

Microseismic monitoring with the quake neural operator

Received: 26 January 2025

Hongyu Sun ^{1,2} 

Accepted: 26 May 2026

Published online: 06 July 2026

 Check for updates

Accurate monitoring of small-scale seismic events is essential for seismological studies. Traditional techniques rely on manual or automated seismic phase picking, which often leads to inaccuracies in microseismic monitoring due to unclear phase onsets. Here we introduce the Quake Neural Operator (QNO), a deep learning algorithm that builds earthquake catalogs directly from continuous data without explicit phase picking. As a multi-task operator, QNO utilizes classification and regression to detect and locate events across arbitrary seismic network geometries. We show that QNO successfully characterizes events where state-of-the-art phase picking fails. Applying QNO to the Geysers geothermal field, we identify nearly an order of magnitude more seismic detections than reported in routine catalogs. These results are validated through comparison with the Phase Neural Operator and visual inspection. QNO holds the potential to reveal undetected seismic activity, enhancing our understanding of subsurface processes critical to both natural phenomena and industry applications.

Microseismic monitoring offers valuable insights into subsurface processes, enhancing our understanding of fault mechanics, rock behavior, and fluid dynamics^{1–3}. Beyond the study of natural earthquakes, the monitoring of seismicity induced by human activities has critical applications in geothermal energy development, mining, and carbon capture and storage^{4–6}. While microearthquakes are typically too weak to be felt, their small vibrations are recorded in continuous seismic data streams, and are typically identified with a sequential workflow: seismic phases are picked from continuous data, picks are associated with seismic events, and event locations are determined using associated arrival times⁷. This process, however, often relies on manual or automated phase picking techniques, which are time-consuming and susceptible to human bias, particularly when handling large datasets with low signal-to-noise ratios (SNRs).

The advancement of deep learning has created opportunities to automate and enhance seismic monitoring^{7–10}. Building on the sequential workflow of seismic monitoring, deep learning techniques have been used to replace steps in this procedure^{11–19}. As an example of a deep learning-based phase picker, EQTransformer achieved more than a twofold increase in the number of detected events over the same time period, using only 18 stations, compared to a traditional

catalog built from manually picked phases using 57 stations¹³. In addition to enhanced detection capability and efficiency, deep learning models also achieve arrival-time precision comparable to that of human analysts. Specifically, the mean absolute errors (MAEs) in P- and S-phase arrival times between human picks and the Phase Neural Operator (PhaseNO) are 0.05 s and 0.09 s, respectively¹⁹. With a sampling rate of 100 Hz, which is commonly used for regional seismic networks, an error of 0.05 s corresponds to only five samples, which is very small relative to the total waveform duration. Additionally, waveform denoising with neural networks before phase picking may enhance detection performance^{20,21}. Zhou et al.²² perform event detection using full-waveform data before phase picking, with both tasks implemented using deep neural networks. Following phase picking, passive-source locations have been estimated from picked arrival times using machine-learning methods like random forests²³ or graph neural networks^{24,25}.

Recently, an increasing number of seismic monitoring and earthquake catalog construction workflows have been proposed^{7,26,27}. Moving from single-task to multi-task learning, deep neural networks have demonstrated the potential to integrate multiple processing steps into a unified AI framework for seismic monitoring^{28–30}. Other

¹Department of Earth, Environmental and Resource Sciences, The University of Texas at El Paso, El Paso, TX, USA. ²School of Earth Sciences, The Ohio State University, Columbus, OH, USA. ✉e-mail: hongyu-sun@outlook.com

studies have developed customized seismic monitoring workflows by incorporating innovative AI techniques. For example, Tan et al.³¹ proposed an earthquake catalog construction workflow that detects events using waveform-based source imaging³² and AI-based semantic segmentation of 3D source images, followed by phase picking and source location. Alternatively, approaches that combine phase picking for detection with waveform-based source imaging for location have been proposed to automate the entire procedure, using either a classic short-term average/long-term average (STA/LTA) picker³³ or deep-learning-based phase pickers^{34–37}. Nonetheless, the overall detection capability and location accuracy may still be constrained by the initial phase-picking step.

Despite the advancements in seismic monitoring with deep learning, microearthquake monitoring still faces significant challenges due to the typically very low amplitudes of these events compared to the high levels of environmental noise. Many stations record waveforms that lack distinct phase onsets, complicating the process of phase picking. The inherently low SNR of microseismic events not only makes phase picking more difficult but also increases the potential for errors in earthquake location estimates. In many cases, template matching^{38–40} remains necessary to improve the detection of low-amplitude events or weak seismic signals after an initial earthquake catalog is built using either conventional²⁶ or deep learning-based phase pickers^{41,42}. However, template matching is known to have drawbacks, including being time-consuming and reliant on predefined templates⁴³.

In addition to location approaches based on picked arrival times, seismic waveforms of detected events have been directly mapped to earthquake locations using deep learning techniques^{8,44–48}. For instance, Zhang et al.⁴⁹ applied a convolutional neural network to locate induced earthquakes in Oklahoma using a fixed seismic network. In contrast, flexibility with respect to station geometry can be achieved by incorporating station locations into the input of graph neural networks^{50,51}, transformer-based models⁵², 3-D U-Net network^{53,54}, or neural operators⁵⁵. These methods often focus on relatively large earthquakes and may not perform well for microseismic events. Furthermore, these approaches typically assume that the waveforms of seismic events have already been extracted from continuous recordings. As such, they are not designed to operate directly on raw continuous data streams in which events have not yet been detected.

In this work, we introduce the Quake Neural Operator (QNO), a multi-task learning framework for end-to-end microseismic monitoring that directly detects and locates earthquakes from continuous waveform data recorded by seismic networks with arbitrary geometry. Moving beyond the traditional sequential workflow of phase picking, association, and source location, QNO jointly learns earthquake detection and source characterization within a unified framework. Because detection and location are intrinsically linked tasks, QNO exploits shared information between them to improve overall performance. We show that coherent spatio-temporal waveform patterns across multiple stations enable reliable earthquake detection and direct source location even when clear phase onsets are unavailable. By leveraging neural operator architectures (see Supplementary Note 1), QNO naturally accommodates irregular seismic network geometries and varying station configurations. Applications to microseismic monitoring demonstrate that QNO improves the detection of low-magnitude events and enables the construction of more comprehensive earthquake catalogs directly from continuous seismic data without explicit phase picking or phase association.

Results

We begin by briefly summarizing the Quake Neural Operator (QNO), as its overall architecture constitutes a central methodological contribution underlying all subsequent experiments; full technical details are provided in the Methods section.

Quake neural operator

The QNO is a multi-task operator learning model for automatic earthquake detection and location across multiple stations in a seismic network with arbitrary geometry (see Methods). It consists of a classification task and a regression task. The classification task $\mathcal{Q}_{\text{class}}$ learns a probability function $p(x, y, z)$ representing the likelihood of earthquake signals and noise in the spatial domain. For a given sample, this function is evaluated at discrete station locations (x_i, y_i, z_i) , yielding the probabilities $\mathbf{p} = \{\mathbf{p}_i\}_{i=1}^n$, where $i \in \{1, 2, \dots, n\}$ and n is the number of stations in the input. The regression task $\mathcal{Q}_{\text{reg}}$ determines the earthquake location and origin time $(\hat{x}, \hat{y}, \hat{z}, \hat{t})$. QNO operates flexibly with different seismic network configurations (see examples in Supplementary Figs. 1 and 2).

Figure 1 depicts the architecture of the QNO model. The discretized input is represented as a three-dimensional tensor with axes for time, station, and channel. Each input tensor contains six channels: three corresponding to the three-component waveforms and three corresponding to the station locations (x_i, y_i, z_i) . The maximum duration for each input waveform segment is 15 s. The station axis dimension varies depending on the number of input stations sampling the spatial domain of the seismic wavefields, and the order of stations is arbitrary. The operation shared by both classification and regression tasks consists of multiple blocks of operator layers, where each block contains Fourier Neural Operator (FNO) layers⁵⁶ and Graph Neural Operators (GNO)⁵⁷ layers connected and repeated several times. FNO extracts temporal information locally at each station, while GNO enables communication among stations. Together, these components enhance robust spatiotemporal communication across the input seismic network. The resulting tensor from the shared component is flattened along the time and channel axes before being fed into separate tasks. Both classification and regression operators are down-projection layers composed of two fully connected layers. The output of $\mathcal{Q}_{\text{class}}$ contains two channels, one for earthquakes and one for noise, for all stations. The regression output $(\hat{x}, \hat{y}, \hat{z}, \hat{t})$ is obtained by applying mean pooling across the station dimension to the predicted location features generated at each station (see Methods).

Benchmark with baselines

We compare the performance of QNO with classic baselines for earthquake detection, location, and catalog building, respectively. Each baseline model was evaluated using the checkpoint that achieved the best validation performance during training. Both QNO and baseline models are trained with the Northern California Earthquake Data Center (NCEDC) dataset (Fig. 2). We select events with a maximum magnitude of two and split the dataset for training, validation, and test sets based on the year of occurrence, reflecting prospective deployment on future data (see Method). For completeness, random split results regardless of year are provided in Supplementary Figs. 4 and 5. As expected, performance under the random split exceeded that under the temporal split (Fig. 4), consistent with a smaller generalization gap when distribution shift is minimized.

For earthquake location, we compare QNO with SeismicGNN⁵⁰. SeismicGNN incorporates spatial information in seismic source characterization by leveraging a graph neural network (GNN) to exchange information among stations. Node features at individual stations with SeismicGNN were extracted and processed with a convolutional neural network and multi-layer perceptron (MLP). As a result, QNO successfully determines the earthquake locations on all of the 1019 events in the test set across a wide range of areas in Northern California (Fig. 3 and Supplementary Fig. 6), although regions with higher seismicity rate and therefore more training samples exhibit better location accuracy (Supplementary Fig. 7). Compared with SeismicGNN, QNO provides more accurate geographical locations with lower mean absolute error (MAE), mean (μ), and standard deviation (σ) between predicted and catalog locations (Fig. 4). Although both methods share

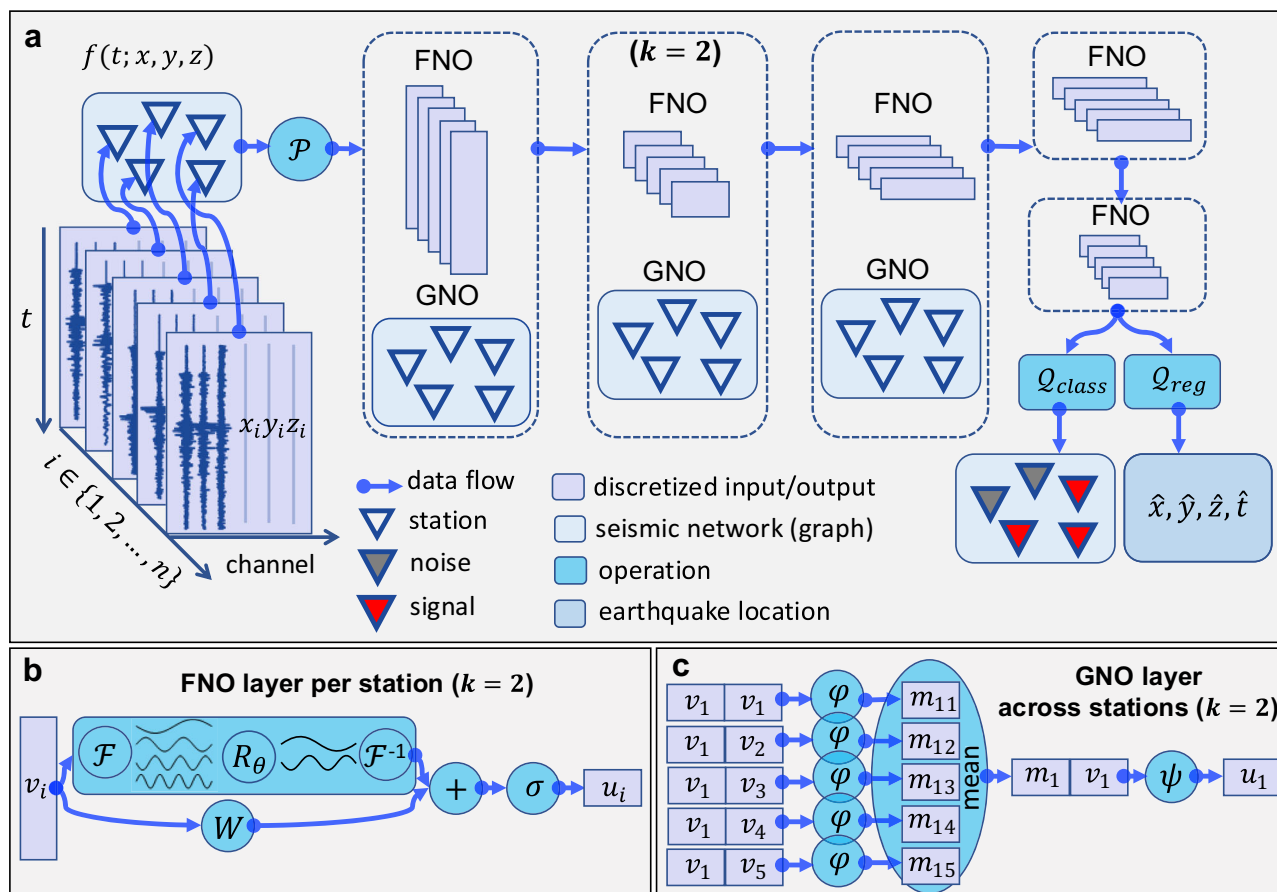


Fig. 1 | Quake neural operator (QNO). **a** Overview of the QNO architecture. An example with five stations ($n = 5$) is illustrated. **b** Operations in the second ($k = 2$) Fourier Neural Operator (FNO) layer. **c** Operations in the second Graph Neural Operator (GNO) layer. Only the feature updates at one station ($i = 1$) are shown. The architecture uses a multi-task learning framework consisting of a classification task Q_{class} for earthquake detection and a regression task Q_{reg} for earthquake location. The seismic waveform $f(t; x, y, z)$, recorded by multiple stations together with their arbitrary locations (x_i, y_i, z_i) , is encoded as the input. The outputs of Q_{class} are

earthquake and noise probabilities at each station, while the outputs of Q_{reg} are the earthquake location and origin time $(\hat{x}, \hat{y}, \hat{z}, \hat{t})$. $\mathcal{P}$: up-projection, $\mathcal{F}$: Fourier transform, $\mathcal{F}^{-1}$: inverse Fourier transform, R_θ : learnable spectral kernel, W : point-wise linear transformation, σ : nonlinear activation function, ϕ : message generation function, ψ : node feature update function, v_i : input feature at station i to a layer, u_i : output feature at station i from a layer, m_{ij} : edge feature between stations i and j , m_i : message at station i obtained by mean aggregation. See Methods and Supplementary Note 1 for full technical descriptions.

a similar distribution of depth prediction, the depth error predicted by QNO is slightly larger than SeismicGNN. In addition, QNO estimates the occurrence time of the input event relative to the starting time of the input waveforms, which enables us to build an earthquake catalog directly from continuous waveforms.

For the baseline for earthquake detection, we use PhaseNO¹⁹, a multi-station phase picking method built with neural operators. PhaseNO shares a slightly similar architecture with QNO, which uses both FNO and GNO together to extract features from multi-station seismic data, leading to better performance than single-station detection and picking models, such as PhaseNet¹¹ and EQTransformer¹³. We first compare the performance of QNO and PhaseNO with decreasing signal-to-noise ratio (SNR) and show the advantage of QNO for detecting microseismicity in strong noise. Then we compare the performance of both methods on earthquake catalog building.

We compare the detection performance between QNO and PhaseNO using event nc73384250 from the test set as an example (Fig. 5). We introduce real noise waveforms to the raw waveform of this event, adjusting the SNR of the resulting waveform from 20 dB to -8 dB. At an SNR of 20 dB, QNO successfully detects all earthquake signals, outputting a probability close to one for all stations, while PhaseNO accurately identifies both P- and S-wave phases. However, as the SNR decreases, the performance of PhaseNO degrades, failing to detect P

phases entirely at an SNR of zero. In contrast, QNO continues to detect the event at most stations even when the SNR reaches zero and consistently determines the event's location. Meanwhile, the location errors of QNO increase quickly with decreasing SNR when SNR is below zero. This experiment demonstrates that, despite the inability to pick phases in strong noise, the waveform coherence among stations remains instructive, providing enough information to detect and locate the earthquake. QNO's ability to detect events under high noise conditions enables us to monitor microseismicity where noise is consistently strong. More examples can be found in the Supplementary Figs. 8–17.

Interestingly, the predicted time and location errors do not always exhibit a strictly monotonic increase as the SNR decreases (Fig. 5). This non-monotonic behavior is likely due to the use of real noise recordings, which may contain microseisms, site effects, or instrument-related transients that introduce structured perturbations into the resulting waveforms. Since noise is added independently at each station and the QNO model integrates information nonlinearly across multiple stations, certain noise realizations may interfere less destructively with feature extraction, resulting in locally smaller errors. To further investigate this effect, we repeated the experiment using idealized Gaussian noise with the same SNR levels. The results in the Supplementary Fig. 17 show a more monotonic increase in errors as the

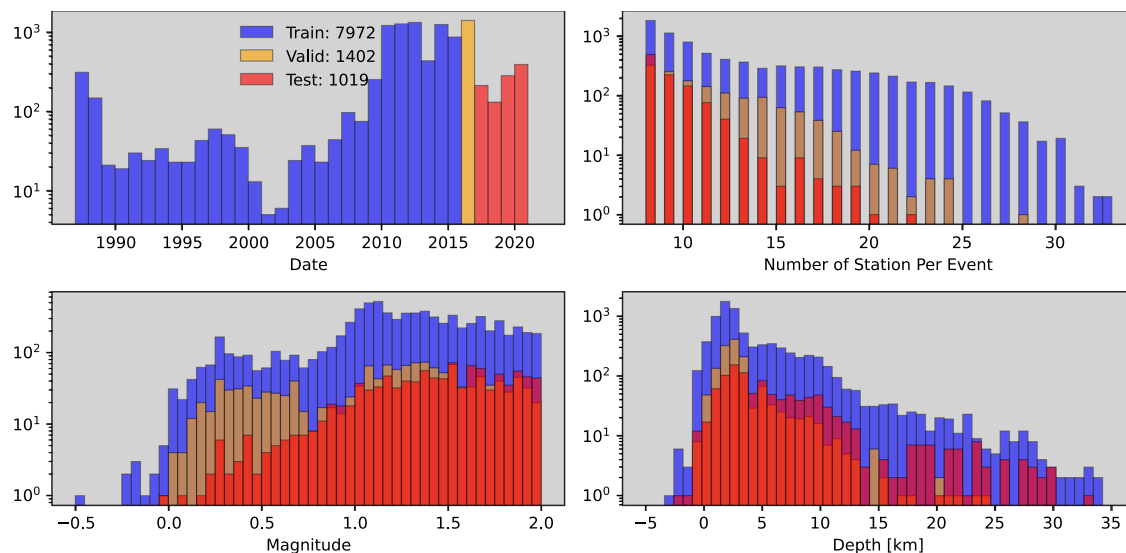


Fig. 2 | NCEDC seismic dataset for QNO training and evaluation. The dataset is split chronologically: earthquakes before 2016, in 2016, and from 2017 to 2021 form the training, validation, and test datasets, respectively. We select earthquakes with

a magnitude of less than two and recorded by at least eight stations. Both training and test datasets cover a wide range of depth and magnitude.

SNR decreases, confirming that structured real noise can introduce irregular, non-monotonic effects on performance. This observation highlights QNO's robustness to diverse noise structures and its ability to leverage spatiotemporal features to characterize seismic sources under complex real-world conditions.

Generalization across seismic network configurations

In the dataset, each sample corresponds to a unique seismic network geometry, as each earthquake is recorded by a different subset of 8-32 stations drawn from the full seismic network (Fig. 3). These station configurations vary significantly across events, and we further enhance this variability during training by randomly adding noise-only stations with real noise waveforms from the STEAD dataset⁵⁸. As a result, the model encounters a wide range of spatial sampling patterns during training, where each sample can be viewed as a different discretization of the spatial domain. In the ideal case of spatial data sampling that is sufficiently globally dense in the sense of Kovachki et al.⁵⁹ Definition 4, QNO would be expected to generalize to unseen station layouts without requiring retraining. In our numerical results, the evaluation of the trained QNO performs well on unseen station layouts, as shown by its consistent performance across diverse station configurations during testing (Fig. 4).

In addition to spatial generalization, QNO is also capable of handling test inputs with different temporal resolutions. While the FNO layers require uniform sampling in time for each sample due to the use of the Fast Fourier Transform (FFT), the actual sampling rate of input waveforms can vary among samples. To demonstrate this, we upsample all waveforms in the test dataset from 1500 to 3000 data points (corresponding to increasing the sampling rate from 100 to 200 Hz) and directly evaluate the QNO model trained on the original 100 Hz inputs on the upsampled dataset. Supplementary Fig. 18 shows that the model maintains good performance under higher temporal resolution, highlighting its generalization capability across both spatial and temporal domains.

Application to microseismic monitoring at the Geysers geothermal field

We apply the same trained QNO model evaluated in Figs. 4, 5 to monitor microseismicity at the Geysers geothermal field and compare its performance in earthquake catalog building with that of PhaseNO + GaMMA⁶⁰ using the same continuous seismic data. Our analysis begins

with raw seismic data from HH, EP, and DP sensors (Supplementary Table 1), covering the period from June 8, 2024 (00:00:00) to June 18, 2024 (00:00:00) in a geographical region centered at (-122.78, 38.79). The monitoring range spans 1.2 degrees in both longitude and latitude. This monitoring period includes the M_w 4.5 mainshock on 2024-06-08 at 19:34:29, which was one of the largest earthquakes in the Geysers region in the past decade. We process the data using the same procedure as for the training dataset, bandpassing between 1 and 10 Hz and normalizing by dividing by the maximum value. Each input waveform is 15 s long, with a 5-s overlap between consecutive inputs, resulting in 360 predictions for every hour of data. With a sampling rate of 100 Hz, each input contains 1500 data points.

Several key hyperparameters are used to construct the final catalogs with either QNO or PhaseNO + GaMMA, ensuring low-quality results are filtered out. QNO predicts the signal and noise probabilities for all input stations. Based on these predictions, we first determine whether a recording is considered to contain an earthquake signal with a threshold $T_{\text{QNO}}^{\text{signal}} \in [0, 1]$ on the probability of an earthquake signal at an individual station. The other is the minimum number of stations on which an earthquake signal has been detected, denoted as $N_{\text{QNO}}^{\text{signal}}$. An event is added to the catalog only when the number of stations with a signal probability greater than $T_{\text{QNO}}^{\text{signal}}$ exceeds $N_{\text{QNO}}^{\text{signal}}$. This approach allows for controlling the quality of the resulting catalog. Similarly, for PhaseNO, there is a threshold for determining phase picks for both P- and S-waves, denoted as $T_{\text{PhaseNO}}^{\text{pick}} \in [0, 1]$. These detected phases are then associated with events using GaMMA by requiring a minimum of $N_{\text{PhaseNO}}^{\text{pick}}$ picks per event to ensure a high-quality association result. Note that an earthquake signal contains both P and S phases, although not all of them can be detected with PhaseNO. Ideally, PhaseNO detects seismic phases, whereas QNO detects the earthquake signal. Thus, the number of stations required to record an earthquake signal for its detection is typically smaller for PhaseNO compared to QNO when using $N_{\text{QNO}}^{\text{signal}} = N_{\text{PhaseNO}}^{\text{pick}}$.

We compare the resulting catalogs using a minimum threshold of $N_{\text{QNO}}^{\text{signal}} = N_{\text{PhaseNO}}^{\text{pick}} = 16$ for both QNO and PhaseNO (Fig. 6 and Supplementary Fig. 19). Additionally, we use $T_{\text{QNO}}^{\text{signal}} = 0.6$ or $T_{\text{QNO}}^{\text{signal}} = 0.8$ for QNO, and $T_{\text{PhaseNO}}^{\text{pick}} = 0.3$ for PhaseNO when determining the existence of an earthquake signal or phase. These hyperparameter settings indicate that QNO applies much stricter criteria than PhaseNO. The detection trends over time are consistent between the QNO and PhaseNO catalogs, with the QNO catalog consistently containing more

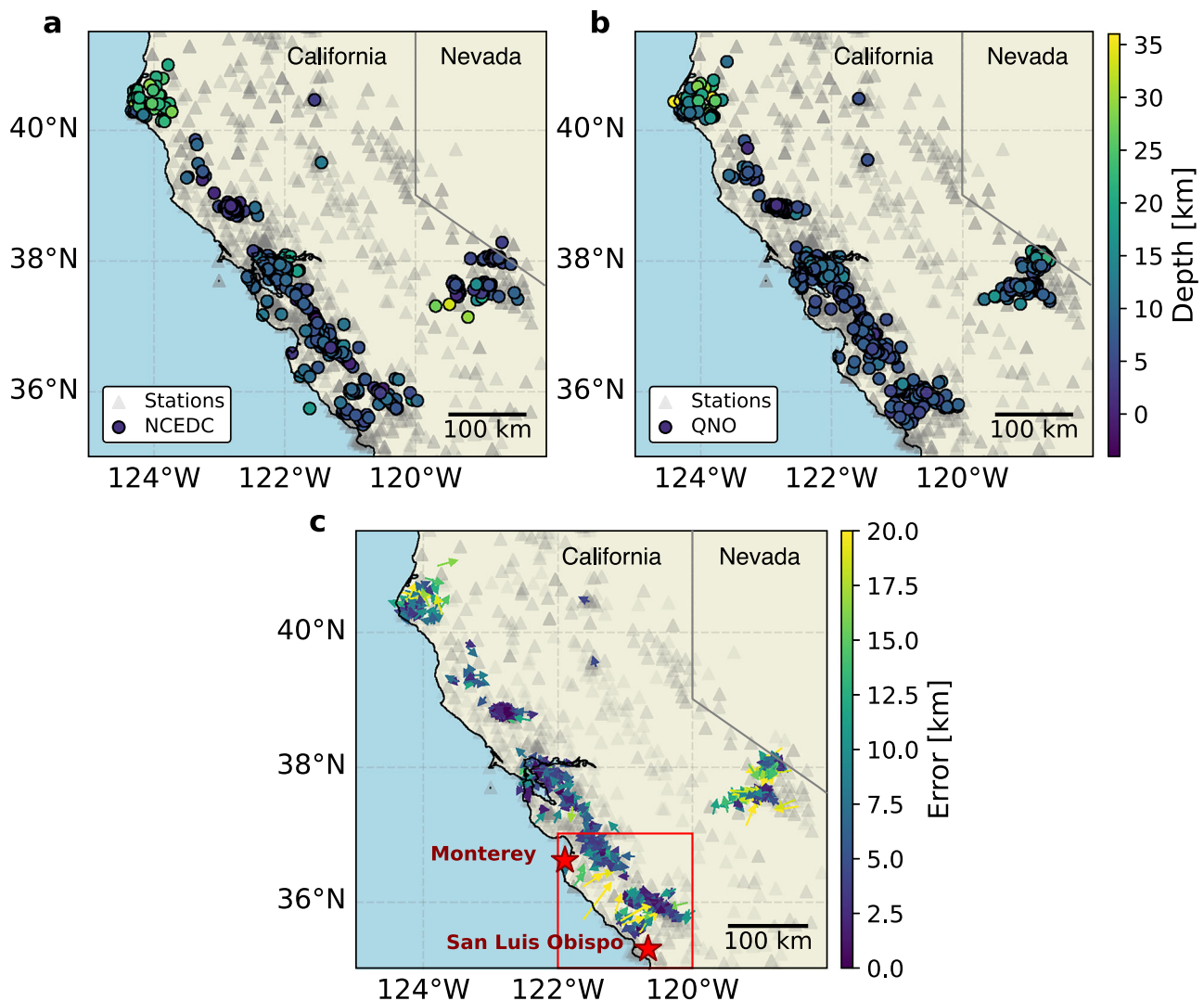


Fig. 3 | Comparison of earthquake locations between QNO and NCEDC catalog in the test dataset. a NCEDC catalog. **b** QNO prediction. **c** Arrows indicate the location errors for each event, pointing from the catalog epicenter to the

corresponding epicenter predicted by QNO. A zoom-in view of the region bounded by San Luis Obispo and Monterey is provided in Supplementary Fig. 7.

detections, suggesting that most earthquakes in the PhaseNO catalog were also detected by the QNO method. This consistency highlights the effectiveness of QNO in constructing accurate earthquake catalogs. Reducing QNO's threshold from 0.8 to 0.6 results in 10,167 detections, nearly one order of magnitude higher than the 874 earthquakes listed in the NCEDC catalog for the same period. When an event is detected based on the classification results and predefined thresholds, we retain the corresponding regression results as the location for that event. The locations of these earthquakes detected by QNO align with those detected by PhaseNO + GaMMA and in the NCEDC catalog (Fig. 6 and Supplementary Fig. 20), suggesting that most of these additional detections are meaningful rather than false positives. This comprehensive catalog, built with QNO, has great potential to reveal rich subsurface information about the status of the geothermal system.

We further assess the consistency among catalogs with a cross-validation test, treating one catalog as the ground truth and evaluating the relative performance of the others (Table 1). Since the true number of events within these continuous waveforms is unknown and each catalog may contain false positives and false negatives due to differences in algorithms, hyperparameters, and thresholds, accurately comparing detection performance across catalogs is challenging. To

facilitate comparison, earthquake events were considered true positives if their origin times fell within 3 s of an event in the benchmark catalog. With recall scores of 0.96 or 0.97, we observe that almost all NCEDC events were detected by both methods. The F1 score (which balances recall and precision) between QNO and PhaseNO+GaMMA is 0.80, and is similarly high for QNO at different thresholds. This indicates strong consistency between PhaseNO and QNO, even with QNO applying a much stricter standard for catalog building.

The comparison of probability functions predicted by QNO and PhaseNO further highlights QNO's advantage in microseismic monitoring (see Figs. 7, 8 and Supplementary Figs. 21–29). Figure 7 presents two representative microseismic events detected by QNO with thresholds of $T_{\text{QNO}}^{\text{signal}} = 0.7$ and $N_{\text{QNO}}^{\text{signal}} = 16$. Waveforms are plotted at all input stations, showing that the environmental noise is very strong, leaving few clear onsets of the seismic phase. QNO successfully identified 17 earthquake signals out of 36 input stations for both events. In contrast, PhaseNO only made picks on a few input waveforms with $T_{\text{PhaseNO}}^{\text{pick}} = 0.3$. These low-quality events detected by PhaseNO are filtered out during association using a threshold of $N_{\text{PhaseNO}}^{\text{pick}} = 16$. Even if the detection could be recovered by lowering the association threshold, the uncertainties in phase picking caused by strong noise would still propagate to the subsequent location step.

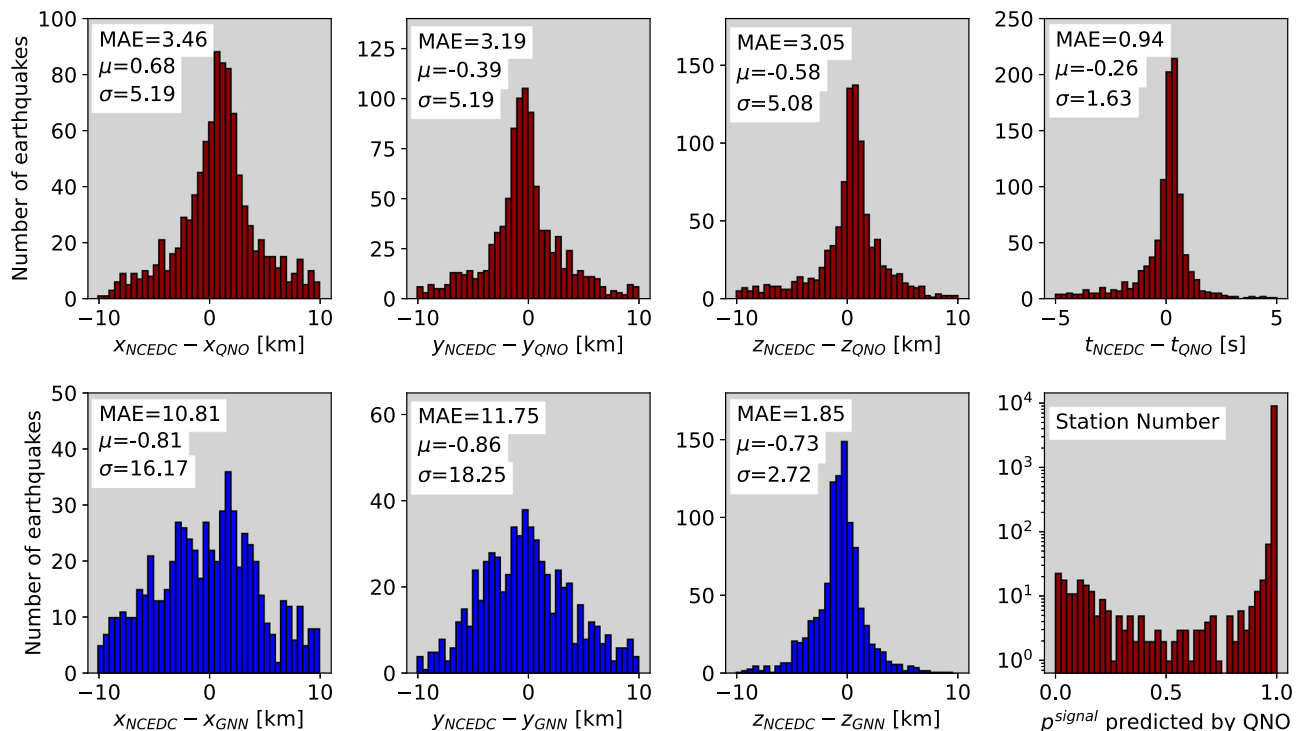


Fig. 4 | Performance comparison between QNO and SeismicGNN on the NCEDC test dataset. The red histograms represent QNO, and the blue ones represent SeismicGNN. The mean absolute error (MAE), mean (μ), and standard deviation (σ)

of the histogram values are labeled in the corresponding histograms. QNO reports a signal probability of at least 0.8 on approximately 98.23% of 9416 recordings at each station of the total 1019 events.

Figure 8 presents two additional seismic events detected by both the QNO and PhaseNO methods. For the event with a relatively high SNR, QNO achieves detection results consistent with PhaseNO. Conversely, for the event with a lower SNR, QNO identifies signals on more stations than PhaseNO. Additional examples are provided in the Supplementary Figs. 21–29.

Computational Cost of QNO

We investigate the capability of using QNO for real-time monitoring. Although training QNO is time-consuming, once the model is well-trained, it can be directly applied to monitor the area of interest. We evaluate the computational cost of one prediction during testing on an Apple M3 Max with results averaged over six runs (Fig. 9). The computational cost is reported for different numbers of stations and graph configurations using only the CPU with two threads. The computational time and memory usage largely depend on the number of edges and stations in a single input sample. For example, a network with $n = 20$ stations forming a fully connected directed graph with self-loops ($20 \times 20 = 400$ edges) requires ≈ 1 s per 15 s input sample and about 2.4 GB of memory. This represents the maximum computational cost for the 20-station case, which can be reduced by decreasing the number of edges. Even with 50 stations, the maximum computational cost is around 5.5 s per 15 s input sample with a memory usage of about 5.5 GB. For a real-time data stream where the system waits 10 s before processing the next prediction (with 5 s overlap between two predictions), this 5.5 s computational time allows QNO to be deployed directly for real-time monitoring using a CPU with two threads.

For a given seismic network with a fixed number of stations and geometry, the number of edges is determined by the graph configuration in the GNO layer, which is constructed based on the physical distance between stations (see Supplementary Note 1 and Supplementary Fig. 3). For a given station, edges are created between that station and any other station (including itself) if the distance between the two is less than or equal to a physical distance threshold D . A D

value larger than the maximum distance among stations results in a fully connected graph with a total of n^2 edges, which requires the highest computational cost. In practice, for large networks one can choose a relatively small D to reduce computational cost while preserving effective inter-station communication, as evidenced by generalization when training with $D = 40$ km and testing with $D = 20$ km and $D = 60$ km (Supplementary Fig. 30). For the 50-station network evaluated in Fig. 9, the computational cost decreases substantially when the number of edges is reduced, which further enables the deployment of a trained QNO model for real-time monitoring.

Discussion

QNO is a multi-task operator learning framework that integrates one classification task for earthquake detection and one regression task for earthquake location. Both tasks are jointly addressed by sharing the underlying neural operator structures, effectively solving the seismic monitoring problem, where earthquake detection and location are closely related. By combining FNO for temporal feature extraction and GNO for spatial information exchange, QNO efficiently handles the complex structure of seismic network data. Furthermore, it can process seismic data from networks with varying geometries while maintaining a fixed model architecture.

Additionally, unlike other multi-station algorithms that fully encode seismic waveforms at each individual station before exchanging information among stations⁵⁰, QNO facilitates communication among stations throughout the entire data flow within the neural operator. The sequential connection of FNO and GNO, along with the repeated application of these layers, ensures a comprehensive exchange of spatiotemporal information, enhancing earthquake detection and location accuracy.

Picking-based seismic monitoring workflows identify seismic arrivals from continuous data and then associate these picks with seismic events. Typically, a minimum number of picks is set as a hyperparameter, and any association results with fewer picks than this

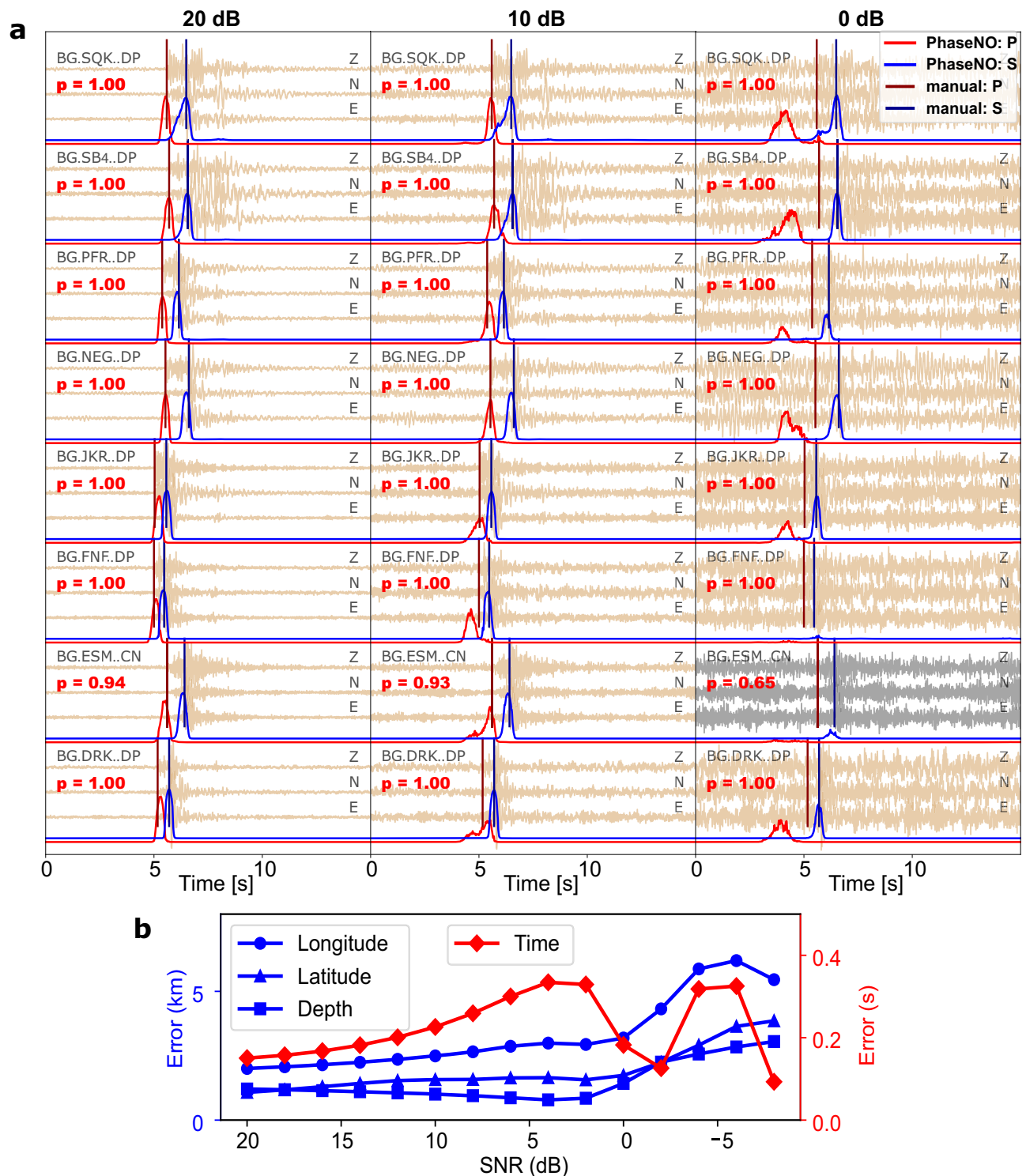


Fig. 5 | Example of detection and location performance under varying noise levels. We evaluate the impact of noise on detection and location performance by adding real noise waveforms to event nc73384250 while controlling the SNR from 20 dB to -8 dB. **a** Comparison of probabilities predicted by QNO and PhaseNO at SNR levels of 20, 10, and 0 dB. The red and blue curves show the pick probabilities predicted by PhaseNO. The two vertical lines mark the P- and S-wave arrival times

from the NCEDC catalog, manually determined by seismologists. The probabilities (p) predicted by QNO for each input station are displayed in red beneath the corresponding station names. “Z”, “N”, and “E” indicate the vertical, north-south, and east-west components of the three-component seismic waveform at each station. **b** Location error of QNO across different noise levels. Additional examples are available in the Supplementary Figs. 8-17.

threshold are filtered out. In cases of microseismicity, where only a few clear picks are detected across the seismic network, these events can easily be excluded by such filtering criteria. By contrast, QNO searches for the waveform information of a seismic event across multiple stations without picking and thus can detect small-magnitude events

effectively. At the same time, the location of these events is determined directly from waveforms, reducing potential location errors due to post-processing steps.

Here, we clarify how QNO differs from operator-learning approaches for phase picking, i.e., PhaseNO¹⁹. PhaseNO is formulated

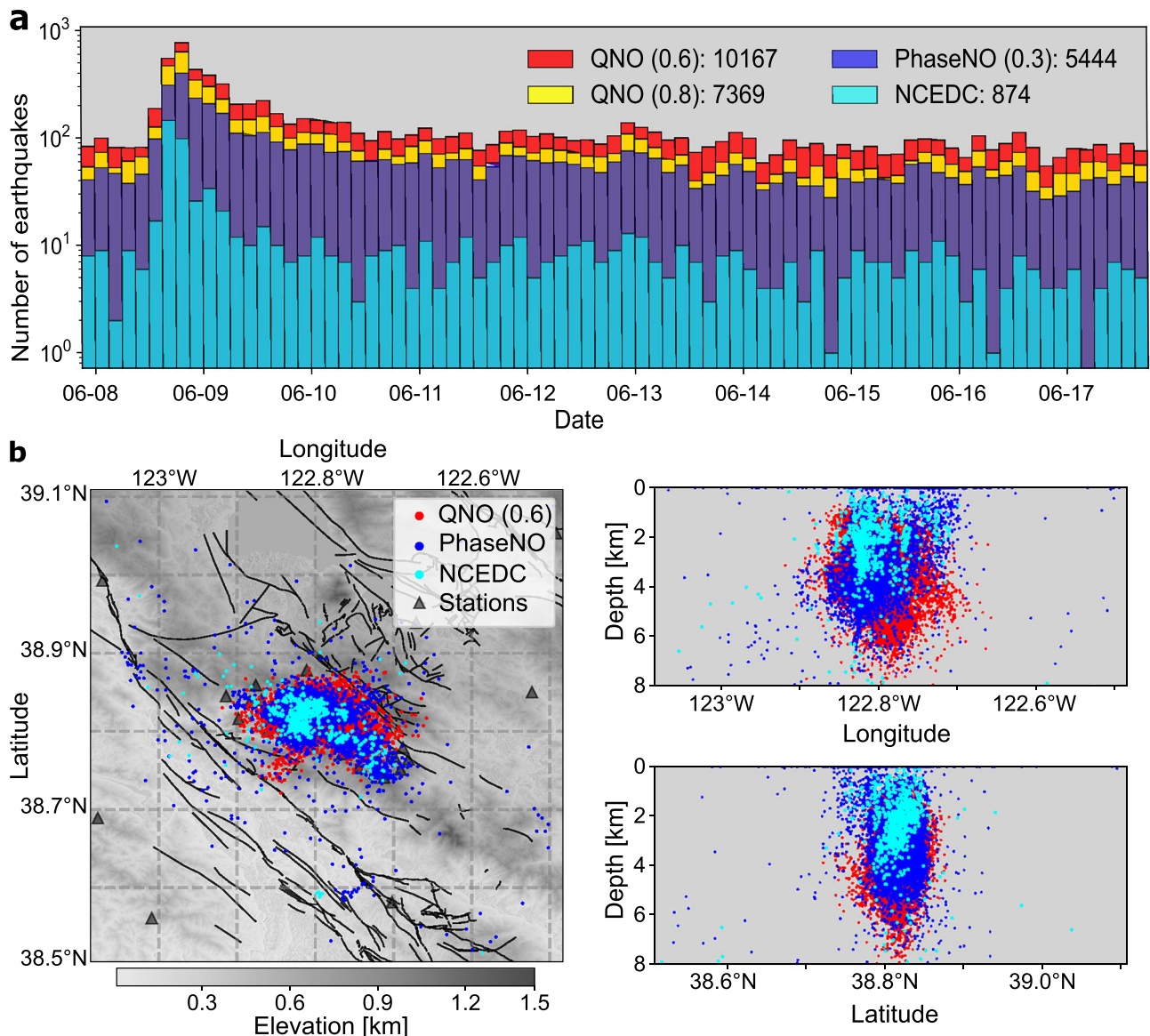


Fig. 6 | Comparison of detection performance between QNO and PhaseNO +GaMMA using ten days of continuous data from the Geysers geothermal field. **a** Number of earthquakes over time; **b** Earthquake locations. The thresholds for detecting a pick in PhaseNO and a signal in QNO at each station are indicated in parentheses in the histogram legends. Here, both QNO and PhaseNO+GaMMA require a minimum threshold of $N_{\text{QNO}}^{\text{signal}} = N_{\text{PhaseNO}}^{\text{pick}} = 16$ to identify a detection based on the number of detected signals (QNO) or picks (PhaseNO). The earthquake detection trends are consistent between QNO and PhaseNO+GaMMA, suggesting

the effectiveness of QNO in earthquake detection. With a threshold of 0.6, QNO yields 10,167 detections, nearly one order of magnitude more than the NCEDC catalog for the same period. Although increasing the QNO threshold from 0.6 to 0.8 reduces the number of detections, QNO still consistently detects more events than PhaseNO, which uses a probability threshold of 0.3. Note that the criteria for declaring a detection in QNO are significantly stricter than those used in PhaseNO in this example.

as a per-station picking task that outputs pseudo-probability distributions of P/S arrival times and relies on labeled picks for supervision; its outputs are station-level and are typically passed to a downstream associator or locator. In contrast, QNO is an end-to-end model that operates directly on multi-station waveforms without picking, jointly learning station-level detection (classification) and event-level location (regression) under a single multi-task objective. Architecturally, PhaseNO employs a U-shaped operator tailored to produce time-aligned station outputs, whereas QNO adopts a simpler sequential backbone with a lower memory footprint and shorter runtime, enabling near-real-time monitoring even on CPUs. We view QNO as complementary to pick-based monitoring pipelines: where high-quality picks are available, they remain useful; where onsets are emergent or noisy, and picks are unreliable, QNO enables direct cataloging from continuous waveform data.

Again, PhaseNO and QNO are not necessarily replacements for one another. Using them jointly can enhance both the detection rate and the reliability of detected events. The picked arrivals from PhaseNO, combined with the initial location results from QNO, allow for further optimization of event locations using the double-difference location algorithm⁶¹. PhaseNO excels in handling very early aftershock periods following large mainshocks where multiple events occur in a short time window and earthquake signals have relatively large amplitudes for PhaseNO to pick arrivals. In contrast, QNO is more effective for microseismicity, where phase onsets are unclear, and seismicity rates are relatively low. A small fraction (5.36%) of PhaseNO-catalog events in the Geysers geothermal dataset occur within 10 s of a preceding event and may be missed by QNO (Supplementary Fig. 31); however, this trade-off may be acceptable given QNO's improved overall microseismic detection performance. Additionally, while the

Table 1 | Cross-validation among earthquake catalogs for the Geysers geothermal field

Test catalog	Metric	Benchmark catalog			
		NCEDC	PhaseNO +GaMMA	QNO (0.8)	QNO (0.6)
NCEDC	Recall	1.00	0.16	0.14	0.10
	Precision	1.00	0.96	0.97	0.97
	F1 score	1.00	0.27	0.24	0.18
PhaseNO +GaMMA	Recall	0.96	1.00	0.74	0.59
	Precision	0.16	1.00	0.86	0.93
	F1 score	0.27	1.00	0.80	0.72
QNO (0.8)	Recall	0.97	0.86	1.00	0.73
	Precision	0.14	0.74	1.00	1.00
	F1 score	0.24	0.80	1.00	0.84
QNO (0.6)	Recall	0.97	0.93	1.00	1.00
	Precision	0.10	0.59	0.73	1.00
	F1 score	0.18	0.72	0.84	1.00

Parenthetic values indicate the probability threshold used to classify a recording at one station as containing an earthquake signal.

deep-learning-based picking algorithm is highly generalizable across different regions for picking seismic events^{62,63}, location algorithms typically require local events in the training dataset to account for regional velocity structures.

Ideally, QNO is designed to process one event per input time window. To investigate QNO's behavior when multiple earthquakes occur within the same input window, we constructed synthetic two-event cases by stacking real events from the NCEDC test dataset and varying both the amplitude ratio r and the inter-event delay time Δ between them (Supplementary Note 3). The delay time was defined as the difference between the earliest P-wave arrival times across all stations for the two events. We find that QNO still produces high signal probabilities at nearly all stations and returns a single location estimate for the input window, even when some stations lack clear phase picks for both events. We then compared this predicted location with catalog locations of both events to determine which was actually located (Supplementary Figs. 32 and 33). Across tested combinations of r and Δ , QNO consistently located the earlier-occurring event (Event 1), with location errors for Event 1 being smaller and less sensitive to r and Δ than those for Event 2. In contrast, Event 2's errors increased as Δ decreased or as its relative amplitude weakened, likely due to reduced visibility of its signals on the input waveforms. In addition, the azimuthal gap of Event 2 in Supplementary Fig. 32 is larger than that of Event 1, which may also contribute to the increased errors.

Although all numerical examples in this study use 15 s waveforms, this input length is not fixed; QNO can be trained for any waveform duration by adjusting the input layer size. However, for a trained model, the application-stage input length must match the training length, while the input sampling rate may differ. In practice, the waveform window length can be chosen based on the distribution of inter-event times derived from a high-resolution earthquake catalog of historical seismicity. Then separate QNO models could be trained for different time windows, for example, using a shorter window when seismicity rates are high to minimize the likelihood of multiple events occurring within the same input window. Moreover, when applied to continuous data, using overlapping predictions between successive time windows can help prevent missing events, as demonstrated in our case study of the Geysers geothermal field.

While shorter time windows are effective in minimizing the occurrence of multiple events within a single window, which is a key

consideration for microseismic monitoring, they may lead to repeated detections of long-duration seismic events. When the rupture duration or significant energy release of an earthquake exceeds the predefined time window, the algorithm may trigger detections in consecutive windows. In the Geysers case study, out of 10,167 raw detections, approximately 18% occurred within 5 s of a preceding detection. These detections may include both temporally close natural events and repeated detections of the same event. While such repeated detections reflect the robustness of the QNO classification and regression operators in capturing different stages of the same event (e.g., codas or late-arriving phases), they may introduce duplicate entries in the resulting catalog. For practical applications, these duplicates could be addressed through standard post-processing clustering. For example, by simply merging detections within a temporal proximity of 5 s, we obtained a refined catalog of 8346 events from the QNO detections (with a probability threshold of 0.6), which still represents a nearly ten-fold increase compared to routine monitoring results.

Method

We propose QNO, a multi-task neural operator designed for earthquake detection and location directly on continuous seismic waveform data without explicit phase picking or phase association. This section provides an overview of the QNO, including its architecture, loss function, and training dataset.

Quake neural operator in multi-task learning

QNO performs a nonlinear mapping from the input function $f(t; x, y, z)$ to the output in a multi-task learning framework. It consists of a shared feature extractor $\mathcal{F}$ followed by separate classification $\mathcal{Q}_{\text{class}}$ and regression $\mathcal{Q}_{\text{reg}}$ branches:

$$\mathbf{h} = \mathcal{F}(f(t; x, y, z)), \quad (1)$$

$$\mathcal{G}(f(t; x, y, z)) = (\mathcal{Q}_{\text{class}}(\mathbf{h}), \mathcal{Q}_{\text{reg}}(\mathbf{h})), \quad (2)$$

where $\mathbf{h}$ represents the shared feature extracted from the input function with a nonlinear operator $\mathcal{F}$. $\mathcal{G}$ denotes the QNO. For multi-station seismic data, (x, y, z) denotes the spatial coordinates of the seismic stations, while t denotes the temporal coordinate. The input samples are the discretized values of this function $f(t; x, y, z)$ in both space and time. In principle, neural operators $\mathcal{F}$, $\mathcal{G}$, $\mathcal{Q}_{\text{class}}$ and $\mathcal{Q}_{\text{reg}}$ are formulated to operate on inputs defined on arbitrary discretizations⁵⁹. In practice, this allows QNO to be trained and evaluated on varying seismic network geometries.

For earthquake detection, the classification task $\mathcal{Q}_{\text{class}}$ processes the shared representation $\mathbf{h}$ to produce a spatial logits field, which is converted into a probability function $p(x, y, z)$ using the softmax function:

$$p(x, y, z) = \text{softmax}(\mathcal{Q}_{\text{class}}(\mathbf{h})). \quad (3)$$

The probability function is evaluated at the station locations. Here, $\mathbf{p} = \{\mathbf{p}_i\}_{i=1}^n \in \mathbb{R}^{n \times 2}$ denotes the probabilities for earthquake signals and noise at stations located at (x_i, y_i, z_i) , where $i \in \{1, 2, \dots, n\}$ and n is the number of stations in the input. Each $\mathbf{p}_i \in \mathbb{R}^2$ represents the probability distribution over signal and noise for the i -th station:

$$\mathbf{p}_i = [p_i^{\text{signal}}, p_i^{\text{noise}}], \quad p_i^{\text{signal}} + p_i^{\text{noise}} = 1. \quad (4)$$

For earthquake location, the regression task outputs the earthquake source location $(\hat{x}, \hat{y}, \hat{z})$ and origin time $\hat{t}$:

$$\hat{\mathbf{r}} = \mathcal{Q}_{\text{reg}}(\mathbf{h}), \quad (5)$$

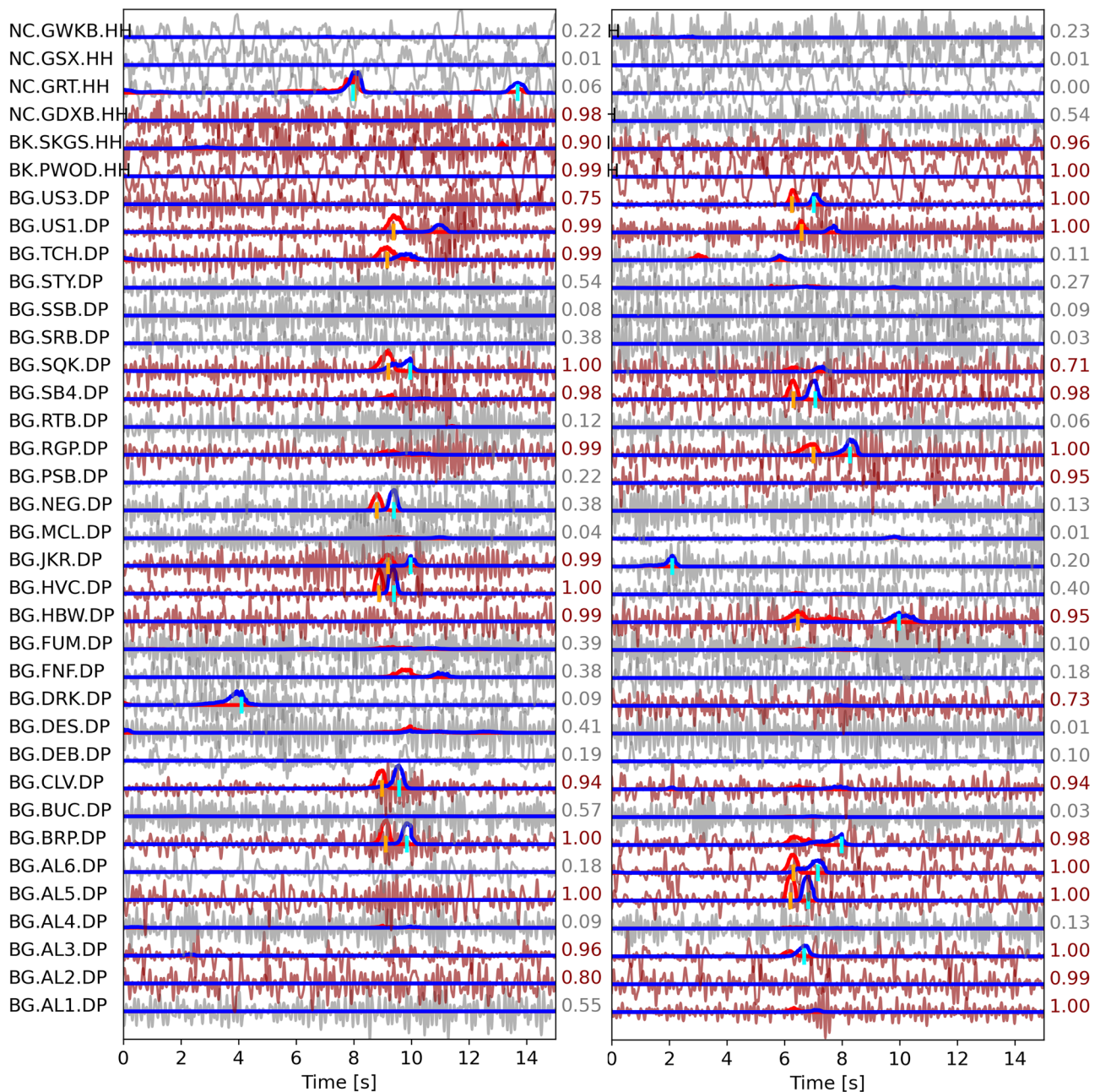


Fig. 7 | Comparison of detection performance between QNO and PhaseNO+GaMMA for two microseismic events. Waveforms of all input stations are plotted. The signal probability p_i^{signal} predicted by QNO is shown at the end of each waveform. Waveforms are highlighted in red if $p_i^{\text{signal}} > 0.7$, indicating detection of an earthquake signal at that threshold. The probability curves predicted by PhaseNO are plotted in red for P-phases and in blue for S-phases. Vertical bars indicate seismic phases determined when the probabilities are larger than 0.3. In the left and

right panels, PhaseNO did not find picks on 10 and 8 stations, respectively, whose waveforms were classified by QNO as containing earthquake signals. If we require both QNO and PhaseNO+GaMMA to have a minimum threshold of $N_{\text{QNO}}^{\text{signal}} = N_{\text{PhaseNO}}^{\text{pick}} = 16$ for a valid detection, then these events are overlooked by PhaseNO+GaMMA. In contrast, QNO detected these events and directly determined their locations from these input waveforms. More examples can be found in the Supplementary Figs. 21–29.

where $\hat{\mathbf{r}} = (\hat{x}, \hat{y}, \hat{z}, \hat{t}) \in \mathbb{R}^4$ is the predicted result.

Architecture of quake neural operator

The architecture of QNO (Fig. 1) mainly relies on two types of neural operators, which can efficiently process the spatiotemporal representation of seismic data over time and space (see Supplementary Note 1): One is the Fourier Neural Operator (FNO)⁵⁶ and the other is the Graph Neural Operator (GNO)⁵⁷. FNO learns global correlations in the temporal axis using Fourier transforms, effectively capturing long-range dependencies in the data. GNO operates on graph structures to model relationships among seismic stations, which effectively deals

with the irregular sampling of seismic data in the spatial domain. Started from the input function $f(t; x, y, z)$, the neural operator $\mathcal{F}$ shared by both the classification and regression tasks can be viewed as a series of mappings through layers of operation:

$$\mathbf{h} = \text{FNO}_{k+2} \circ \text{FNO}_{k+1} \circ \text{GNO}_k \circ \text{FNO}_k \circ \dots \circ \text{GNO}_1 \circ \text{FNO}_1 \circ \mathcal{P}(f(t; x, y, z)), \quad (6)$$

where $\circ$ denotes the mapping between two layers. With $k=3$, QNO uses three combinations of layers of FNO and GNO. The input function is first passed through an up-projection layer $\mathcal{P}$, which maps the input

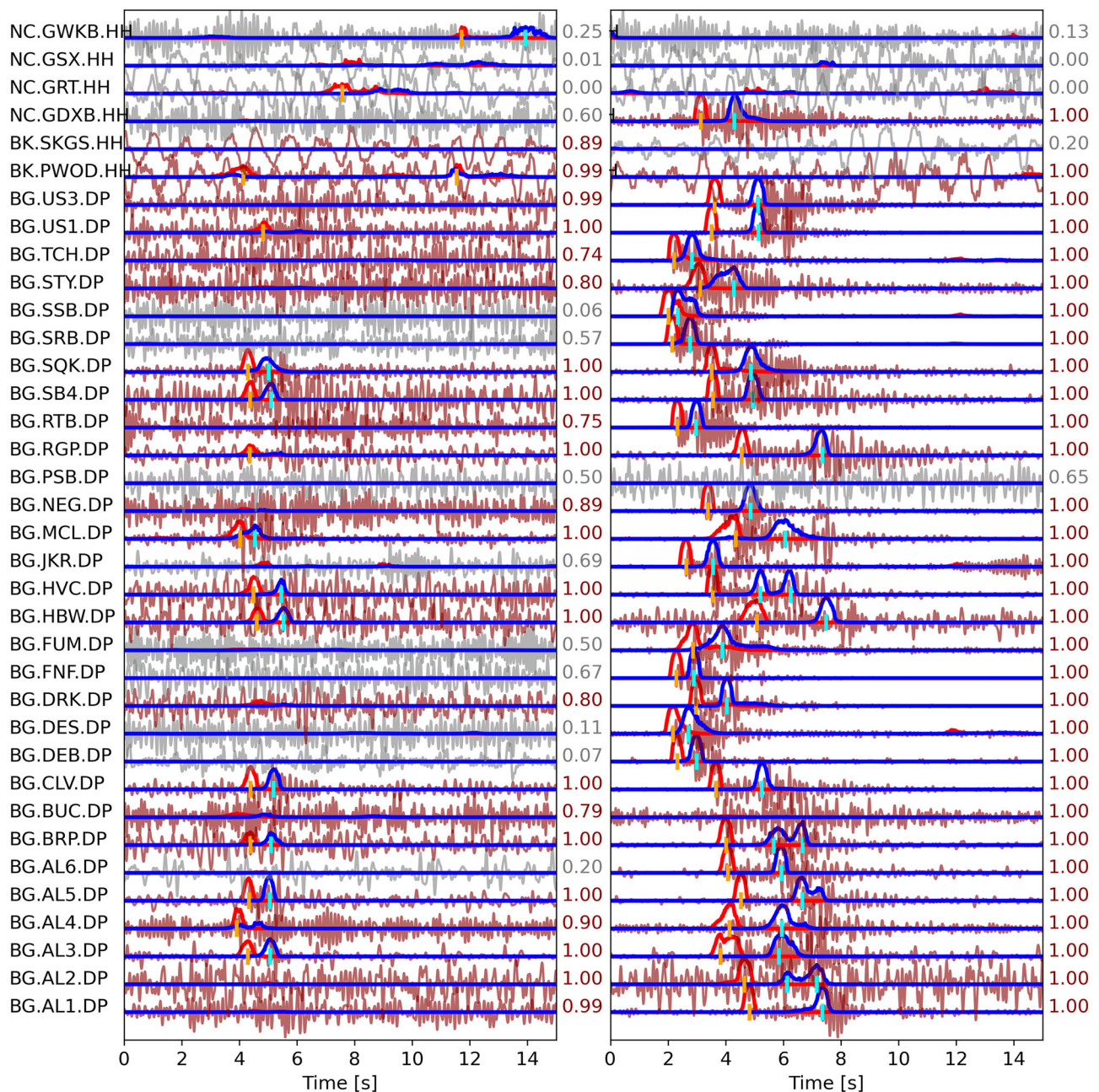


Fig. 8 | Two seismic events detected by both the QNO and PhaseNO+GaMMA methods. For the event with a relatively high SNR (right panel), QNO and PhaseNO produce consistent detection results. In contrast, for the event with a lower SNR (left panel), QNO detects signals on more stations compared to PhaseNO.

function to a high-dimensional representation. Then the up-projected data is passed through three combinations of FNO and GNO layers, allowing sufficient exchange of information between the time and space domains.

Each FNO layer performs a 1-D spectral convolution along the temporal axis at each station: we apply a Fourier transform to the per-station features, multiply the lowest M_k frequency modes by learned complex weights while zeroing higher modes, inverse-transform back to the time domain, and then combine the result with a pointwise (1×1) linear projection, followed by a nonlinearity⁵⁶. To reduce boundary artifacts associated with applying Fourier transforms to non-periodic time series, we apply zero-padding (50 samples on each end) prior to the FNO layers. This suppresses spurious high-frequency components induced by discontinuities at the signal boundaries and stabilizes the spectral representation used by the operator layers. In addition, each

FNO layer retains only the first M_k lowest-frequency modes, as high-frequency components are more difficult to learn and are truncated during training⁵⁶. The number of modes in each FNO layer is 24, 12, 8, 8, and 8, respectively. The width (or channel number) of the discretized function at each station varies with the dimension. Across the FNO layers, the per-station discretized representation $v(t; \mathbf{x}_i) \in \mathbb{R}^{C_k \times T_k}$, with $\mathbf{x}_i = (x_i, y_i, z_i)$, takes the following shapes by layer: 48×1500 , 96×500 , 192×100 , 192×50 , and 24×50 . The first dimension is the channel width C_k ; the second is the number of time samples T_k . As the network progresses through downsampling, the number of Fourier modes M_k is reduced in proportion to the compressed resolution, while the channel dimensions are increased to enrich feature representations. Nonlinearity is introduced in all FNO layers using the Gaussian Error Linear Unit⁶⁴, which applies a smooth, probabilistic gating mechanism that approximates the input multiplied by the

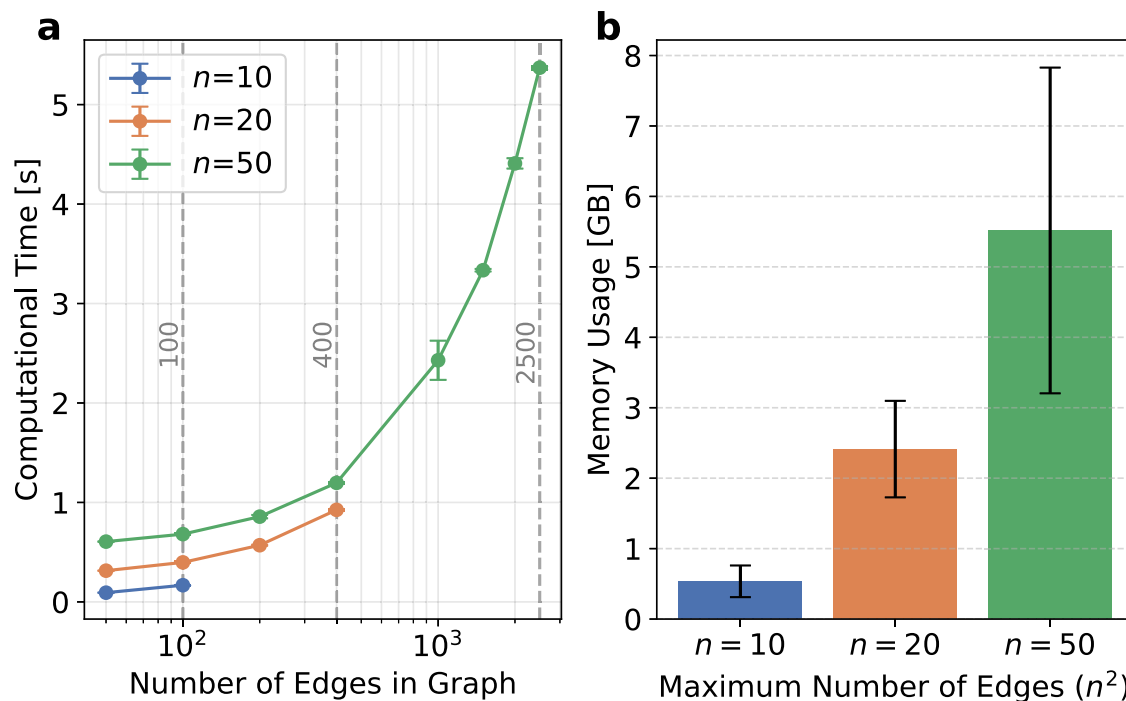


Fig. 9 | QNO's computational cost of one prediction during testing, evaluated on an Apple M3 Max using CPU with two threads, with results averaged over six runs. a Computational time (mean $\pm$ standard deviation) as a function of the number of edges in the graph for different numbers of stations n . Each curve

corresponds to a fixed n , and dashed vertical lines indicate the maximum possible number of edges n^2 for that configuration. **b** Memory usage (mean $\pm$ standard deviation) for graphs at the maximum number of edges n^2 for $n = 10, 20$, and 50 , reported in gigabytes.

cumulative distribution function of a standard normal distribution. The output of the last FNO layer is flattened into 1200 channels before feeding into $\mathcal{Q}_{\text{class}}$ and $\mathcal{Q}_{\text{reg}}$.

At each GNO layer, stations are treated as nodes in a graph, and their features are updated through a message-passing framework⁶⁵. The graph is constructed in the input spatial domain using a distance threshold D (two stations are connected with an edge if their pairwise distance $\leq D$). Unless otherwise noted, for example, when explicitly analyzing generalization with respect to D (see Supplementary Fig. 30), we set $D = 40$ km. Per-station temporal features $v(t; \mathbf{x}_i) \in \mathbb{R}^{C_k \times T_k}$ produced by the preceding FNO layer serve as node features. For each edge (i, j) , an edge message is computed by a differentiable map φ that takes the two node features concatenated along the temporal axis as the input, $m_{ij} = \varphi(v(\mathbf{x}_i), v(\mathbf{x}_j))$. Node i then performs mean aggregation $\bar{m}_i = \frac{1}{|N(\mathbf{x}_i)|} \sum_{j \in N(\mathbf{x}_i)} m_{ij}$ and updates its representation via a second map $\psi: u(\mathbf{x}_i) = \psi(v(\mathbf{x}_i), \bar{m}_i)$. In QNO, both φ and ψ are two-layer MLPs with hidden width $4C_k$, where C_k is the channel dimension of the node features output by the k -th FNO layer preceding the k -th GNO layer. The message-passing framework in the GNO layer is permutation-invariant, accommodates irregular station layouts, and couples spatial communication with the temporal representations learned by the FNO. QNO uses three GNO layers interleaved with FNO layers.

The final output $\mathbf{h}$ from the shared part branches into two separate parts for classification and regression, respectively. To reduce the dimensionality of the shared feature representation $\mathbf{h}$, we use two different down-projection layers for the separated branches of the regression and classification tasks. The classification output is generated by passing $\mathbf{h}$ through the down projection layer $\mathcal{Q}_{\text{class}}$ and applying the softmax function. The regression output is generated by passing $\mathbf{h}$ through the down-projection layer $\mathcal{Q}_{\text{reg}}$, producing the predicted earthquake source location and origin time. Both $\mathcal{Q}_{\text{class}}$ and $\mathcal{Q}_{\text{reg}}$ are two-layer fully connected neural networks.

Loss function for the multi-task learning framework

The total loss function of the multi-task neural operator is a weighted sum of the individual task losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{class}} + \alpha \mathcal{L}_{\text{reg}}, \quad (7)$$

where α is a weight that balances the contribution of the regression task relative to the classification task. $\mathcal{L}_{\text{class}}$ is the classification loss; and $\mathcal{L}_{\text{reg}}$ is the regression loss. Here we choose $\alpha = 1$.

The cross-entropy loss function for the classification task is an expectation over the joint distribution $(X, Y) \sim \mathcal{D}_1$, where X denotes evaluations of the seismic waveform function $f(t; x, y, z)$ at the station locations of a seismic network, and $Y = [Y^{\text{signal}}, Y^{\text{noise}}]$ denotes the corresponding labels:

$$\mathcal{L}_{\text{class}} = \mathbb{E}_{(X, Y) \sim \mathcal{D}_1} \left[- \sum_{i=1}^n \left(Y_i^{\text{signal}} \log p_i^{\text{signal}} + Y_i^{\text{noise}} \log p_i^{\text{noise}} \right) \right]. \quad (8)$$

The regression task uses mean squared error (MSE) to predict the source parameters of an earthquake $\hat{\mathbf{r}} = (\hat{x}, \hat{y}, \hat{z}, \hat{t})$:

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{(X, \mathbf{r}_{\text{true}}) \sim \mathcal{D}_2} \left[(\hat{x} - x_{\text{true}})^2 + (\hat{y} - y_{\text{true}})^2 + (\hat{z} - z_{\text{true}})^2 + (\hat{t} - t_{\text{true}})^2 \right], \quad (9)$$

where $\mathbf{r}_{\text{true}} = (x_{\text{true}}, y_{\text{true}}, z_{\text{true}}, t_{\text{true}})$ contains the ground-truth earthquake source locations and origin times. These labels usually come from an existing earthquake catalog used to construct the training dataset.

Training dataset

We use a training dataset previously employed to train two deep neural phase pickers, PhaseNet¹¹ and PhaseNO¹⁹. Using a similar training dataset enables a fair comparison and evaluation of the performance

of each method after training. The dataset consists of three-component earthquake waveforms and event catalogs spanning 30 years from the Northern California Earthquake Data Center (NCEDC). To train QNO for microseismic monitoring, we focus on events with magnitudes up to 2. To improve accuracy and minimize uncertainties, we select events recorded by at least eight stations. The event depths range from -4 to 36 km (Fig. 2). Depths are referenced to sea level; negative values indicate earthquakes occurring above sea level in high-elevation regions (e.g., mountains in California).

We adopt a temporal split as the primary setting: events prior to 2016 form the training set, while events from 2016 and 2017–2021 are designated as the validation and test sets, respectively. This selection yields 7972 events in the training set, 1402 events in the validation set, and 1019 events in the test set. This design reflects practical deployment, where models trained on historical earthquakes are applied to future events; following common practice, this temporal split implicitly disregards the question of stationarity. It also introduces temporal distribution shift due to changes in station coverage, noise, and catalog completeness. Unless otherwise specified, all evaluations use the model trained on events before 2016. For a complementary random split and additional analyses, see Supplementary Note 4 and Supplementary Figs. 4 and 5.

We perform data augmentation and pre-processing when preparing the training dataset. In the NCEDC dataset downloaded from the data center, waveforms from all stations associated with an event contain earthquake signals. However, in real-world scenarios, not all stations record earthquake waveforms. To simulate this, we generate up to 16 virtual stations with random locations within the computational domain and assign noise waveforms to these virtual stations. Each station is assigned a classification label: [1, 0] for earthquake signals and [0, 1] for noise, where the first channel indicates p_i^{signal} and the second channel represents p_i^{noise} . To improve QNO's ability to detect events in the presence of strong noise, we also added noise to the earthquake waveforms at real stations. The noise waveforms were randomly selected from the 235,000 noise samples available in the STEAD dataset⁵⁸, and the noise varies across different stations. Then these raw waveforms were first preprocessed by removing the trend using a demeaning procedure and applying a bandpass filter with a frequency range of 1 to 10 Hz. After filtering, the waveforms were normalized by dividing each channel by its maximum value.

We use a relatively short time window of 15 s to train QNO. This window length is appropriate for local microseismic monitoring where waves decay fast during propagation. A short time also reduces the possibility of the existence of multiple events in one sample. Since QNO faces the challenge of handling multiple events within a single input time window (see Supplementary Note 3), selecting a short time window is a straightforward way to address this issue and is particularly effective for microseismic monitoring, where inter-event times are generally much longer than those in aftershock sequences of large earthquakes. Moreover, we use overlapping time windows when processing continuous data, which helps to reduce the possibility of missing events.

We construct a computational domain that can cover the locations of the local earthquake and stations for each sample in the dataset. Coordinates on the computational domain have values between 0 and 1, which is favorable for the learning process. The physical lower bounds of longitude λ_0 and latitude ϕ_0 for the computational domain are given by:

$$\lambda_0 = \frac{\lambda_{\max} + \lambda_{\min}}{2} - \frac{d}{2}, \quad (10)$$

$$\phi_0 = \frac{\phi_{\max} + \phi_{\min}}{2} - \frac{d}{2}, \quad (11)$$

where $\lambda_{\max}$ is the maximum longitude, $\lambda_{\min}$, the minimum longitude, $\phi_{\max}$, the maximum latitude, $\phi_{\min}$, the minimum latitude of all stations around an earthquake. Each sample is mapped with a varying center so that all stations in the graph are around the middle of the computational domain. In addition, d represents the extent of the computational domain on the Earth's surface. The chosen range d should be large enough to encompass all the stations within the graph. Since the propagation range of small-scale events is typically short, a selection of $d = 1.2^\circ$ is sufficient and appropriate for monitoring local earthquakes in our experiments. After determining the physical lower bounds of the computational domain, we can calculate the relative position of each station within this domain:

$$x_i = \frac{\lambda_i - \lambda_0}{d}, \quad (12)$$

$$y_i = \frac{\phi_i - \phi_0}{d}, \quad (13)$$

$$z_i = \frac{\eta_i - \eta_{\min}}{\eta_{\max} - \eta_{\min}}, \quad (14)$$

where λ_i , ϕ_i , and η_i are respectively the longitude, latitude, and depth of the i -th station. $\eta_{\max}$ is the maximum depth and $\eta_{\min}$ is the minimum depth of the computational domain. We choose $\eta_{\min} = -4$ km and $\eta_{\max} = 36$ km to include the depth of all earthquakes in the dataset. The computational domain and the relative positions (x_i , y_i , z_i) of the stations are computed independently for each data sample during training. For real-world scenarios, the relative positions are computed only once for a given seismic network. These transformed coordinates are treated as node attributes and three additional channels of the input, along with the three-component waveform information.

Similarly, the regression label for earthquake location is the earthquake's relative location (x_{true} , y_{true} , z_{true}) in the computational domain:

$$x_{\text{true}} = \frac{\Lambda - \lambda_0}{d}, \quad (15)$$

$$y_{\text{true}} = \frac{\Phi - \phi_0}{d}, \quad (16)$$

$$z_{\text{true}} = \frac{H - \eta_{\min}}{\eta_{\max} - \eta_{\min}}, \quad (17)$$

where (Λ , Φ , H) denotes the ground-truth earthquake location, which in practice is usually taken from the earthquake catalog. The time predicted by QNO is the occurrence time of an event relative to the start of the input time series. Assuming the origin time T is within a range of 10 s earlier ($t_{\min} = -10$ s) and 10 s later ($t_{\max} = 10$ s) than the starting time of an input waveform, the time t_{true} of the regression label that we train QNO to predict should be:

$$t_{\text{true}} = \frac{T - t_{\min}}{t_{\max} - t_{\min}}. \quad (18)$$

We train QNO with the Adaptive Moment Estimation (Adam) optimizer⁶⁶ with a batch size of one and a learning rate of 10^{-4} . Adam is a stochastic gradient-based optimizer that computes adaptive learning rates for each parameter based on estimates of first and second moments of the gradients. The training takes around three hours for one epoch on one NVIDIA Tesla V100 GPU. We train the model with 19 epochs and then evaluate it on the test dataset and 10-day continuous data from the Geysers geothermal field.

When evaluating performance on the test dataset (Fig. 4), we first convert the predicted coordinates ($\hat{x}, \hat{y}, \hat{z}, \hat{t}$) of each sample to physical coordinates corresponding to longitude, latitude, depth, and origin time. We then compute the differences between the converted predictions and the corresponding catalog values (Δ, Φ, H, T). Location errors in degrees are converted to kilometers using a factor of 111 km/° for latitude and approximately 88 km/° for longitude, based on the mean latitude of the study area.

Data availability

All waveform data are publicly available from the Northern California Earthquake Data Center (NCEDC), doi:10.7932/NCEDC⁶⁷. The NCEDC catalog for the Geysers geothermal data is publicly available at NCEDC (last access: August 1, 2025). Source data are provided with this paper.

Code availability

The code developed in this study for the Quake Neural Operator is publicly available on GitHub at <https://github.com/sun-hongyu/QNO> and archived on Zenodo at <https://doi.org/10.5281/zenodo.20072888>⁶⁸. The PhaseNO software is available at <https://github.com/sun-hongyu/PhaseNO>. The SeismicGNN software is available at <https://github.com/martijnende/SeismicGNN>.

References

- Florez, M. A., Li, B. Q., Shelly, D. R., Angulo, M. V. & Sanabria-Gómez, J. D. Long-term dynamics of earthquake swarms in the Yellowstone caldera. *Sci. Adv.* **11**, eadv6484 (2025).
- Suzuki, R. et al. The forearc seismic belt: A fluid pathway constraining down-dip megathrust earthquake rupture. *Science* **389**, 190–194 (2025).
- Tan, X. et al. A clearer view of the current phase of unrest at Campi Flegrei caldera. *Science* **390**, 70–75 (2025).
- Fu, P. et al. Close observation of hydraulic fracturing at EGS Collab Experiment 1: Fracture trajectory, microseismic interpretations, and the role of natural fractures. *J. Geophys. Res. Solid Earth* **126**, e2020JB020840 (2021).
- Verliac, M. & Le Calvez, J. Microseismic monitoring for reliable CO₂ injection and storage—Geophysical modeling challenges and opportunities. *Lead. Edge* **40**, 418–423 (2021).
- Leong, Z. X. & Zhu, T. Machine learning-assisted microearthquake location workflow for monitoring the Newberry enhanced geothermal system. *J. Geophys. Res. Mach. Learn. Comput.* **1**, e2024JH000159 (2024).
- Zhang, M., Liu, M., Feng, T., Wang, R. & Zhu, W. LOC-FLOW: an end-to-end machine learning-based high-precision earthquake location workflow. *Seismol. Res. Lett.* **93**, 2426–2438 (2022).
- Perol, T., Gharbi, M. & Denolle, M. Convolutional neural network for earthquake detection and location. *Sci. Adv.* **4**, e1700578 (2018).
- Cianetti, S., Lomax, A., Michélini, A. & Giunchi, C. Which is better: deep-learning or manual seismic arrival-time picking? *Geophys. J. Int.* **244**, ggaf494 (2026).
- Sun, H. Advancing seismic monitoring with operator learning. In *Fifth International Meeting for Applied Geoscience & Energy, Post-Convention Workshop W-11: Machine Learning in Geosciences in the Age of Generative AI, Expanded Abstracts*, 2976–2980 (Society of Exploration Geophysicists, 2026).
- Zhu, W. & Beroza, G. C. PhaseNet: a deep-neural-network-based seismic arrival-time picking method. *Geophys. J. Int.* **216**, 261–273 (2019).
- Zhu, L. et al. Deep learning for seismic phase detection and picking in the aftershock zone of 2008 Mw 7.9 Wenchuan earthquake. *Phys. Earth Planet. Inter.* **293**, 106261 (2019).
- Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y. & Beroza, G. C. Earthquake Transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nat. Commun.* **11**, 3952 (2020).
- Chai, C. et al. Using a deep neural network and transfer learning to bridge scales for seismic phase picking. *Geophys. Res. Lett.* **47**, e2020GL088651 (2020).
- Sun, H., Yang, Y., Azzadenesheli, K., Clayton, R. W. & Ross, Z. E. Accelerating time-reversal imaging with neural operators for real-time earthquake locations. *arXiv preprint arXiv:2210.06636* (2022).
- Feng, T., Mohanna, S. & Meng, L. EdgePhase: a deep learning model for multi-station seismic phase picking. *Geochem. Geophys. Geosyst.* **23**, e2022GC010453 (2022).
- Yuan, C., Ni, Y., Lin, Y. & Denolle, M. Better together: ensemble learning for earthquake detection and phase picking. *IEEE Trans. Geosci. Remote Sens.* **61**, 1–17 (2023).
- McBrearty, I. W. & Beroza, G. C. Earthquake phase association with graph neural networks. *Bull. Seismol. Soc. Am.* **113**, 524–547 (2023).
- Sun, H., Ross, Z. E., Zhu, W. & Azzadenesheli, K. Phase neural operator for multi-station picking of seismic arrivals. *Geophys. Res. Lett.* **50**, e2023GL106434 (2023).
- Hulbert, C. et al. Tremor waveform extraction and automatic location with neural network interpretation. *IEEE Trans. Geosci. Remote Sens.* **60**, 5912009 (2022).
- Shi, Q., Denolle, M. A., Ni, Y., Williams, E. F. & You, N. Denoising offshore distributed acoustic sensing using masked auto-encoders to enhance earthquake detection. *J. Geophys. Res. Solid Earth* **130**, e2024JB029728 (2025).
- Zhou, Y., Yue, H., Kong, Q. & Zhou, S. Hybrid event detection and phase-picking algorithm using convolutional and recurrent neural networks. *Seismol. Res. Lett.* **90**, 1079–1087 (2019).
- Chen, Y., Saad, O. M., Savvaidis, A., Chen, Y. & Fomel, S. 3D microseismic monitoring using machine learning. *J. Geophys. Res. Solid Earth* **127**, e2021JB023842 (2022).
- McBrearty, I. W. & Beroza, G. C. Earthquake location and magnitude estimation with graph neural networks. In *2022 IEEE international conference on image processing (ICIP)*, 3858–3862 (IEEE, 2022).
- McBrearty, I. W. & Beroza, G. C. Double difference earthquake location with graph neural networks. *Earth, Planets Space* **77**, 127 (2025).
- Zhou, Y. et al. An earthquake detection and location architecture for continuous seismograms: phase picking, association, location, and matched filter (PALM). *Seismol. Res. Lett.* **93**, 413–425 (2022).
- Zhou, Y., Ding, H., Ghosh, A. & Ge, Z. AI-PAL: self-supervised AI phase picking via rule-based algorithm for generalized earthquake detection. *J. Geophys. Res. Solid Earth* **130**, e2025JB031294 (2025).
- Zhu, W., Tai, K. S., Mousavi, S. M., Bailis, P. & Beroza, G. C. An end-to-end earthquake detection method for joint phase picking and association using deep learning. *J. Geophys. Res. Solid Earth* **127**, e2021JB023283 (2022).
- Si, X., Wu, X., Li, Z., Wang, S. & Zhu, J. An all-in-one seismic phase picking, location, and association network for multi-task multi-station earthquake monitoring. *Commun. Earth Environ.* **5**, 22 (2024).
- Zhu, W., Song, J., Wang, H. & Münchmeyer, J. Towards end-to-end earthquake monitoring using a multitask deep learning model. *arXiv preprint arXiv:2506.06939* (2025).
- Tan, F. et al. Next generation seismic source detection by computer vision: untangling the complexity of the 2016 Kaikoura earthquake sequence. *J. Geophys. Res. Solid Earth* **129**, e2024JB028735 (2024).
- Kao, H. & Shan, S.-J. The source-scanning algorithm: mapping the distribution of seismic sources in time and space. *Geophys. J. Int.* **157**, 589–594 (2004).
- Winder, T., Bacon, C. A., Smith, J. D., Hudson, T. S. & White, R. S. QuakeMigrate: a Python package for automatic earthquake detection and location using waveform migration and stacking. *Seismica* **5**, 1–22 (2026).

34. Shi, P. et al. MALMI: an automated earthquake detection and location workflow based on machine learning and waveform migration. *Seismol. Res. Lett.* **93**, 2467–2483 (2022).
35. Mohammadigheymasi, H. et al. IPIML: a deep-scan earthquake detection and location workflow integrating pair-input deep learning model and migration location method. *IEEE Trans. Geosci. Remote Sens.* **61**, 5914109 (2023).
36. Liu, Y., Zheng, J., Wang, R., Peng, S. & Shen, S. DMLoc: Automatic microseismic locating workflow based on deep learning and waveform migration. *Seismol. Res. Lett.* **95**, 2997–3007 (2024).
37. Isken, M. P. et al. Qseek: a data-driven framework for automated earthquake detection, localization and characterization. *Seismica* **4**, 1–16 (2025).
38. Turin, G. An introduction to matched filters. *IRE Trans. Inf. Theory* **6**, 311–329 (2003).
39. Gibbons, S. J. & Ringdal, F. The detection of low magnitude seismic events using array-based waveform correlation. *Geophys. J. Int.* **165**, 149–166 (2006).
40. Peng, Z. & Zhao, P. Migration of early aftershocks following the 2004 Parkfield earthquake. *Nat. Geosci.* **2**, 877–881 (2009).
41. Beaucé, E. et al. BPMF: a backprojection and matched-filtering workflow for automated earthquake detection and location. *Seismol. Res. Lett.* **95**, 1030–1042 (2024).
42. Yoon, C. E. & Shelly, D. R. Distinct yet adjacent earthquake sequences near the Mendocino Triple Junction: 20 December 2021 Mw 6.1 and 6.0 Petrolia, and 20 December 2022 Mw 6.4 Ferndale. *Seismic Rec.* **4**, 81–92 (2024).
43. Ho, L. M., Walter, J. I., Hansen, S. E., Sánchez-Roldán, J. L. & Peng, Z. Evaluating automated seismic event detection approaches: an application to Victoria Land, East Antarctica. *J. Geophys. Res. Mach. Learn. Comput.* **1**, e2024JH000185 (2024).
44. Lomax, A., Michelini, A. & Jozinović, D. An investigation of rapid earthquake characterization using single-station waveforms and a convolutional neural network. *Seismol. Res. Lett.* **90**, 517–529 (2019).
45. Mousavi, S. M. & Beroza, G. C. Bayesian-deep-learning estimation of earthquake location from single-station observations. *IEEE Trans. Geosci. Remote Sens.* **58**, 8211–8224 (2020).
46. Zhang, X. & Zhang, M. Universal neural networks for real-time earthquake early warning trained with generalized earthquakes. *Commun. Earth Environ.* **5**, 528 (2024).
47. Zhang, J., Kato, A., Zhu, H. & Zhang, J. Small earthquake location via machine learning with insufficient data. *Earth, Planets Space* **77**, 133 (2025).
48. Feng, Q. & Han, L. Knowledge-augmented deep learning for passive seismic event localization. *Geophysics* **91**, L1–L11 (2026).
49. Zhang, X. et al. Locating induced earthquakes with a network of seismic stations in Oklahoma via a deep learning method. *Sci. Rep.* **10**, 1941 (2020).
50. van den Ende, M. P. & Ampuero, J.-P. Automated seismic source characterization using deep graph neural networks. *Geophys. Res. Lett.* **47**, e2020GL088690 (2020).
51. Zhang, X., Reichard-Flynn, W., Zhang, M., Hirn, M. & Lin, Y. Spatio-temporal graph convolutional networks for earthquake source characterization. *J. Geophys. Res. Solid Earth* **127**, e2022JB024401 (2022).
52. Münchmeyer, J., Bindi, D., Leser, U. & Tilmann, F. Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network. *Geophys. J. Int.* **226**, 1086–1104 (2021).
53. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, 234–241 (Springer, 2015).
54. Zhang, J., Zhu, H., Li, Z., Wu, X. & Zhang, J. Multistation seismic location via machine learning: application to Oklahoma and Southern California. *Geophys. J. Int.* **241**, 1852–1866 (2025).
55. Sun, H. Operator learning for earthquake location: methodology and application to out-of-network events in the Mendocino Triple Junction. *Bull. Seismol. Soc. Am.* **116**, 205–224 (2026).
56. Li, Z. et al. Fourier neural operator for parametric partial differential equations. (ICLR, 2021).
57. Anandkumar, A. et al. Neural operator: graph kernel network for partial differential equations. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations* (ICLR, 2019).
58. Mousavi, S. M., Sheng, Y., Zhu, W. & Beroza, G. C. Stanford earthquake dataset (STEAD): a global data set of seismic signals for AI. *IEEE Access* **7**, 179464–179476 (2019).
59. Kovachki, N. et al. Neural operator: Learning maps between function spaces with applications to PDEs. *J. Mach. Learn. Res.* **24**, 1–97 (2023).
60. Zhu, W., McBrearty, I. W., Mousavi, S. M., Ellsworth, W. L. & Beroza, G. C. Earthquake phase association using a Bayesian Gaussian mixture model. *J. Geophys. Res. Solid Earth* **127**, 023249 (2022).
61. Waldhauser, F. & Ellsworth, W. L. A double-difference earthquake location algorithm: method and application to the northern Hayward fault, California. *Bull. Seismol. Soc. Am.* **90**, 1353–1368 (2000).
62. Münchmeyer, J. et al. Which picker fits my data? A quantitative evaluation of deep learning based seismic pickers. *J. Geophys. Res. Solid Earth* **127**, e2021JB023499 (2022).
63. Ni, Y. et al. A global-scale database of seismic phases from cloud-based picking at petabyte scale. *Seismica* **4**, 1–10 (2025).
64. Hendrycks, D. & Gimpel, K. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415* (2016).
65. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, 1263–1272 (ICML, 2017).
66. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
67. NCEDC. Northern California Earthquake Data Center [Dataset]. UC Berkeley Seismological Laboratory. <https://doi.org/10.7932/NCEDC> (NCEDC, 2014).
68. Sun, H. Quake Neural Operator (QNO) (v1.0.0) [Software]. Zenodo. <https://doi.org/10.5281/zenodo.20072888> (2026).
69. Brown, S. T. et al. Bridges-2: A platform for rapidly-evolving and data intensive research. In *Practice and Experience in Advanced Research Computing 2021: Evolution Across All Dimensions*, PEARC '21 (Association for Computing Machinery, New York, NY, USA, 2021).

Acknowledgements

This work used Bridges-2⁶⁹ at Pittsburgh Supercomputing Center [EES240076] from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation [grant numbers 2138259, 2138286, 2138307, 2137603, and 2138296].

Author contributions

H.S. conceived the study, developed the methodology, performed the data analysis and visualization, and wrote the manuscript.

Funding

H.S. discloses support for the research of this work from the U.S. National Science Foundation (NSF) Collaborations in Artificial Intelligence and Geosciences (CAIG) program [grant number 2530787].

Competing interests

H.S. is an inventor on a pending non-provisional patent application related to the automated microseismic detection and location algorithm described in this study, filed by The University of Texas at El Paso (U.S. Patent Application No. 19/369,710, published as US 2026/0126558 A1, claiming priority to provisional application No. 63/715,422). The author declares no other competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-026-73965-6>.

Correspondence and requests for materials should be addressed to Hongyu Sun.

Peer review information *Nature Communications* thanks Aimé Fournier, Denis Anikiev and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2026