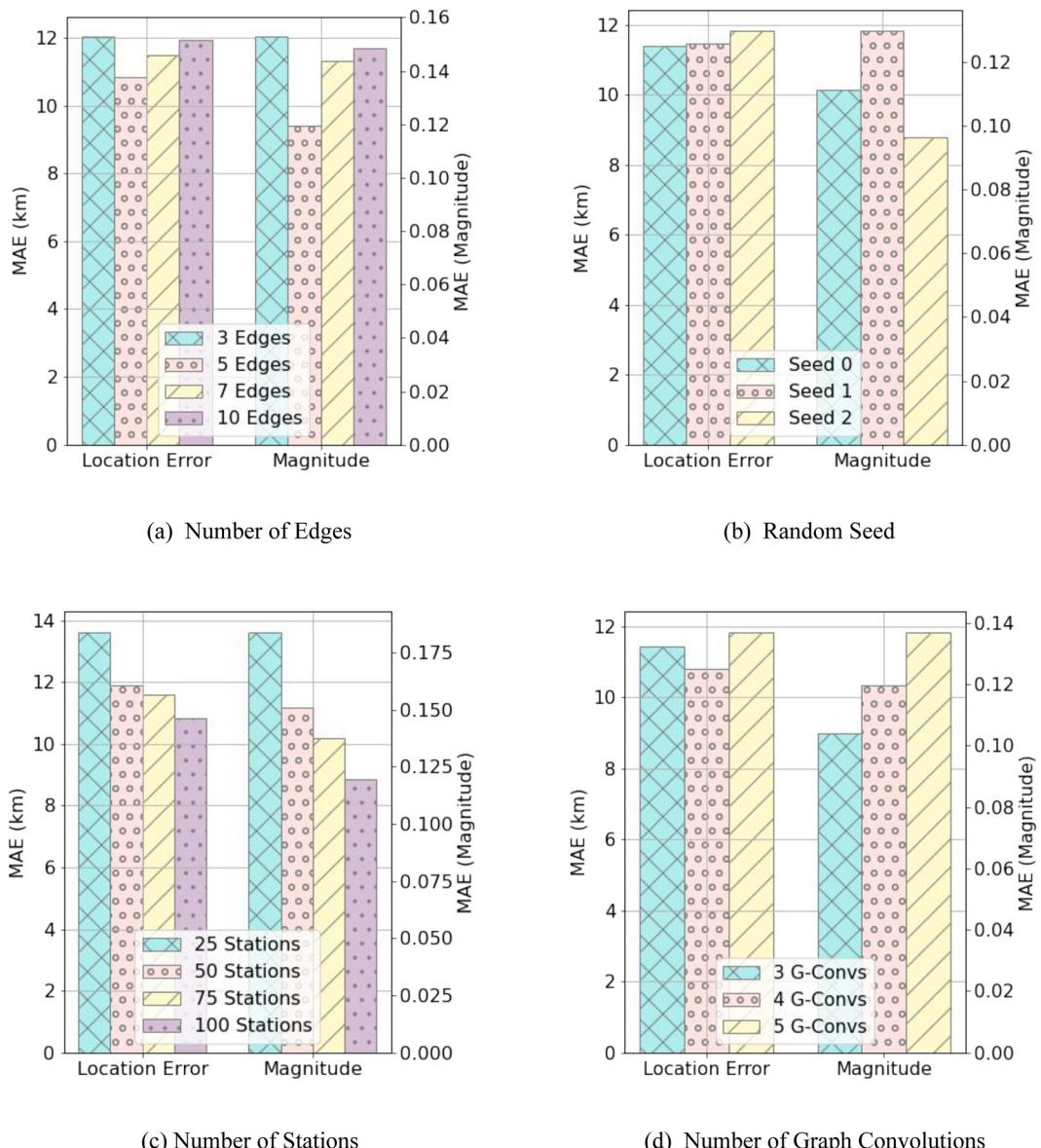


**Figure 8.** Testing comparison on 30 fixed stations from the Southern California data set with seven convolutional edges. “STGNN,” “GNN,” and “FCN” denote the performance of our framework, van den Ende and Ampuero (2020) and Zhang et al. (2020). In the scatter plot, each point represents an event, and a position on the diagonal line corresponds to perfect agreement between the predicted value ( $x$ -axis) and the true value ( $y$ -axis). Latitude and longitude values are displayed in degrees and depth values are displayed in kilometers. Magnitude is omitted for the fully convolutional network (FCN), as this model makes only location predictions.

velocity structure or types of seismicity (i.e., anthropogenically induced earthquakes in the Oklahoma data set) as well as magnitude ranges. Performance improves significantly when a small amount of training data is used to tune the model. Using transfer learning to adapt a model from one region to another is more effective than training a randomized model when a limited data set is available. However, best results are achieved when a model is trained for the region of implementation using a catalog of several hundred events.

#### 4.4. Synthetic Testing

While substantial improvements have been made in the prediction of latitude and longitude, magnitude does not improve in every data set, and depth predictions are inaccurate for all models. To test the capacity of our model under ideal circumstances, we train our model using synthetic data. The synthetic waveforms are generated using Green’s Functions created with PyFk Xi (2020–2022) from a 1-D sedimentary half-space model, with an epicentral resolution of 1 km, a depth resolution of 0.5 km, and a sampling rate of 20.48 Hz. Thirty recording sensors are used with the same configuration as the Fixed Oklahoma Data Set. No label or waveform noise is applied. The high degree of accuracy suggests that the fundamental architecture of the model has the capacity to learn depth and magnitude estimation (Figure 11). The difference between the simulated waveforms and the recorded data are (a) label noise, (b) waveform noise, and (c) subsurface complexity. While the fundamental structure holds promise, STGNN must be improved to address these factors before it can be effectively applied to real seismic data sets for depth and magnitude prediction.

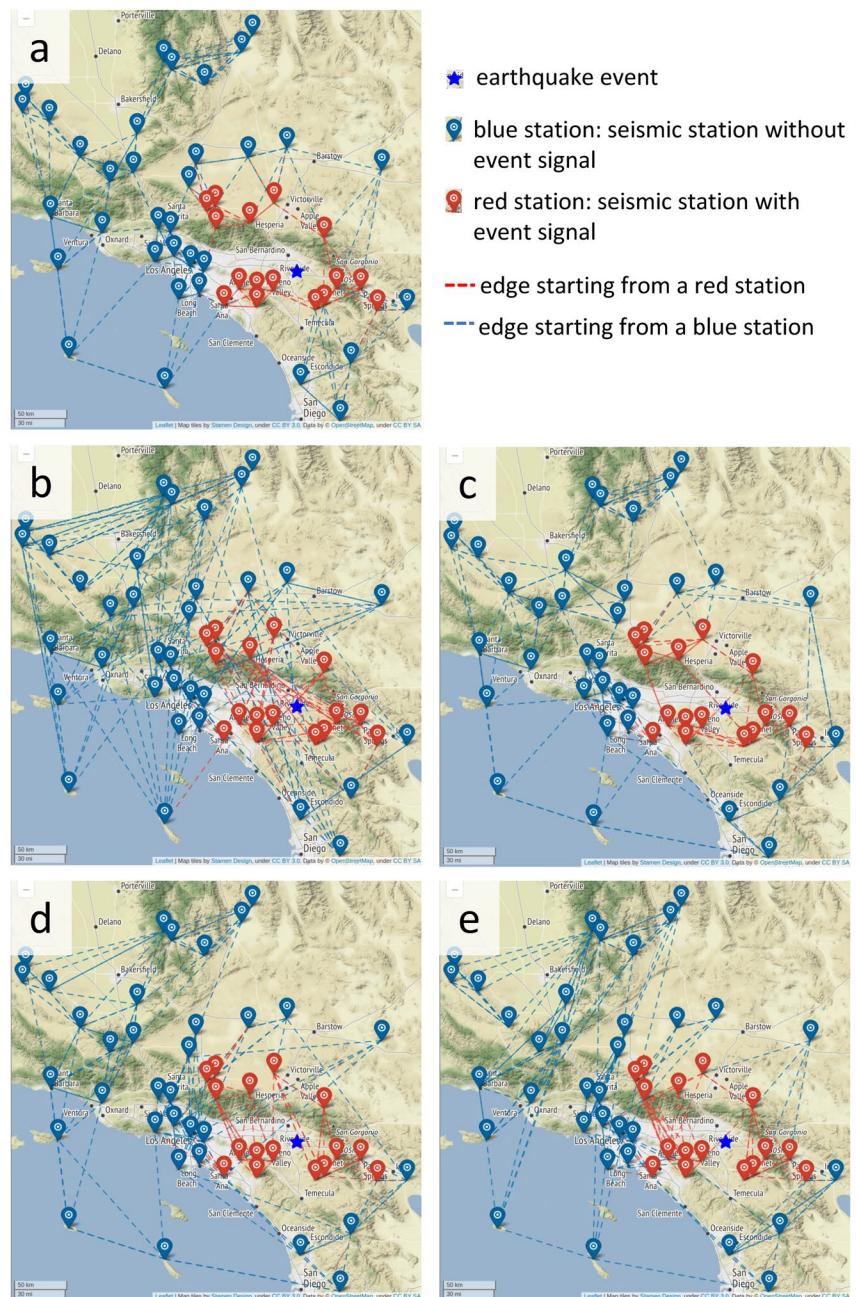


**Figure 9.** Stability analysis permuting (a) the number of edges used to connect nodes during graph convolution, (b) the random seed used to select stations for the model input, (c) the number of stations used for prediction, and (d) the number of graph convolutions implemented. When a parameter is not permuted, 100 stations, 5 edges, 4 graph convolutions, and a random seed of 0 is used.

## 5. Conclusions and Future Work

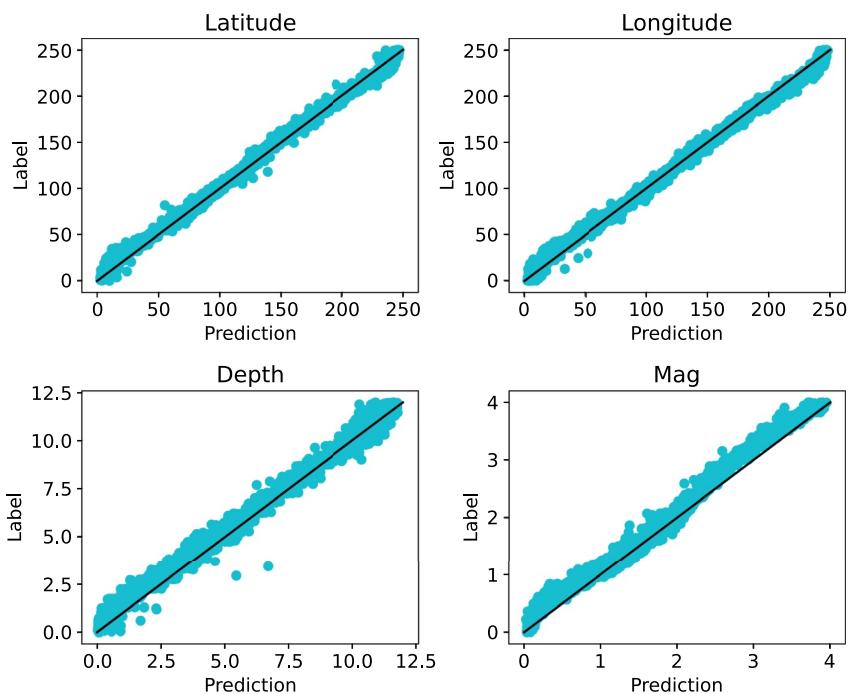
In this work, we design a graph convolutional neural network for earthquake source characterization based on waveform records from multiple stations. With experiments performed in two seismic environments, we demonstrate that STGNN outperforms both the FCN and GNN baselines. One of the major advantages of our framework is that STGNN does not require static input or a manually generated graph structure. Instead, all feature generation and fusion processes are learned automatically from the data to synthesize waveform features and spatial data.

Future improvements to our work include enhancing the model's ability to predict depth and magnitude. This may be overcome with higher quality training data, or through methods such as physics-informed machine learning. Our work thus far has focused on developing architecture to characterize an earthquake given a discrete time series known to contain an event. Further adaptation of the core model is required to effectively process continuous waveforms in which an event may not be present, or in which multiple events are contained within one



**Figure 10.** Graphs constructed by different layers of the graph neural network, (a) graph convolution layer based on geographic distance among seismic stations (b) 1st, (c) 2nd, (d) 3rd, and (e) 4th graph convolution layer based on the extracted feature similarity. The information from stations with the event signal are clustered in deeper layers.

window. An additional feature to incorporate is uncertainty quantification. Given the relatively high degree of error in all methods for earthquake location, uncertainty is a standard feature in comprehensive catalogs. Uncertainty can be incorporated internally (i.e., to aid in station selection) and also applied the final predictions to identify poorly constrained events. Another interesting application is to transform the learning process in an online learning manner in which a model might adaptively retrain as more recent earthquakes are included in the catalog.



**Figure 11.** Testing performance of Spatiotemporal Graph Neural Network on synthetic data. In the scatter plot, each point represents an event, and a position on the diagonal line corresponds to perfect agreement between the predicted value ( $x$ -axis) and the true value ( $y$ -axis). Latitude and longitude values are displayed in degrees and depth values are displayed in kilometers.

## Data Availability Statement

Waveform data used in this study were downloaded from the IRIS Web Services (<https://service.iris.edu/>) and the Southern California Earthquake Data Center (SCEDC, 2013). The data can be accessed through FDSN: Oklahoma (NX; <https://doi.org/10.7914/sn/n4>) (Albuquerque Seismological Laboratory (ASL)/USGS, 2013) and Southern California (CI; <https://doi.org/10.7914/sn/ci>) (California Institute of Technology and United States Geological Survey Pasadena, 1926). Faults were downloaded from the US Quaternary Fault and Fold Database (<https://doi.org/10.5066/F7S75FJM>) (United States Geological Survey and California Geological Survey, 2022). The maps in our paper were made using Generic Mapping Tools (Wessel et al., 2013) and figures were created with Matplotlib (Hunter, 2007) and Inkscape (Inkscape Project, 2020). Synthetic waveforms were created with PyFk (Xi, 2020–2022).

## Acknowledgments

This work was supported by the Center for Space and Earth Science at Los Alamos National Laboratory (LANL) and by the Laboratory Directed Research and Development program of LANL under project numbers of 20210542MFR. MZ was supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (RGPIN-2019-04297).

## References

- Albuquerque Seismological Laboratory (ASL)/USGS. (2013). *Central and Eastern US Network*. International Federation of Digital Seismograph Networks. Retrieved from <https://www.fdsn.org/networks/detail/N4/>
- Bergen, K. J., Johnson, P. A., Maarten, V., & Beroza, G. C. (2019). Machine learning for data-driven discovery in solid earth geoscience. *Science*, 363(6433), eaau0323. <https://doi.org/10.1126/science.aau0323>
- Beskardes, G. D., Hole, J. A., Wang, K., Michaelides, M., Wu, Q., Chapman, M. C., et al. (2018). A comparison of earthquake back-projection imaging methods for dense local arrays. *Geophysical Journal International*, 212(3), 1986–2002. <https://doi.org/10.1093/gji/ggx520>
- Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., & Wassermann, J. (2010). ObsPy: A python toolbox for seismology. *Seismological Research Letters*, 81(3), 530–533. <https://doi.org/10.1785/gssr.81.3.530>
- California Institute of Technology and United States Geological Survey Pasadena. (1926). *Southern California seismic network*. International Federation of Digital Seismograph Networks. Retrieved from <https://www.fdsn.org/networks/detail/CI>
- Gajewski, D., Anikiev, D., Kashtan, B., & Tessmer, E. (2007). Localization of seismic events by diffraction stacking. In *SEG technical program expanded abstracts 2007* (pp. 1287–1291).
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Hutton, K., Woessner, J., & Hauksson, E. (2010). Earthquake monitoring in southern California for seventy-seven years (1932–2008). *Bulletin of the Seismological Society of America*, 100(2), 423–446. <https://doi.org/10.1785/0120090130>
- Inkscape Project. (2020). Inkscape. Retrieved from <https://inkscape.org>
- Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4), 1–21. <https://doi.org/10.1145/2382577.2382579>

- Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P. (2019). Machine learning in seismology: Turning data into insights. *Seismological Research Letters*, 90(1), 3–14. <https://doi.org/10.1785/0220180259>
- Kriegerowski, M., Petersen, G. M., Vasyura-Bathke, H., & Ohrnberger, M. (2019). A deep convolutional neural network for localization of clustered earthquakes based on multistation full waveforms. *Seismological Research Letters*, 90(2A), 510–516. <https://doi.org/10.1785/0220180320>
- Li, L., Tan, J., Schwarz, B., Stanek, F., Poiata, N., Shi, P., et al. (2020). Recent advances and challenges of waveform-based seismic location methods at multiple scales. *Reviews of Geophysics*, 58(1), e2019RG000667. <https://doi.org/10.1029/2019rg000667>
- Li, Z., Meier, M.-A., Hauksson, E., Zhan, Z., & Andrews, J. (2018). Machine learning seismic wave discrimination: Application to earthquake early warning. *Geophysical Research Letters*, 45(10), 4773–4779. <https://doi.org/10.1029/2018gl077870>
- Li, Z., & van der Baan, M. (2016). Microseismic event localization by acoustic time reversal extrapolation. *Geophysics*, 81(3), KS123–KS134. <https://doi.org/10.1190/geo2015-0300.1>
- Lin, Y., Syracuse, E. M., Maceira, M., Zhang, H., & Larmat, C. (2015). Double-difference traveltime tomography with edge-preserving regularization and a priori interfaces. *Geophysical Journal International*, 201(2), 574–594. <https://doi.org/10.1093/gji/ggv047>
- McBrearty, I. W., & Beroza, G. C. (2022). Earthquake location and magnitude estimation with graph neural networks. *arXiv preprint*. arXiv:2203.05144.
- Mousavi, S. M., & Beroza, G. C. (2020a). Bayesian-deep-learning estimation of earthquake location from single-station observations. *IEEE Transactions on Geoscience and Remote Sensing*, 58(11), 1–14. <https://doi.org/10.1109/tgrs.2020.2988770>
- Mousavi, S. M., & Beroza, G. C. (2020b). A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*, 47(1), e2019GL085976. <https://doi.org/10.1029/2019gl085976>
- Münchmeyer, J., Bindl, D., Leser, U., & Tilmann, F. (2020). The transformer earthquake alerting model: A new versatile approach to earthquake early warning. *Geophysical Journal International*, 225(1), 646–656. <https://doi.org/10.1093/gji/ggaa609>
- Münchmeyer, J., Bindl, D., Leser, U., & Tilmann, F. (2021). Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network. *Geophysical Journal International*, 226(2), 1086–1104. <https://doi.org/10.1093/gji/ggab139>
- Nanometrics Seismological Instruments. (2013). Nanometrics Research Network. International Federation of Digital Seismograph Networks. Retrieved from <https://www.fdsn.org/networks/detail/NX/>
- Perol, T., Gharbi, M., & Denolle, M. (2018). Convolutional neural network for earthquake detection and location. *Science Advances*, 4(2), e1700578. <https://doi.org/10.1126/sciadv.1700578>
- Pesicek, J. D., Child, D., Artman, B., & Cieslik, K. (2014). Picking versus stacking in a modern microearthquake location: Comparison of results from a surface passive seismic monitoring array in Oklahoma. *Geophysics*, 79(6), KS61–KS68. <https://doi.org/10.1190/geo2013-0404.1>
- Ross, Z. E., Yue, Y., Meier, M.-A., Hauksson, E., & Heaton, T. H. (2019). PhaseLink: A deep learning approach to seismic phase association. *Journal of Geophysical Research: Solid Earth*, 124(1), 856–869. <https://doi.org/10.1029/2018jb016674>
- SCEDC. (2013). Southern California earthquake center [Dataset]. Caltech. <https://doi.org/10.7909/C3WD3xH1>
- Shen, H., & Shen, Y. (2021). Array-based convolutional neural networks for automatic detection and 4D localization of earthquakes in Hawai ‘i. *Seismological Society of America*, 92(5), 2961–2971. <https://doi.org/10.1785/0220200419>
- Tiira, T. (1999). Detecting teleseismic events using artificial neural networks. *Computers & Geosciences*, 25(8), 929–938. [https://doi.org/10.1016/s0098-3004\(99\)00056-4](https://doi.org/10.1016/s0098-3004(99)00056-4)
- United States Geological Survey and California Geological Survey. (2022). Quaternary fault and fold database for the United State. Retrieved from <https://www.usgs.gov/natural-hazards/earthquake-hazards/faults>
- van den Ende, M. P., & Ampuero, J.-P. (2020). Automated seismic source characterisation using deep graph neural networks. *Geophysical Research Letters*, 47(17), e2020GL088690. <https://doi.org/10.1029/2020GL088690>
- Wang, J., & Teng, T. (1995). Artificial neural network-based seismic detector. *Bulletin of the Seismological Society of America*, 85(1), 308–319. <https://doi.org/10.1785/bssa0850010308>
- Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics*, 38(5), 1–12. <https://doi.org/10.1145/3326362>
- Wessel, P., Smith, W., Scharroo, R., Luis, J., & Wobbe, F. (2013). Generic mapping tools: Improved version released. *EOS*, 94(45), 409–410. <https://doi.org/10.1002/2013eo450001>
- Xi, Z. (2020–2022). Pyfk <https://github.com/ziyixi/pyfk>
- Yano, K., Shiina, T., Kurata, S., Kato, A., Komaki, F., Sakai, S., & Hirata, N. (2021). Graph-partitioning based convolutional neural network for earthquake detection using a seismic array. *Journal of Geophysical Research: Solid Earth*, 126(5), e2020JB020269. <https://doi.org/10.1029/2020jb020269>
- Zhang, H., & Thurber, C. H. (2003). Double-difference tomography: The method and its application to the Hayward Fault, California. *Bulletin of the Seismological Society of America*, 93(5), 1875–1889. <https://doi.org/10.1785/0120020190>
- Zhang, X., Zhang, J., Yuan, C., Liu, S., Chen, Z., & Li, W. (2020). Locating induced earthquakes with a network of seismic station in Oklahoma via a deep learning method. *Scientific Reports*, 10(1), 1941. <https://doi.org/10.1038/s41598-020-58908-5>
- Zhang, X., Zhang, M., & Tian, X. (2021). Real-time earthquake early warning with deep learning: Application to the 2016 m 6.0 central apennines, Italy earthquake. *Geophysical Research Letters*, 48(5), 2020GL089394. <https://doi.org/10.1029/2020gl089394>
- Zhang, Z., Rector, J. W., & Nava, M. J. (2017). Simultaneous inversion of multiple microseismic data for event locations and velocity model with Bayesian inference. *Geophysics*, 82(3), KS27–KS39. <https://doi.org/10.1190/geo2016-0158.1>
- Zhebel, O., & Eisner, L. (2015). Simultaneous microseismic event localization and source mechanism determination. *Geophysics*, 80(1), KS1–KS9. <https://doi.org/10.1190/geo2014-0055.1>
- Zhu, W., & Beroza, G. C. (2019). PhaseNet: A deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, 216(1), 261–273.
- Zhu, W., Mousavi, S. M., & Beroza, G. C. (2019). Seismic signal denoising and decomposition using deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11), 9476–9488. <https://doi.org/10.1109/tgrs.2019.2926772>