Spatio-Temporal Graph Convolutional Networks for Earthquake Source Characterization

*

Xitong Zhang[1,2,*], Will Reichard-Flynn[1,*], Miao Zhang[3], Matthew Hirn[2,4,5], and Youzuo Lin[1,◇]

[1]Geophysics Group, Earth and Environmental Sciences Division, Los Alamos National Laboratory

[2]Department of Computational Mathematics, Science and Engineering, Michigan State University

[3]Department of Earth and Environmental Sciences, Dalhousie University

[4]Department of Mathematics, Michigan State University

[5]Center for Quantum Computing, Science and Engineering, Michigan State University

Key Points:

- We design multiple station-based graph deep neural networks for earthquake source characterization.
- Both geographic distance and waveform feature simliarity 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.

*: X. Zhang and W. Reichard-Flynn equally contribute to this work.

Corresponding author: ◇ Y. Lin, ylin@lanl.gov

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). Most 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 in graph-structured data, we develop a Spatio-Temporal Graph Convolutional Neural Network (STGCN) 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 an adaptive feature integration process. Given input waveforms collected from multiple stations, the neural network constructs different graphs and fuses spatial-temporal consistency effectively from various stations based on graphs' edges. Using a recent graph neural network and a fully convolutional neural network as baselines, we apply STGCN to earthquakes cataloged by Southern California Seismic Network from 2000 to 2019 and induced earthquakes collected in Oklahoma from 2014 to 2015. STGCN 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 graph neural networks and multiple stations for better automatic estimation of earthquake epicenters.

Plain Language Summary

Machine learning-based approaches have recently become prevalent in seismological tasks such as earthquake source characterization, which is the interest of this paper. The location and magnitude of an earthquake can be best determined by relating the motion recorded at multiple stations in a network. Therefore, it would be beneficial to

combine the waveforms from multiple seismic stations for 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 datasets and compared to two published baselines (graph convolutional neural network and fully convolutional network). Compared with the baselines, STGCN 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. (L. 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 traveltime-based inversion (Z. Zhang et al., 2017; Z. Li & van der Baan, 2016; Lin et al., 2015; H. Zhang & Thurber, 2003) and waveform-based inversion (Beskardes et al., 2018; Zhebel & Eisner, 2015; Pesicek et al., 2014; Gajewski et al., 2007). Traveltime-based methods implement a multi-step process, in which the arrival times of P and S waves are determined through phase detection and then 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. Though traveltime-based methods are commonly used in seismic applications, they are susceptible to 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 in-

version integrates all phase and amplitude information recorded in seismographs, resulting in high quality source characterization, however, which 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 datasets have allowed machine learning techniques to attain spectacular success in seismological applications (Kong et al., 2019; Bergen et al., 2019) including phase picking (Zhu & Beroza, 2019), seismic discrimination (Z. Li et al., 2018), waveform denoising (Zhu et al., 2019), phase association (Ross et al., 2019), earthquake location (Perol et al., 2018), as well as magnitude estimation (Mousavi & Beroza, 2020b). Although machine learning has long been applied to seismic event detection (J. 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 (CNN's) 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, multi-station 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. X. 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 (X. 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: (1) CNN's are designed to function on evenly-spaced grids (i.e. photographs) where information is exclusively shared between adjacent cells, and (2) CNN's require the input of station locations to be static (i.e. recordings from station 01 must always be found at position 01 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 non-adjacent 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.

Münchmeyer et al. (2020) developed an attention-based transformer model for earthquake early warning, which was extented to predict hypocenters and magnitudes of events in Münchmeyer et al. (2021). While this model is successful in implementing a multi-station approach that allows for dynamic inputs, high computational complexity restricts inputs to a relatively small number of stations. Another method for implementing flexible, multi-station 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 multi-station 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 ob-

tained by feature fusion along the spatial dimension. Yano et al. (2021) introduce a multi-station technique in which edges are manually constructed. While this technique allows for more meaningful features to be constructed than in global pooling, manually-selected edges require station inputs to be fixed during training and implementation, introducing the same limitation inherent to CNN's. Similarly, McBrearty and Beroza (2022) proposes a GCN framework using multiple pre-defined graphs constructed on both labels and station locations. The model requires the arrival time and is evaluated by a synthetic dataset.

In this study, to harness the full functionality of Graph Convolutional Neural Networks (GCN's) while maintaining flexibility in the location and number of seismic stations, we design a data-driven framework, spatio-temporal graph convolutional neural network (STGCN), that creates edges automatically to combine waveform features and spatial information. In order to evaluate the performance of our approach, we compare STGCN to two baselines: the GCN model designed by van den Ende and Ampuero (2020) and the Fully Convolutional Network (FCN) designed by X. Zhang et al. (2020). We apply all three models to the two datasets oupon which the baselines were originally tested and trained: (1) regional $2.5 < M < 6$ earthquakes recorded by 185 seismic stations in Southern California from 2000 to 2019 (van den Ende & Ampuero, 2020), and (2) local $0 < M < 4$ earthquakes recorded by 30 seismic stations in Oklahoma from 2014 to 2015 (X. Zhang et al., 2020). Next, model stability is evaluated with different hyperparameters. Finally, we examine the transferrability of STGCN to seismic networks outside of the training domain.

The layout of this article is as follows. In Section 2, we describe the fundamentals of graph-based CNN models and STGCN. 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 STGCN in the context of previous work. Finally, in Section 5, we present concluding remarks and discuss future work.

2 Methodology

In this section, we describe our framework and the major components of our STGCN. A graph is constructed by a set of nodes and edges. Our proposed framework constructs input-dependent graphs automatically, in which a node represents a seismic station and the edge connecting two nodes denotes that extracted features from these two nodes will be combined during convolution. The input to the network is collection of three-channel waveforms from each seismic station, along with the latitude and longitude of the recording stations. The output is the earthquake magnitude and location denoted by latitude, longitude and depth.

2.1 Overview

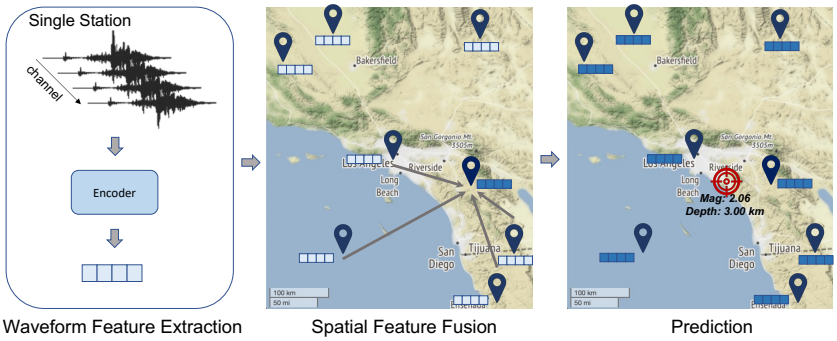


Figure 1: The overview of STGCN. There are three major components in STGCN: (1) Waveform feature extraction for obtaining time domain feature from each station independently. (2) Spatial feature fusion for time domain feature integration from different stations based on their geographic locations and extracted feature similarity. (3) Earthquake location and magnitude prediction given spatial features from the previous step.

Graph convolutional neural networks (GCN's) are designed to handle graphical data, or data that can be represented by vertices connected by edges. In GCN's, convolution and pooling operates along connecting edges. In CNN's, on the other hand, convolution and pooling operates on regions closest together on a Euclidean grid, meaning that in-

put order directly impacts information-sharing and featurization. This is not the case for GCN's, in which edges are not restricted to Euclidean grids but may instead be constructed by any criteria. Two major advantages of GCN architectures are that they do not require a fixed input order, and can handle graphs with different sets of vertices. These properties of GCN's fit well in 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 GCN framework as the basis for STGCN.

Inspired by Y. 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:

- Waveform feature extraction: We first extract time-domain 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.
- 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 view of the recorded wavefield over the seismic network. The graph convolutional architecture considers both geographic locations and waveform feature similarity among multiple seismic stations.

- 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

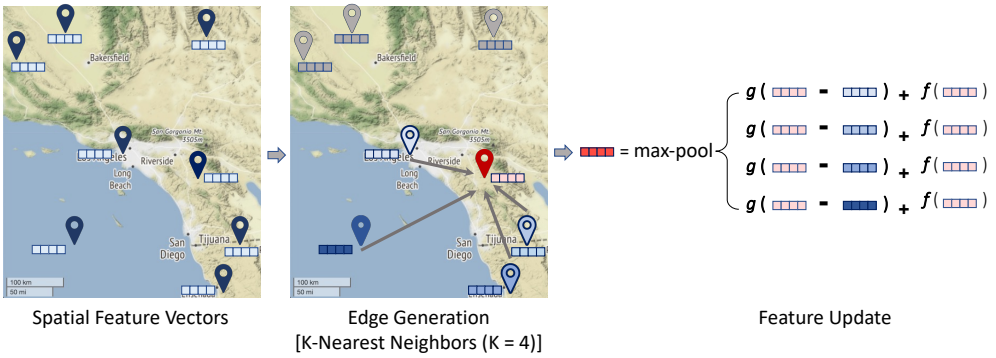


Figure 2: The overview of a graph convolutional layer. Each graph convolutional layer consists of two parts: (1) Edge generation among different stations. (2) Feature updating for each station based on the generated links. In the figure, the feature of the red station is updated based on four nearby blue stations. $g(\cdot)$ and $f(\cdot)$ represent two learnable networks.

The spatial feature fusion process is the most important component and consists of four 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. As shown in the Figure 2, each graph convolution layer can be broken down into two steps:

- 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 station i and j directly by $||x_i - x_j||_2^2$, and thus we can combine two waveform features from two stations further away, 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 hyperparapeter. In our framework, both geographic proximity and waveform feature similarity are considered. 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

$$\widetilde{x_i} = \max_{j \in \mathcal{N}_{\text{distance}}(i)} g(x_i - x_j) + f(x_i) + \max_{j' \in \mathcal{N}_{\text{feature}}(i)} g(x_i - x'_j) + f(x_i), \tag{1}$$

where the max operation refers to the element-wise max-pooling. 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. $g(\cdot)$ and $f(\cdot)$ are two trainable fully connected neural networks. $\widetilde{x_i}$ 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. The update is asymmetric for station i and j to encourage the update pro-

cesses of i and j to be different, as it is possible that only one of the stations records the event.

2.3 Architecture

A graphical illustration of the architecture is presented in Figure 3. Time domain waveform features are extracted from each station independently using an encoder with eleven convolutional layers. The extracted features are used in spatial feature fusion, in which time domain features are concatenated to station locations before each graph convolution. Our STGCN uses four groups of graph convolutional layers to obtain spatially hierarchical features. Two graphs are generated within each group: one in which edges are generated based on geographic distance, and one in which edges are generated based on waveform feature similarity. After convolution, the features obtained from both graphs are summed together prior to max pooling. For graphs in which geographic distance dictates edges, two scalars containing station coordinates are concatenated to each updated feature before each convolution. After all four groups of convolutions, the features from each group are concatenated together as a hierarchical representation for final source characterization regression.

After all feature outputs are concatenated, the features are individually processed with a final CNN layer. The output is then regressed to scalar predictions of latitude, longitude, depth, and magnitude using a fully-connected neural network. The objective function is

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{4}||y_i - \widehat{y_i}|| \tag{2}$$

where $\widehat{y_i}$ and y_i are the prediction and ground truth values of ith sample, respectively, represented as vectors of latitude, longitude, depth and magnitude.

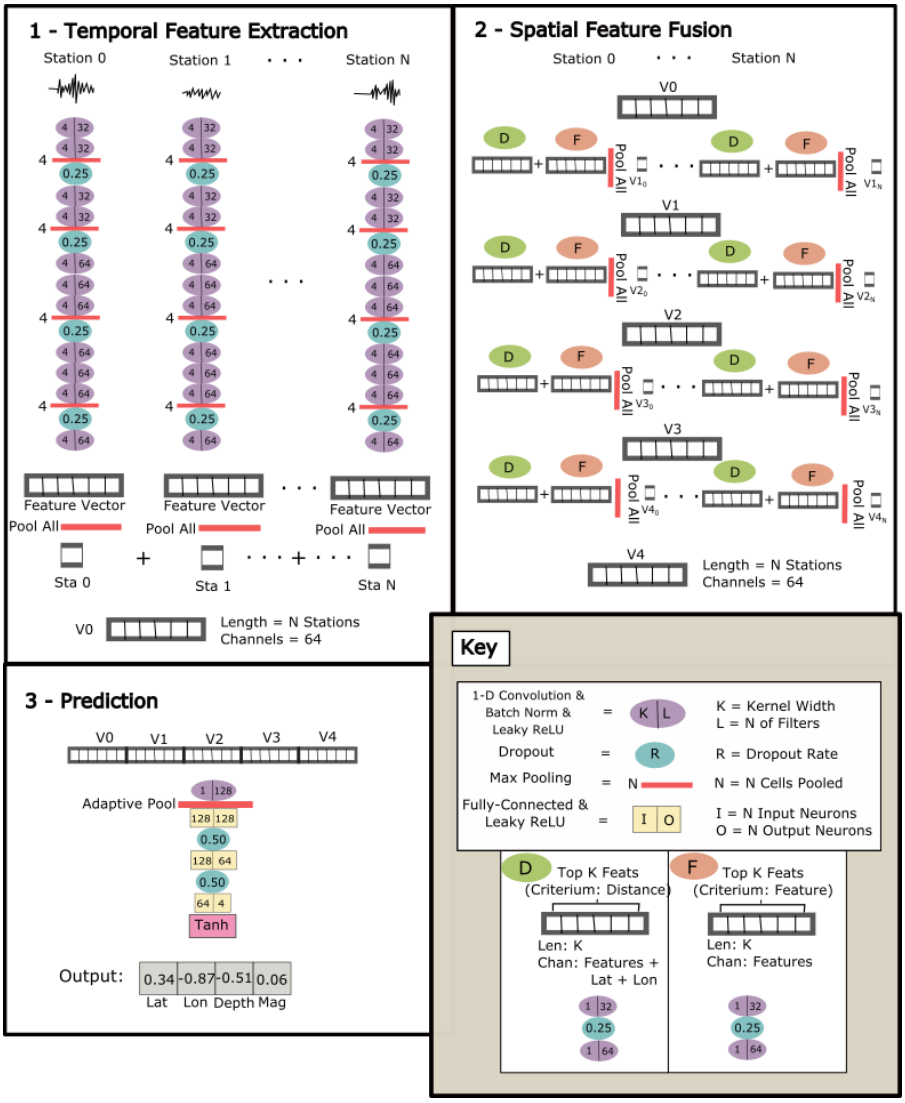
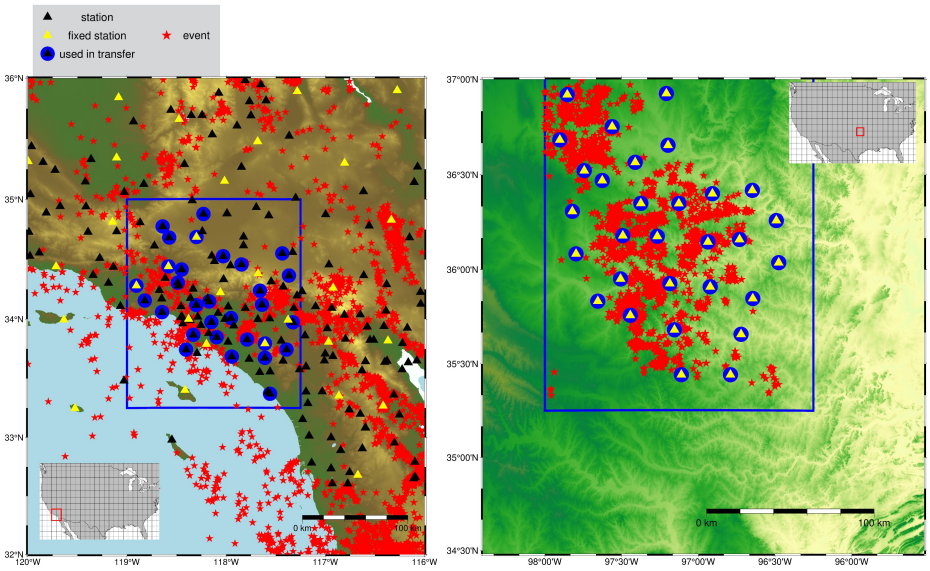


Figure 3: Overview of STGCN. STGCN includes three components, following the framework outlined in Figure 1.

3 Experiments and Results

In this section, the data, experiment settings, and results are discussed. We evaluate STGCN with three major experiments: (1) performance on two datasets compared to GCN and CNN baselines, (2) stability analysis of STGCN with various settings, and (3) the transferrability of STGCN to regions outside of the training domain.

3.1 Data Description



(a) Southern California (b) Oklahoma

Figure 4: Maps of the two target regions used in this study: (a) Southern California and (b) Oklahoma. The distribution of all seismic stations (black triangles) and earthquakes (red stars) are shown. The areas used selected the transferability study are contained within the blue squares. In the map of Southern California, the 30 stations selected for fixed input testing are yellow triangles, and the 30 stations selected for the transferability study are surrounded by a blue circle.

Consistent with target regions in the GCN baseline (van den Ende & Ampuero, 2020) and the FCN baseline (X. Zhang et al., 2020), we correspondingly collected earthquake datasets from Southern California and Oklahoma. The former data including station inventory, earthquake catalogue and waveforms are downloaded from the Southern California Seismic Network (SCSN) and Southern California Earthquake Data Center (SCEDC) (Hutton et al., 2010) from January 2000 to June 2019 and accessed using ObsPy (Beyreuther et al., 2010). STGCN makes predictions by outputting values between -1 and 1. Thus we constrain our the labels of events to fit within a normalized range. We limit stations and

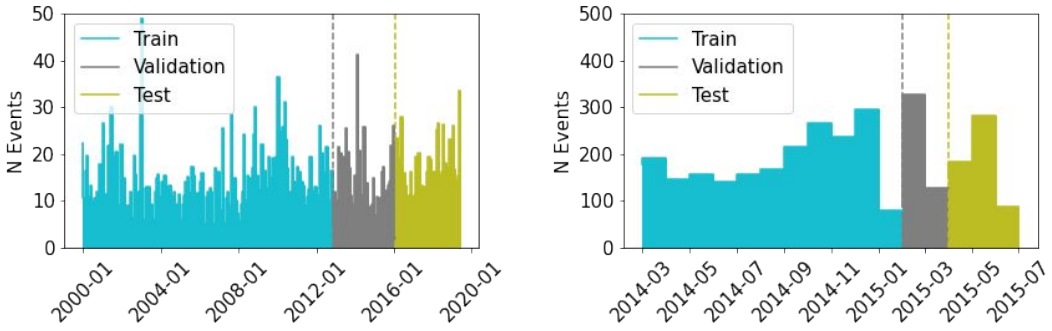
events to a geographic subset from 32° to 36° latitude, and from −120° to −116° longitude (van den Ende & Ampuero, 2020). We select events from a depth range of 0-30 km and a magnitude range of $2.5 < M < 6$. The final dataset contains $2,209$ events recorded by 185 broadband seismic stations. On average, 48 seismic stations are functional for all events. The maximum number of functional seismic stations we can download raw waveforms from is 142. The spatial distribution of events and stations is illustrated in Figure 4. After removing the instrument response, the signals are bandpass filtered from $1-8$ Hz. In the second target region, we collect induced earthquake dataset in Oklahoma from March 2014 to July 2015 (Nanometrics Seismological Instruments, 2013). We limit the dataset to events between 34.482° to 37° latitude, and from −98.405° to −95.527° longitude with depths from 0-12 km (X. Zhang et al., 2020). Magnitudes range from $1.5 < M < 4$. The instrument response is removed, and waveforms are bandpass filtered from $1 - 8$ Hz. The final dataset contains $3,456$ events recorded from 30 stations.

An arbitrary scaling factor of $1e7$ is multiplied across both datasets to raise the extremely small amplitudes to an acceptable range without eliminating magnitude information. Each recording contains 200 sec 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. We use a sliding window to handle the uncertainty of the arrival time that would occur in practical use by cropping shorter time segments from longer raw waveforms at different positions in time. Thus, the actual arrival signal can locate at different time steps and the model will learn to extract proper representation from raw seismic waveforms that have different arrival times during training. In the end, we use a sliding window with a length of 100 sec and a stride of 5 sec to create ten 100 sec samples from each 200 sec recording. Each sample is associated with a label containing latitude, longitude, depth and magnitude values normalized from -1 to 1.

One advantage of our GCN over CNN's or GCN's with fixed edges 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 STGCN and the GCN baseline using Southern California data with dynamic inputs, in which functioning stations are randomly selected for each event. However, to make a fair comparison between STGCN and the FCN baseline, the same stations must occupy the same position in each input. Using the Southern California and Oklahoma datasets, we train STGCN as well as both baselines on thirty fixed stations to compare the performance of all methods. The GCN models can be adaptively trained to make predictions given any number of input stations. If the number of functioning stations is less than the target number of stations for any given event, the input is padded with zeroed channels and the coordinates of the missing stations are set to $(-1, -1)$. For the two datasets, events are omitted where < 25 stations are functioning. In the Southern California dataset where phase reports from SCEDC are available, only events with > 5 stations recording available P and/or S phases are kept, considering the sparse coverage of the stations in a large region. Overall, each event in the Southern California dataset is detected by an average of 31 stations.

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 λ 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 dataset into testing and training data, and reserve 20% of the training data for validation. The datasets are not randomly shuffled, but rather separated by time in which training data precedes 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



(a) Southern California (b) Oklahoma

Figure 5: 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.

used to train a model to detect more recent events on a network where station configuration and seismic characteristics may evolve over time. Figure 5 shows the monthly event frequency distribution in the training and testing dataset.

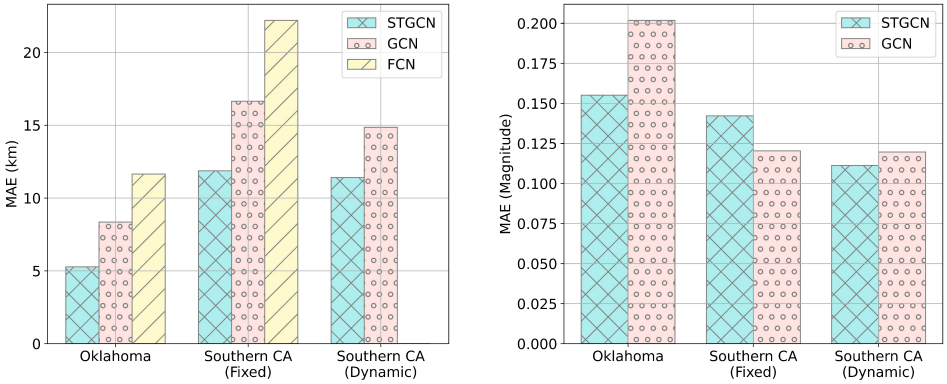
When testing transferability, models are tuned using a learning rate of $3e{-}5$ for 2,000 epochs with an early stopping cutoff of 100 epochs without validation improvement. All weights in the model were permitted to retrain.

3.3 Performance Comparison

To evaluate our developed framework, we compare the testing mean absolute error (MAE) of our proposed model (referenced as STGCN) with the baseline model by van den Ende and Ampuero (2020) (referenced as GCN) when applied to 100 randomly-selected stations from the Southern California dataset. MAE is calculated as:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}|, \tag{3}$$

where $\widehat{y_i}$ is the model's prediction, y_i is the true value, and n is the total number of predictions. In graph convolution, seven edges ($K{=}7$) were generated between the each sta-



(a) MAE of Location Prediction

(b) MAE of Magnitude Prediction

Figure 6: (a) MAE of each tested model where the location error is measured in km. 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 dataset with 30 fixed stations, the Southern California dataset with 30 fixed stations, and the Southern California dataset with 100 dynamically selected stations.

tion and the most similar nodes. The number of edges K is a tunable parameter, the impact of which we evaluate in the following section. Both models make predictions normalized between -1 and 1. The values are first reverted from the normalized scalars to degrees of latitude and longitude, kilometers of depth, and magnitude values. For distance error calculations, degrees of latitude and longitude are converted to kilometers using conversions of 110 km/degree and 92 km/degree, respectively. The previous analysis examines the performance of STGCN when applied to dynamically selected stations from a large network. To further demonstrate STGCN's capabilities, we extend our tests to two different datasets (Oklahoma and Southern California), tested in comparison to two baselines (a GCN baseline (van den Ende & Ampuero, 2020) and a FCN baseline (X. Zhang et al., 2020)). As the FCN baseline requires a fixed input consisting of 30 sta-

Latitude	MAE (km)	MSE (10^2 km)	R^2
STGCN	$\mathbf{7.788 \pm 10.849}$	$\mathbf{1.783 \pm 12.697}$	**0.977**
GCN	10.201 ± 11.791	2.431 ± 12.438	0.969

Longitude	MAE (km)	MSE (10^2 km)	R^2
STGCN	$\mathbf{7.563 \pm 9.408}$	$\mathbf{1.457 \pm 8.209}$	**0.982**
GCN	10.095 ± 12.086	2.480 ± 11.865	0.970

Depth	MAE (km)	MSE (10^2 km)	R^2
STGCN	$\mathbf{3.486 \pm 2.958}$	$\mathbf{0.209 \pm 0.377}$	**0.256**
GCN	3.837 ± 3.166	0.247 ± 0.399	0.120

Magnitude	MAE	MSE	R^2
STGCN	$\mathbf{0.111 \pm 0.115}$	$\mathbf{0.0257 \pm 0.0824}$	**0.837**
GCN	0.120 ± 0.126	0.0302 ± 0.105	0.807

Table 1: Performance of the STGCN model proposed in this paper and the GCN baseline when applied to the Southern California dataset with dynamic inputs. MAE refers to the mean absolute error (Equation 3) and MSE refers to the mean squared error (Equation 4), where a lower value indicates less error. The R^2 value (Equation 5) 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.

tions, the 30 stations active for the greatest number of events in the Southern California dataset were used as the inputs for all samples. The selected stations are highlighted

in Figure 4. As the Oklahoma network consists of only 30 stations, all 30 stations were used. The performance overview is shown in Figure 6, which clearly shows that our proposed model achieve higher localization accuracy than baselines for all datasets. The FCN baseline doesn't support magnitude prediction, and two GCN-based models achieve comparable performance.

STGCN makes predictions with an average of 8.3 km less location error, a 49% improvement across all tested datasets when compared to the FCN baseline, and has the ability to predict magnitude as well as location. Across all datasets, STGCN makes predictions with an average of 3.8 km less location error than the GCN baseline, a 28% improvement. While magnitude does not improve for every individual dataset, STGCN shows an overall improvement in magnitude when all tested datasets are considered.

The detailed evaluation results of Southern California dataset with dynamic seismic stations are shown in Table 1. In terms of MAE, our GCN model outperforms the GCN baseline for all predictions (latitude, longitude, depth, magnitude), with most improvement achieved in latitude and longitude prediction. In addition to MAE, the mean squared error and R^2 values are displayed. Mean squared error is calculated as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2, \tag{4}$$

where $\widehat{y_i}$ is the model's prediction, y_i is the true value, and n is the total number of predictions. R^2 is calculated as:

$$R^2 = 1 - \sum_{i=1}^{n}\frac{(\widehat{y_i} - y_i)^2}{(y_i - \bar{y})^2}, \tag{5}$$

where $\widehat{y_i}$ 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. STGCN demonstrates better performance with both measures of accuracy and is more consistent (smaller standard deviations in prediction accuracy). However, both STGCN and the GCN baseline demonstrate exceedingly low R^2 values for depth prediction. In terms of magnitude, STGCN and GCN perform comparably when all measures of accuracy are considered.

Latitude	MAE (km)	MSE (10^2 km)	R^2
STGCN	$\mathbf{4.487 \pm 9.264}$	$\mathbf{1.060 \pm 9.484}$	**0.947**
GCN	7.166 ± 12.414	2.055 ± 14.820	0.897
FCN	9.219 ± 16.418	3.545 ± 23.070	0.822

Longitude	MAE (km)	MSE (10^2 km)	R^2
STGCN	$\mathbf{4.151 \pm 7.035}$	$\mathbf{0.667 \pm 5.502}$	**0.937**
GCN	5.934 ± 8.144	1.015 ± 5.547	0.904
FCN	9.308 ± 11.883	2.279 ± 8.244	0.785

Depth	MAE (km)	MSE (10^2 km)	R^2
STGCN	1.760 ± 1.473	0.053 ± 0.083	0.026
GCN	$\mathbf{1.701 \pm 1.423}$	$\mathbf{0.049 \pm 0.078}$	**0.090**
FCN	1.865 ± 1.546	0.059 ± 0.084	-0.086

Magnitude	MAE	MSE	R^2
STGCN	$\mathbf{0.154 \pm 0.123}$	$\mathbf{0.0388 \pm 0.0668}$	**0.787**
GCN	0.195 ± 0.142	0.0582 ± 0.0831	0.681

Table 2: Performance of STGCN, GCN and FCN baselines when applied to the Oklahoma dataset with fixed inputs. MAE refers to the mean absolute error (Equation 3) and MSE refers to the mean squared error (Equation 4), where a lower value indicates less error. The R^2 value (Equation 5) 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.

Latitude	MAE (km)	MSE (10^2 km)	R^2
STGCN	$\mathbf{8.022 \pm 9.664}$	$\mathbf{1.577 \pm 9.297}$	**0.970**
GCN	11.263 ± 11.696	2.637 ± 8.010	0.949
FCN	14.415 ± 21.827	6.842 ± 34.697	0.869

Longitude	MAE (km)	MSE (10^2 km)	R^2
STGCN	$\mathbf{7.840 \pm 11.645}$	$\mathbf{1.971 \pm 19.095}$	**0.972**
GCN	11.485 ± 12.199	2.807 ± 10.252	0.960
FCN	16.369 ± 24.872	8.865 ± 47.323	0.874

Depth	MAE (km)	MSE (10^2 km)	R^2
STGCN	$\mathbf{3.869 \pm 3.380}$	$\mathbf{0.264 \pm 0.411}$	$\mathbf{-0.016}$
GCN	4.264 ± 3.384	0.296 ± 0.403	-0.141
FCN	4.105 ± 3.324	0.279 ± 0.431	-0.074

Magnitude	MAE	MSE	R^2
STGCN	0.142 ± 0.117	0.0340 ± 0.0624	0.796
GCN	$\mathbf{0.120 \pm 0.118}$	$\mathbf{0.0283 \pm 0.0880}$	**0.830**

Table 3: Performance of STGCN, GCN and FCN baselines when applied to the Southern California dataset with fixed inputs. MAE refers to the mean absolute error (Equation 3) and MSE refers to the mean squared error (Equation 4), where a lower value indicates less error. The R^2 value (Equation 5) 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.

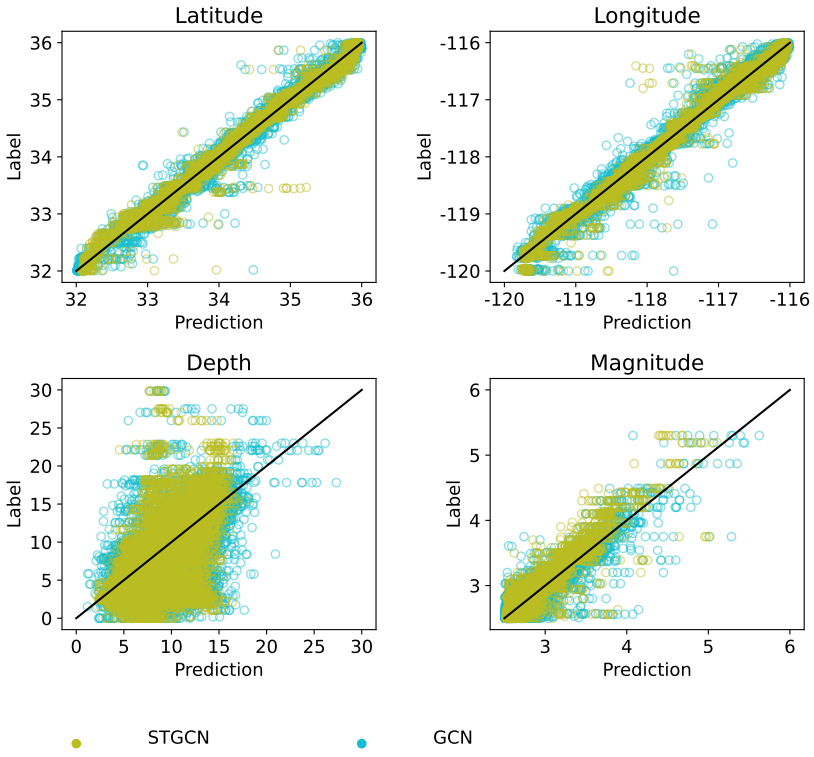


Figure 7: Testing comparison on 100 dynamically selected stations from the Southern California dataset. "STGCN" and "GCN" denote the performance of our framework and the published baseline, respectively. In the scatter plot, each point represents an event, and a position on the diagonal line corresponds 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

In both datasets with fixed seismic stations, the proposed model shows significant improvement over both baselines in terms of location error, with most improvement arising from latitude and longitude predictions. This improvement is supported by several performance metrics (Table 2 and Table 3). Overall location error is 5.28 km for the Oklahoma dataset and 11.87 km for the Southern California dataset. The higher loss for the Southern California dataset 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

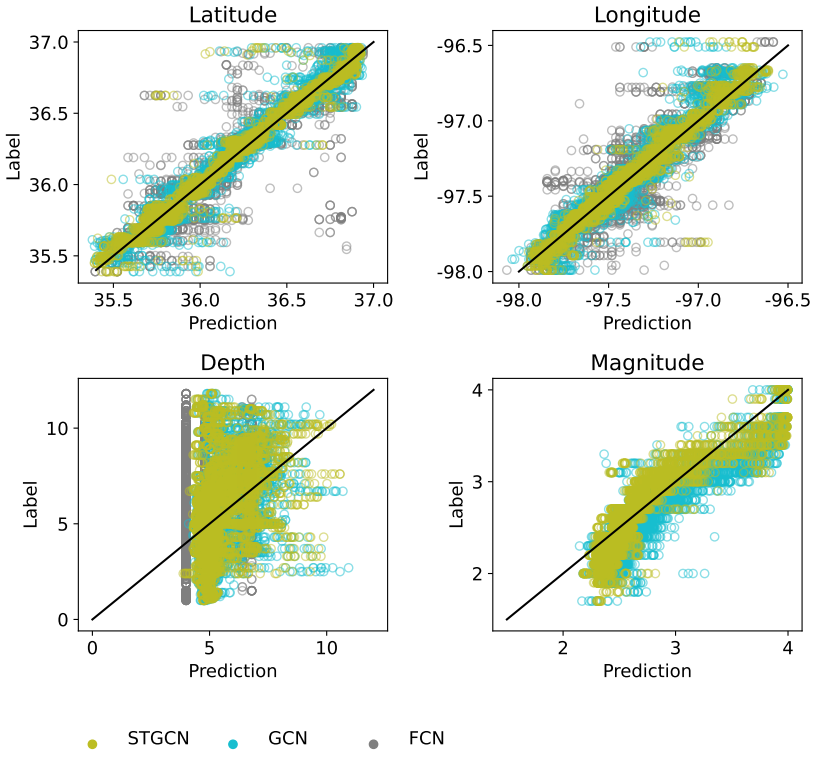


Figure 8: Testing comparison on 30 fixed stations from the Oklahoma dataset. "STGCN", "GCN", and "FCN" denote the performance of our framework, the published GCN baseline, and the published FCN baseline, respectively. In the scatter plot, each point represents an event, and a position on the diagonal 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. Magnitude is omitted for the FCN, as this model makes only location predictions

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.

Figure 7, 8 and 9 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 mapping to latitude and

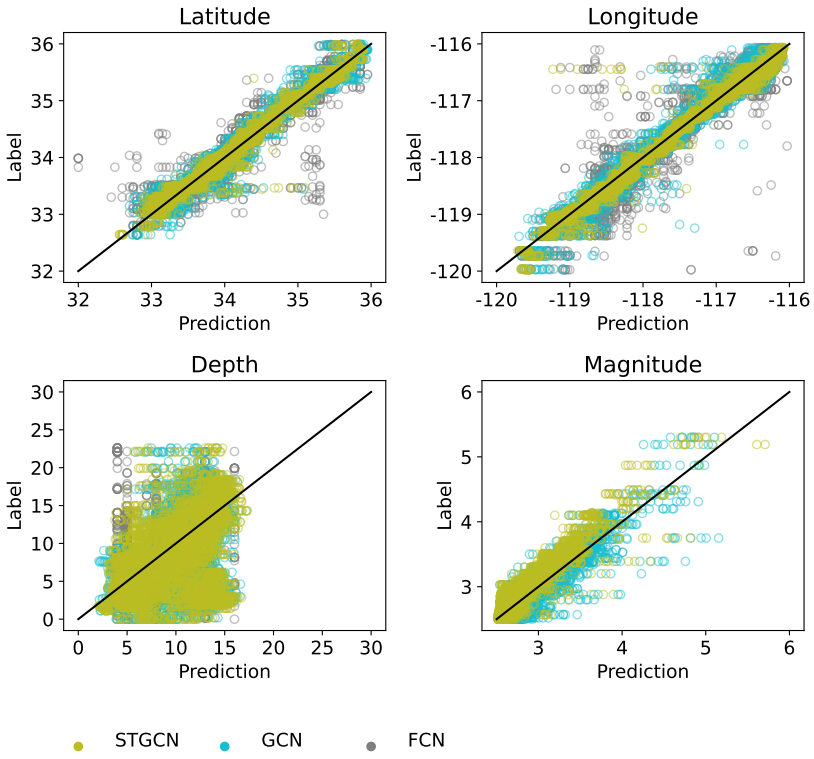


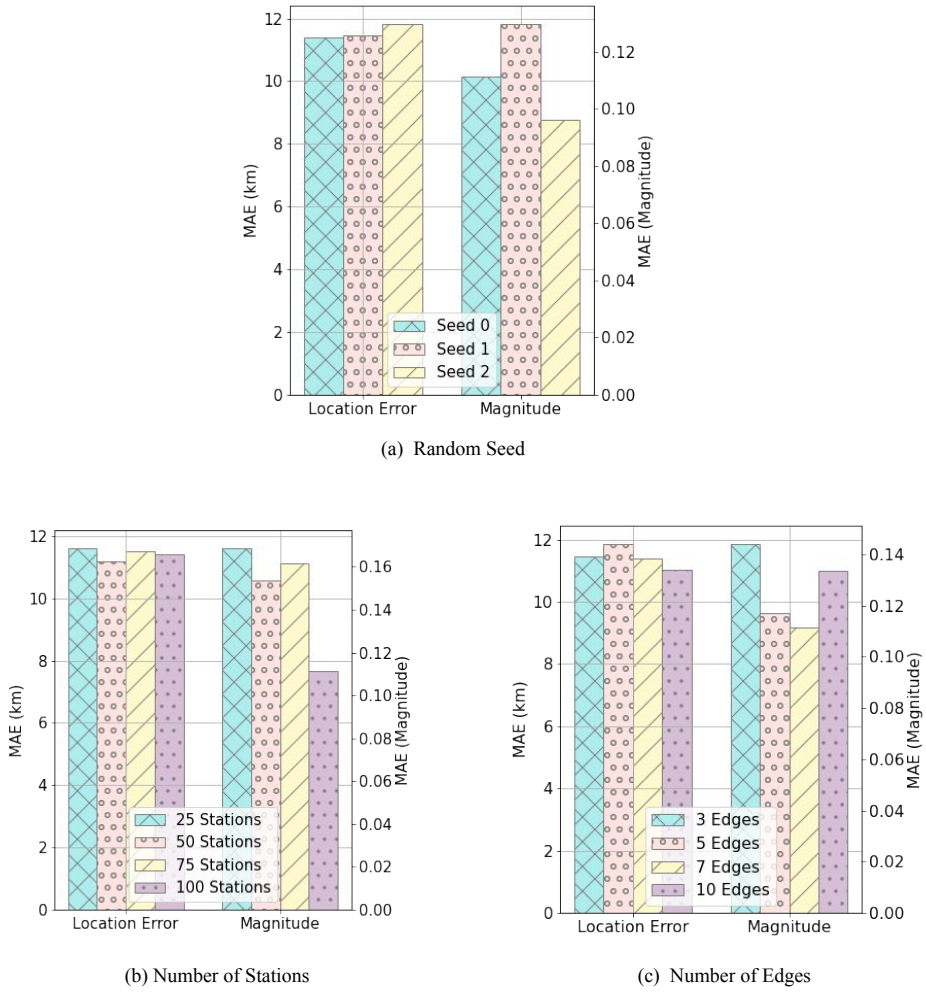
Figure 9: Testing comparison on 30 fixed stations from the Southern California dataset. "SPCGN", "GCN", and "FCN" denote the performance of our framework, the published GCN baseline, and the published FCN baseline, respectively. In the scatter plot, each point represents an event, and a position on the 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. Magnitude is omitted for the FCN, as this model makes only location predictions.

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

While our proposed model predicts magnitude with less error than the GCN baseline on the Oklahoma dataset, the model has greater magnitude errors when applied to the Southern California dataset (Table 1). All models perform extremely poorly when predicting depth. Therefore STGCN does not improve depth or magnitude prediction,

where it remains comparable to the baseline models. However, STGCN substantially improves latitude and longitude predictions, resulting in higher quality location estimations .

3.4 Stability Analysis



(a) Random Seed

(b) Number of Stations

(c) Number of Edges

Figure 10: Stability analysis permuting (a) the random seed used to select stations for the model input, (b) the number of stations used for prediction, and (c) the number of edges used to connect nodes during graph convolution.

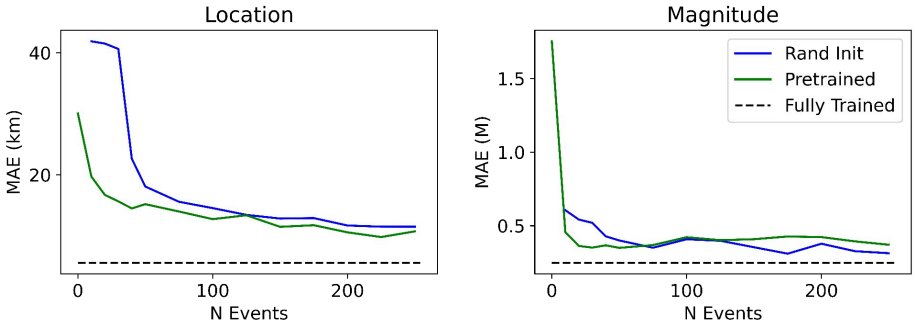
There are three critical hyper-parameters in STGCN: the number of neighbors considered for edge generation, the total amount of observed stations, and the random selection of seismic stations when creating datasets. We use the Southern California Dasaset to vary these hyperparameters in order to assess the stability of STGCN. The results of the paramater permutation are shown in Figure 10.

For each prediction, a random subset of functional stations were selected. We permute the random seed during the selection of 100 stations, making predictions using 7 edges. We find that the random subsets return similar results for all predictions except for magnitude, which shows a higher degree of variation. With the exception of magnitude, prediction accuracy remains similar when 25, 50, 75, or 100 stations are used. Magnitude prediction improves substantially when 100 stations are selected. A similar pattern is observed in the edge stability, where the number of generated edges has the greatest influence on magnitude performance. Overall, the model appears to be generally stable, with magnitude demonstrating the greatest sensitivity to hyperparameter tuning.

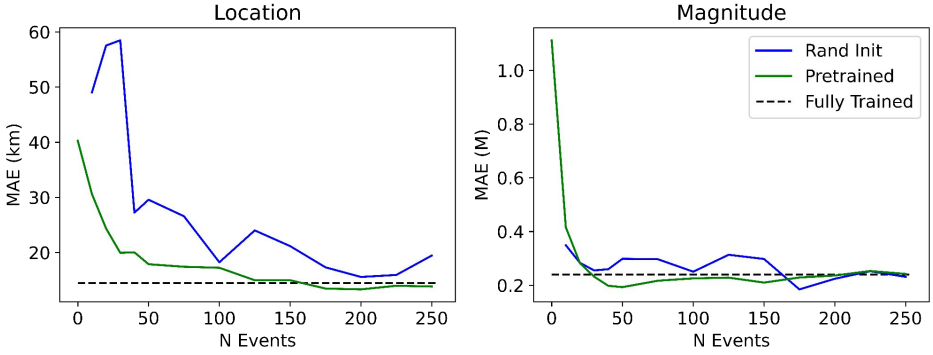
3.5 Transferability

In many real use cases, a studied network may have a small or nonexistent catalogue of events with which to train a predictive model. It is therefore useful to test the effectiveness of a pretrained model when applied to events in an unseen region. Figure 11 shows the performance of a model trained on the Southern California dataset and tested on the Oklahoma dataset and vice versa when tuned with samples ranging from 0-250. Regardless of the number of training samples, the validation and testing data remained the same for each training and testing. The performance of a tuned model is compared to the performance of a randomly-initialized model trained with the same number of samples to examine whether or not pretraining is beneficial.

Two equal-area regions were selected from the Oklahoma and Southern California datasets. From the Southern California dataset, 30 fixed stations were selected which



(a) Train: California Test: Oklahoma



(b) Train: Oklahoma Test: California

Figure 11: Transferability of (a) model trained on Southern California data and tested on Oklahoma data and (b) model trained on Oklahoma data and tested on Southern California data. The plots show the prediction error of the pretrained model (green) and randomly initialized model (blue) when a range of 0 (no retraining) to 250 events are used for training. The panels to the left show the euclidean location errors between the predicted and true hypocenter measured in km, and the panels to the right show the magnitude errors. The dashed line corresponds to the performance when randomly initialized weights are trained with all available training data from the region.

most closely resemble the distribution of the Oklahoma dataset with respect to minimum, maximum, and mean distance between stations. The Oklahoma dataset consists of a much larger training dataset comprising 2,025 events while the Southern California dataset contains only 254 events. Overall, when pretrained models are applied to a new region with no tuning, the models perform poorly. However, the pretrained models nonetheless predict location with greater accuracy than the models trained from random weights. The benefits of transfer learning are most marked for very small datasets - after approximately 100 events are used for training, using pretrained models has less of an advantage over randomly initialized weights.

While the ranges of area and depth are equal between the two datasets, the magnitudes of the Southern California dataset are normalized from 2.5-6 while the magnitudes of the Oklahoma dataset are normalized between 1.5-4. The tuned models were able to adapt to the change in normalization given only ten events.

4 Discussion

Our GCN 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, and station subsetting. As STGCN 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 X. Zhang 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 com-

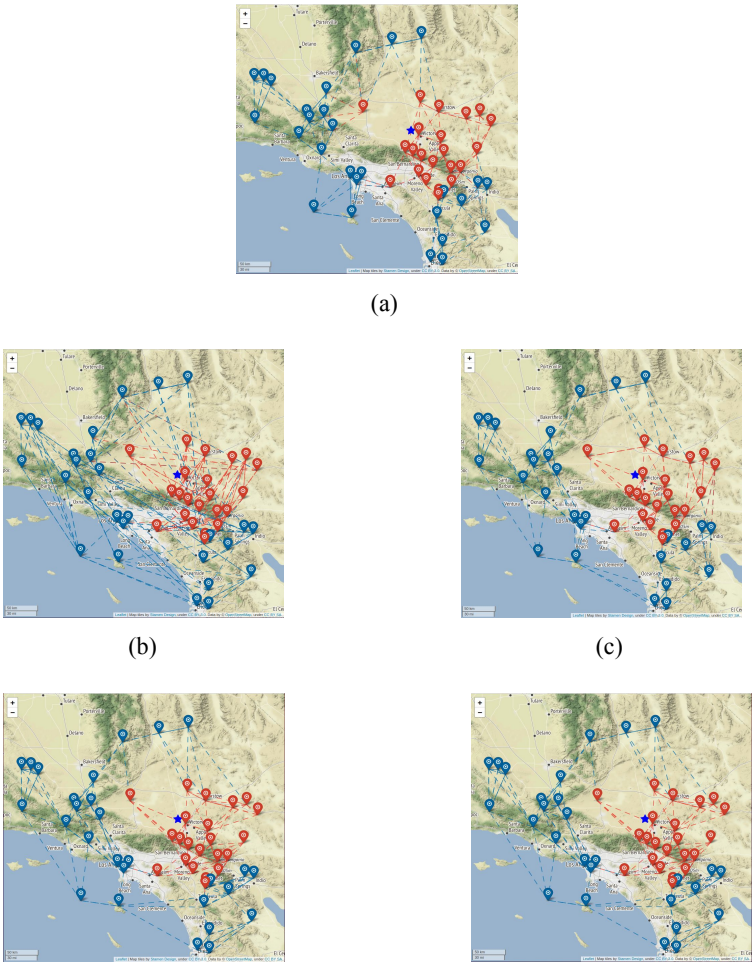


Figure 12: Graphs constructed by different layers of the graph neural network, (a) graph convolution layer based on locations of seismic stations (b) 1st, (c) 2nd, (d) 3rd and (e) 4th graph convolution layer based on the similarity of extracted features of seismic stations. Stations that detected an event in the catalogue are denoted by red symbols, while stations that did not record the event are shown in blue. Red and blue edges are generated for updating features of red and blue stations, respectively. Star represents the event location. The information from stations with the event signal are clustered in deeper layers.

prises approximately 27 million parameters while our GCN with scalar predictions comprises fewer than 0.24 million parameters.

The baseline GCN (van den Ende & Ampuero, 2020) implements edgeless graph convolution (i.e. station-by-station convolutions with global pooling) while GCN model developed in this paper implements convolution and pooling over dynamically-generated edges. Figure 12 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 different clusters than the 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. Moreover, the generated graphs from the 3rd and 4th graph convolution layer based on the extracted feature similarity converge to the same structure, indicating that the number of graph convolutional layers is large enough to connect informative seismic stations together. If we only consider the geographic proximity, one seismic station recording the earthquake will connect to seismic stations without signal records only. It denotes that the feature similarity is a proper complement of geographic proximity during aggregating features from different seismic stations.

After training in one region, STGCN does not transfer well to other regions without retraining. This indicates that the models are encoding site-specific information such as velocity structure or types of seismicity (i.e. anthropogenically induced earthquakes in the Oklahoma dataset) as well as different magnitude range which affect predictions in a different region. 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 dataset is available. However, best results are achieved when a model is trained for the region of implementation using a catalogue of several hundred events.

While STGCN makes improvements in functionality and location error with respect to the baseline models, the proposed framework faces challenges. Substantial improvements have been made in the prediction of latitude and longitude, and an overall improvement in magnitude is observed. However, magnitude does not improve in every dataset, and depth predictions are highly inaccurate for all models. Accurate depth estimation also poses a challenge for classical inversion methods (Zonno & Kind, 1984; Billings et al., 1994; M. Zhang et al., 2014). As the machine learning models tested in this work are trained in a purely supervised manner, the learned predictions are fundamentally limited by the accuracy of the training data. Errors in training data are likely to be a leading driver in model error in earthquake characterization, as systematically demonstrated by X. Zhang et al. (2020) by observing the effects of induced label noise on models trained with synthetic data.

We perform a similar test, training our model using synthetic data generated with Pyrocko (Developers, n.d.). For each sample, receivers were placed randomly along a flat surface, and a double-couple source with a random strike, dip, rake, magnitude, and location was seeded. Both stations and events were placed with uniform probability in a 4° latitude by 4° longitude area (between 7° and 11° in the simulated volume). For events, depth was constrained from $0.7 - 10$ km, strike from $0 - 180^\circ$, dip from $0 - 90^\circ$, and rake from $0 - 360^\circ$ with a magnitude range of $2.5 < M < 6$. Using a precalculated Green's Function (`https://greens-mill.pyrocko.org/iceland_reg_v2-453e36`), wave propagation was simulated through a 1-D velocity structure and recorded by the stations. As the simulated waveforms have a sampling frequency of 2 Hz, the samples were decimated to 20.24 Hz to be compatible with our model. We layered random noise over the synthetic signals to make prediction more challenging. Non-detecting stations which record only random noise without earthquake signal are also included in the input files. For smaller events ($2.5 < M < 4$), $0 - 23\%$ of receivers were non-detecting, and for larger events ($4 < M < 6$), $0 - 13\%$ of receivers were non-detecting. A total of 30 receivers were included in each sample.

As demonstrated by Figure 13, when label error is eliminated, depth predictions dramatically improve. This indicates that the inability to correctly predict depth is a reflection of data quality rather than shortcomings within the model design. Note that the synthetic experiment was designed for method validation and may not applicable to our field data due to different aspects (e.g., waveform frequency, velocity structure, etc). Future improvement in depth prediction must therefore be solved by accounting for incorrect depth labels. One solution may be to train using higher-quality datasets in which meticulous relocation has been implemented. However, reliance on large quantities of relocated sample events significantly restricts the areas in which supervised models can operate. Another solution may be to avoid purely supervised methods, implementing solutions which combine physics-based constraints with data-driven learning to overcome inaccuracy in depth labels.

Another limitation that STGCN shares with the baselines is the ability to make predictions only within a certain range of area, depth, and magnitude, which is also the limitation of all machine-learning-based frameworks. 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, STGCN is more limited than non-machine learning methods with regard to magnitude prediction. Magnitudes falling above or below the training range cannot be predicted by STGCN or the deep learning baselines. The limited range of predictions adversely impacts the usefulness of the deep learning methods for applications such as Earthquake Early Warning, where magnitude saturation must be avoided. The limitations posed by fixed prediction ranges are made less severe by STGCN's ability to be tuned to new ranges with small amounts of training data. However, the fixed prediction ranges nonetheless represent a weakness in our framework.

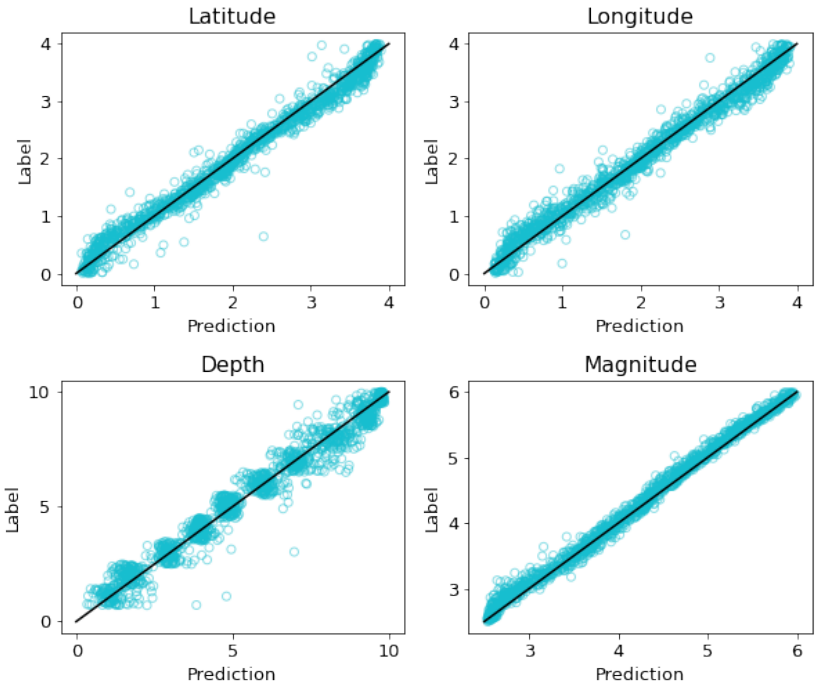


Figure 13: Testing performance of STGCN on synthetic data from 30 randomly-placed stations. In the scatter plot, each point represents an event, and a position on the diagonal line corresponds to perfect agreement between the predicted value (x-axis) and the true value (y-axis). Latitude and longitude values are displayed in degrees and depth values are displayed in kilometers.

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 STGCN outperforms both the FCN and GCN baselines, yields stable results using a range of hyperparameters, and can be applied to new datasets after retraining with a small number of events. One of the major advantages of our framework compared with other deep learning source characterization networks is that STGCN does not require static input or a manually gen-

erated 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 model capacity to predict depth, a problem which synthetic tests reveal to be primarily caused by label error. 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 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 catalogues. 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 catalogue.

6 Open Research

Waveform data used in this study were downloaded from the Incorporated Research Institutions for Seismology (`http://ds.iris.edu/ds/nodes/dmc`) and the Southern California Earthquake Data Center (`https://scedc.caltech.edu/data/waveform.html`). The maps in our paper were made using Generic Mapping Tools (Wessel et al., 2013) and Python. The forward modelling was performed with Pyrocko (Developers, n.d.) using the $iceland_reg_v2$ Green's function available at `https://greens-mill.pyrocko.org/iceland_reg_v2-453e36`.

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

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).

Beskardes, G. D., Hole, J. A., Wang, K., Michaelides, M., & Wu, Q. (2018). A comparison of earthquake back-projection imaging methods for dense local arrays. *Geophysical Journal International*, *212*(3), 1986-2002.

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.

Billings, S., Kennett, B., & Sambridge, M. (1994). Hypocentre location: genetic algorithms incorporating problem-specific information. *Geophysical Journal International*, *118*(3), 693–706.

Developers, T. P. (n.d.). *Pyrocko: A versatile seismology toolkit for Python.* Retrieved 2018-02-23, from `http://pyrocko.org` doi: 10.5880/GFZ.2.1.2017.001

Gajewski, D., Anikiev, D., Kashtan, B., & Tessmer, E. (2007). Localization of seismic events by diffraction stacking. In *Seg technical program expanded abstracts 2007* (p. 1287-1291).

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.

Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*(4), 1–21.

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.

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*, 510 – 516.

Li, L., Tan, J., Schwarz, B., Stanek, F., Poiata, N., Shi, P., ... Gajewski, D. (2020). Recent advances and challenges of waveform-based seismic location methods at multiple scales. *Reviews of Geophysics*, e2019RG000667.

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.

Li, Z., & van der Baan, M. (2016). Microseismic event localization by acoustic time reversal extrapolation. *Geophysics*, *81*(3), KS123-KS134.

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.

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*, 1 – 14.

Mousavi, S. M., & Beroza, G. C. (2020b). A machine-learning approach for

earthquake magnitude estimation. *Geophysical Research Letters*, *47*(1), e2019GL085976.

Münchmeyer, J., Bindi, D., Leser, U., & Tilmann, F. (2020, Dec). The transformer earthquake alerting model: a new versatile approach to earthquake early warning. *Geophysical Journal International*, *225*(1), 646–656. Retrieved from `http://dx.doi.org/10.1093/gji/ggaa609` doi: 10.1093/gji/ggaa609

Münchmeyer, J., Bindi, D., Leser, U., & Tilmann, F. (2021, Apr). Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network. *Geophysical Journal International*, *226*(2), 1086–1104. Retrieved from `http://dx.doi.org/10.1093/gji/ggab139` doi: 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/` doi: 10.7914/SN/NX

Perol, T., Gharbi, M., & Denolle, M. (2018). Convolutional neural network for earthquake detection and location. *Science Advances*, *4*, e1700578.

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.

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.

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.

Tiira, T. (1999). Detecting teleseismic events using artificial neural networks. *Comput. Geosci.*, *25*, 929 – 938.

van den Ende, M. P., & Ampuero, J.-P. (2020). Automated seismic source characterisation using deep graph neural networks. *Geophysical Research Letters*, e2020GL088690.

Wang, J., & Teng, T. (1995). Artificial neural network-based seismic detector. *Bull. Seismol. Soc. Am.*, *85*, 308 – 319.

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 (tog)*, *38*(5), 1–12.

Wessel, P., Smith, W., Scharroo, R., Luis, J., & Wobbe, F. (2013). Generic mapping tools: Improved version released. *EOS*, *94*, 409-410 – 41.

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.

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.

Zhang, M., Tian, D., & Wen, L. (2014). A new method for earthquake depth determination: stacking multiple-station autocorrelograms. *Geophysical Journal International*, *197*(2), 1107-1116.

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 Report*, *10*.

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.

Zhang, Z., Rector, J. W., & Nava, M. J. (2017). Simultaneous inversion of multiple microseismic data for event locations and velocity model with bayesian

652 inference. *Geophysics*, *82*(3), KS27-KS39.

653 Zhebel, O., & Eisner, L. (2015). Simultaneous microseismic event localization and
654 source mechanism determination. *Geophysics*, *80*(1), KS1-KS9.

655 Zhu, W., & Beroza, G. C. (2019). Phasenet: a deep-neural-network-based seismic
656 arrival-time picking method. *Geophysical Journal International*, *216*(1), 261–
657 273.

658 Zhu, W., Mousavi, S. M., & Beroza, G. C. (2019). Seismic signal denoising and
659 decomposition using deep neural networks. *IEEE Transactions on Geoscience
660 and Remote Sensing*, *57*(11), 9476–9488.

661 Zonno, G., & Kind, R. (1984). Depth determination of north italian earthquakes
662 using grafenberg data. *Bulletin of the Seismological Society of America*, *74*(5),
663 1645–1659.