Ⓔ

Hybrid Event Detection and Phase-Picking Algorithm Using Convolutional and Recurrent Neural Networks

by Yijian Zhou, Han Yue, Qingkai Kong, and Shiyong Zhou

ABSTRACT

We developed a hybrid algorithm using both convolutional and recurrent neural networks (CNNs and RNNs, respectively) to pick phases from archived continuous waveforms in two steps. First, an eight-layer CNN is trained to detect earthquake events from 30-second-long three-component seismograms. The event seismograms are then sent to a two-layer bidirectional RNN to pick *P*- and *S*-arrival times. The data for training and validation and testing of the networks are obtained from the continuous waveforms of 16 stations recording the aftershock sequence of the 2008 Wenchuan earthquake. The augmented training set has 135,966 *P*–*S*-wave arrival-time pairs. The CNN achieved 94% and 98% hit rate for event and noise segments in the test set, respectively. The RNN picking accuracies for *P* and *S* waves are -0.03 ± 0.48 (mean error ± standard deviation) and 0.03 ± 0.56 s, respectively.

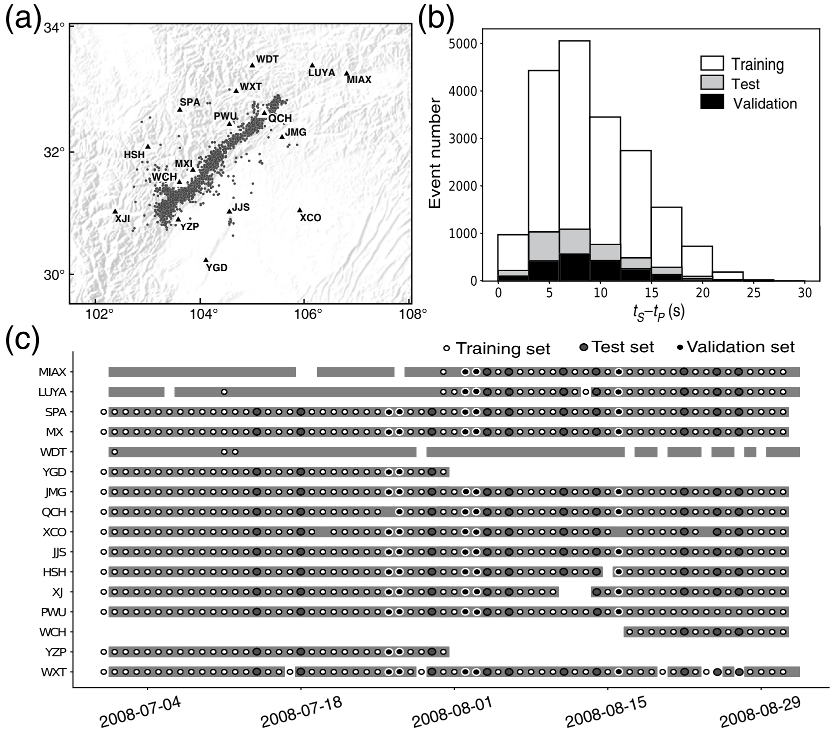
Supplemental Content: Detailed description of operations of the convolutional layer and gated recurrent unit (GRU) cell.

INTRODUCTION

Earthquake event detection and phase picking form the first step of determining earthquake location and building a catalog, which is fundamentally important to earthquake studies. The manual phase-picking procedure is generally composed of detecting earthquake signals from continuous waveforms (detection) and zooming in to precisely pick arrival times (phase picking). For a long time, this process has been realized with either labor intensive manually inspection or low-accuracy automatic algorithms. Traditional automatic algorithms such as short-term average over long-term average (STA/LTA; Allen, 1978) meet difficulties when applying to records from different stations and with different signal-to-noise ratios (SNRs; Withers *et al.*, 1998). Applying STA/LTA to continuous waveforms produces a large number of false detections. More advanced methods such as the matched filter technique (Turin, 1960; Gibbons and Ringdal, 2006) require a predetermined catalog to construct template sets and run with high computational costs, which is more suitable for relocation purposes instead of real-time phase picking. Machine-learning (ML) algorithms have been applied to earthquake detection since the 1990s (e.g., Joswig, 1990), with the advantage of achieving generalized detection for waveforms that have not been seen before. (See further discussions of ML-based detection algorithms in Kong *et al.*, 2018.) Supervised deep-learning algorithms are recently developed in various fields and applied to earthquake detection problems, achieving promising results. Such data-driven methods are particularly suitable for problems that are difficult to express through mathematics, but can be easily handled by human experiences. Neural networks can be trained by tremendous labeled examples such as manually picked *P*/*S* phase. Perol *et al.* (2018) trained convolutional neural networks (CNNs) to detect and roughly locate the induced seismic events in Oklahoma, United States. Their networks reach a similar detection completeness as that of autocorrelation (Brown *et al.*, 2008) and the Fingerprinting and Similarity Thresholding algorithm (Yoon *et al.*, 2015) but with a much lower time cost after the model is trained. Ross *et al.* (2018) applied a CNN to *P*- and *S*-wave phase detection and picking problems using a large amount of training data. Mousavi *et al.* (2018) combined a CNN and a recurrent neural network (RNN) as a single network for detection purposes and achieved high-detection accuracy with a low rate of false positives. Zhu and Beroza (2019) creatively treat the phase picking as a classification problem for each data point and convert the phase picking into phase probability distributions using a CNN. This model picks *P* and *S* waves at high accuracy for various types of instruments and different data quality.

Most of the mentioned deep-learning algorithms cover either signal detection or phase-picking problems, although these two tasks need to be integrated to pick phase arrival times from continuous waveforms. In this article, we propose continuous waveform phase picking with a sequential application of deep-learning algorithms for event detection and phase-picking purposes, respectively. The logic of algorithm design

doi: 10.1785/0220180319



▲ **Figure 1.** (a) Seismic stations are plotted as black filled triangles. Wenchuan aftershocks are plotted as black filled dots. (b) Histograms of event number at each signal duration ($t_S–t_P$ time). Distribution of training set, test set, and validation set are plotted as white-, gray-, and black-filled bars, respectively. (c) The data continuity of each station is plotted as gray bars. Daily waveforms used for training, validation, and test purposes are plotted as white-, gray-, and black-filled dots, respectively.

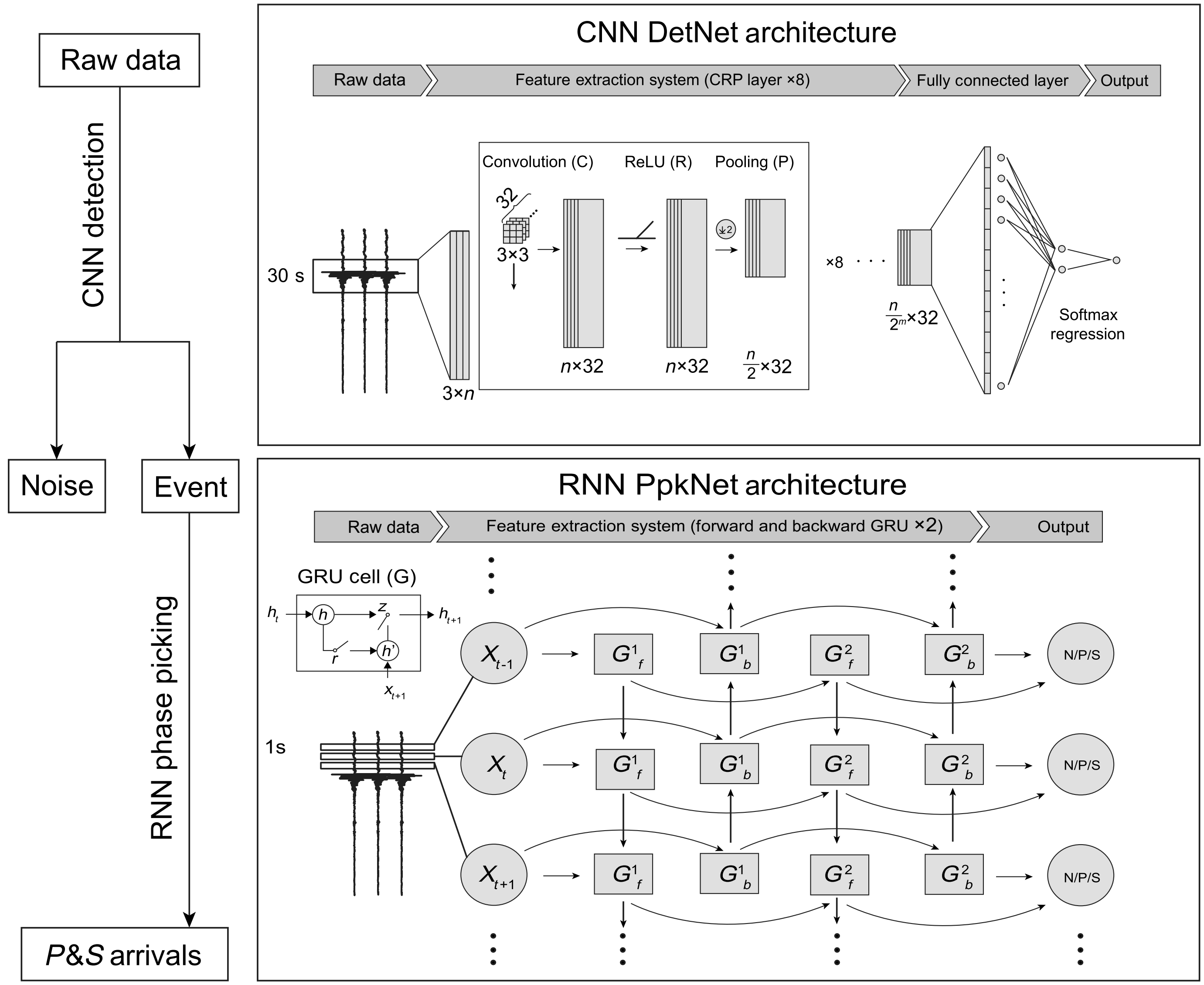
is stated as follows. Seismic signals are a combination of both nonsequential and sequential signals. Three components of seismic signals sample ground motions in different directions resembling the RGB channels of images. The temporal variation and causal relationship of seismic signals draw important constraints on phase detection, resembling word recognition of a speech. For example, P wave always proceeds S waves, and thus information on S-wave arrival can help detecting P wave; the shape of coda waves is important to identify phase type and to mark the initial arrivals. Such mixed behavior of seismic signals inspired us to use CNNs and RNNs in event detection and phase picking, just as their common applications for image and speech recognition problems. With such logic, we propose a hybrid method composed of two separate neural networks, exploiting the advantage of both algorithms.

DATASETS

We used data from the aftershock monitoring project of the 2008 M_w 7.9 Wenchuan earthquake. Continuous waveforms are made available through the artificial intelligence phase detection competition of Alibaba (see Data and Resources). The seismic network is composed of 16 broadband three-component seismic stations covering a spatial range of ~500 km × 400 km (Fig. 1) and a time span of approximately two months (30 June–31 August), originally sampled at 100 Hz. About 25,487 manually picked P- and S-wave arrival-time pairs were provided by the competition (Zheng *et al.*, 2010) as labels for training. These manual picks are not complete through the recording period. We divide the 61-day data randomly into three sets, including 46 days, 5 days, and 10 days of data as training, validation, and test sets (Fig. 1c), respectively, including 19,489, 1981 and 4008 P- and S-wave arrival pairs. We calculated distributions for differential arrival times of P and S waves ($t_S–t_P$) for each set and found similar distributions (Fig. 1b). The P and S arrival-time difference resembles seismic signal duration, which is mostly less than 15 s in our dataset. The sliced time window is supposed to cover both P and S phases, which are the main features that human inspectors use in earthquake detection. Thus, we chose a 30-second-long time window to cut original seismograms with the P arrival on the first half to form segmented training data. Such operation ensures (1) both P and S arrivals can be covered by each time window, and (2) signal covers a significant portion in each time window. To build the event labeled data in the training set, we cut the original data around the manually picked P arrivals. The training data are augmented by both random selection of initial times from 0 to 15 s before the P arrivals and adding different levels of Gaussian noise to serve the purpose of adding in more variations for better generalization and increasing the training dataset (the benefit of this data augmentation is shown in the Ⓔ supplemental content to this article). Noise-labeled data in the training set is made by random selection of time windows not overlapping with seismic signals. Because the manual picks are not complete, the noise sets for training are possible to contain a few event samples, but we expect these account for just a very small portion. We finally generated similar amount of event (135,966) and noise (140,197) data segments for the training set. Before being fed into the neural networks, the stream data were preprocessed by (1) removing the mean value and linear trend with respect to the common operation made before manual picking; (2) normalizing to constrain the absolute values between 0 and 1 considering the absolute amplitude is not important in manual picking; and (3) applying a high-pass filter of 1 Hz to avoid low-frequency information, which are more significantly influenced by instrument drifts.

NEURAL NETWORK DESIGN

A hybrid method is used to achieve event detection and phase picking. First, we treat the event detection as a classification problem, in which we trained the CNN (detection network [DetNet]) to classify the input time windows as event or noise.



▲ **Figure 2.** The workflow of hybrid detection and phase-picking algorithm. Event detection network (DetNet) is made through convolutional neural network (CNN). Then seismograms labeled as events are sent to the recurrent neural network (RNN) for P/S phase-picking network (PpkNet). Architecture of both neural networks is plotted in two boxes on the right (see Ⓔ supplemental content to this article for more details). In the bottom box, the detailed structure of gated recurrent unit (GRU) cell (G) is plotted, in which x_t and h_t represent the input and hidden state of each time (t), respectively. $G^i_{f/b}$ represents the forward/backward GRU of the ith layer. N/P/S refers to label of noise/after P/after S; ReLU, rectified linear unit.

Second, we treat the phase picking as a sequential labeling problem, in which an RNN (phase-picking network [PpkNet]) was applied to label each sequential 1 s frame of events detected from the DetNet to noise, *P* wave, or *S* wave (Fig. 2). The detailed architecture of the DetNet and PpkNet is described in the following sections.

DetNet Design Using CNN

A convolutional layer in a CNN consists of three operations: convolution, activation, and max-pooling (Fig. 2). The convolution operation achieves filtering and a linear combination of the input data. We choose rectified linear unit (ReLU; Nair and Hinton, 2010) as the activation layer applied elementwisely on the output of convolution. ReLU outputs are then modified by max pooling over every two samples in the temporal dimension to downsample the data. The three operations enable the network to detect objects in different locations of the input, with slight input variations (LeCun *et al.*, 2015). We used eight convolutional layers in the DetNet, with each layer containing 32 convolutional kernels of size 3. The first layer converts the input three-channel waveforms to 32 feature maps, and the same number of feature maps is maintained in the network. We use a fully connected layer to summarize the output of the last convolutional layer and calculate the probability using the softmax function (Fig. 2). The class with the higher score is the prediction.

The PpkNet requires both *P* and *S* waves present in a time window. For event detection from continuous waveforms, we use a 15 s overlapped time window to cut original waveforms and feed waveform segments to the DetNet. This approach makes sure at least one window covers both *P* and *S* arrivals. If two consecutive windows are predicted as events, the one with higher classification score is considered to contain more complete waveforms and thus is used as the candidate for the PpkNet.

PpkNet Design Using RNN

RNNs model time series by recurrent hidden states, which are updated through time. In each RNN layer, we model the output classification probability using gated recurrent units (GRUs; Cho *et al.*, 2014; Chung *et al.*, 2014), which enables efficient training. We cut 30-second-long three-component seismograms into 59 sequential frames with 1 s duration and 0.5 s stride and concatenate the three-component segments into a 1D vector with 300 data points. We used two sequential forward and two backward layers in the PpkNet, in which GRU cells read the frames in either direction to capture contextual information. The GRU cells of each layer have the identical structure, and each contains 64 units. The outputs of the final RNN layers are sent to a fully connected layer with softmax activation functions. Each timestep (the 1 s frame) is classified as one of three classes: noise (before *P* arrival), *P* wave (between *P* and *S* arrivals), or *S* wave (after *S* arrival) based on the classification score given by the softmax normalization. Because the waveform segments are 50% overlapped, two segments containing the input arrival are both labeled as after arrival, the true arrival locates at the latter half (>0.5 s) of the first labeled segment. Thus, we choose the arrival time to be 0.75 s after the initial time of the first labeled segment as the arrival time, which ensures theoretical maximum picking accuracy of <0.25 s.

Training Strategies

Both CNNs and RNNs are trained by a minibatched stochastic gradient descent algorithm. The optimizer updates the weights of the neural networks based on the gradient of a loss function. After each iteration, samples in the training set are shuffled, and the whole process is repeated. The loss function for the DetNet is defined as the cross entropy with L2 regularization (Ng, 2004). The loss function for the PpkNet is defined as the average cross entropy for all timesteps. All weights were initialized by the Xavier initializer (Glorot and Bengio, 2010) and optimized by the Adam optimizer (Kingma and Ba, 2015). We constructed our model using Tensorflow (Abadi *et al.*, 2016) and run the training on two Nvidia GTX 1080ti graphic processing units.

We adopt regularization in the CNN training to prevent overfitting, which includes:

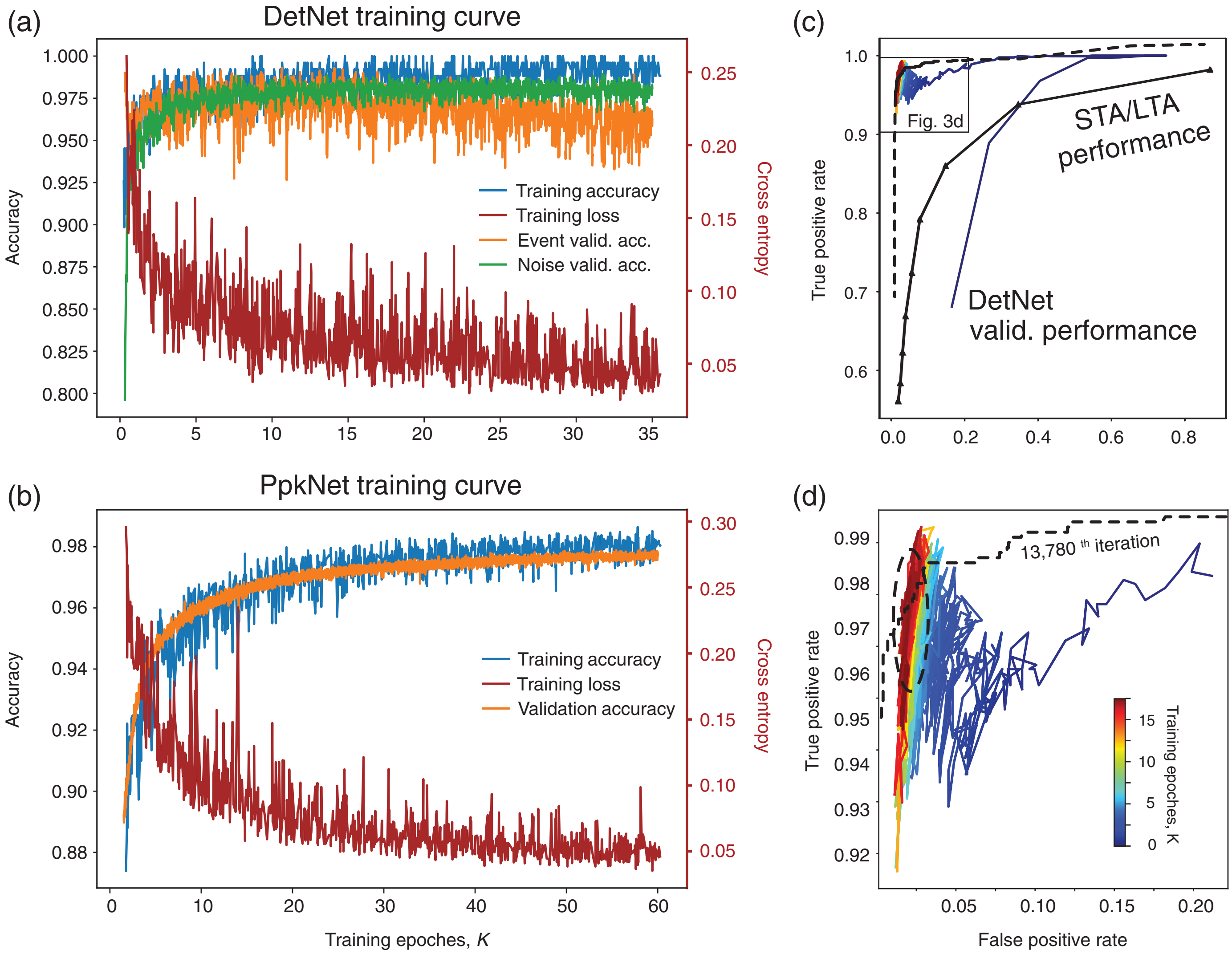
1. early stopping based on validation set (Prechelt, 1998) to stop the training process;
2. L2 regularization (Ng, 2004) to penalize the loss function and the absolute value of weighting numbers; and
3. batch normalization (Ioffe and Szegedy, 2015) to reduce the distribution shift from shallow to deep layers by normalizing each convolutional outputs.

RESULTS AND DISCUSSION

Training Progress and Performance Validation

We measured the classification accuracy in the training and validation process of both the DetNet and PpkNet, as plotted in Figure 3. The training accuracy for the DetNet converged to more than 99% after about 25,000 iterations (Fig. 3a), and the whole process lasted for ∼1 hr. The validation accuracy reached a peak value of about 97% and 98% for event and noise, respectively, and started to drop after about the 15,000th iteration (Fig. 3a) indicating the onset of overfitting. We used early stopping and chose the 13,780th checkpoint for further tests, the validation accuracies of which are 96.8% and 98.1% for events and noise, respectively. The training accuracy for the PpkNet converged to more than 98% after 40,000 iterations (Fig. 3b), which took a training period of ∼2 hr. Although the validation accuracy is lower than training (∼97.5%), it did not show a decreasing trend (Fig. 3b), indicating no overfitting. Thus, we simply chose the latest checkpoint for further tests.

We plotted the true-positive rate versus false-positive rate of the DetNet on the validation set at different training iterations to evaluate the training progress and final performance of the DetNet (Fig. 3c,d). The true-positive rate is defined as the ratio between correctly detected events over the number of event samples (detection rate), and the false-positive rate (false detection rate) is defined as the ratio of noise samples that falsely classified as events over the number of noise samples. A good algorithm achieves high positive rate at low false-positive rate; thus, an ideal detection point locates on the top-left corner in the figure regime. As a comparison, we also plot the performance of the 13,780th checkpoint versus that of the STA/LTA algorithm over the same dataset at different thresholds (Fig. 3d), which resembles a modified receiver operating characteristic (ROC) curve. We modify the ROC curve to resemble completeness–redundancy relationship (Yue *et al.*, 2018) because seismologists are concerned more with the correctly picked and mispicked events than about noise. This modification also enables a direct comparison with an STA/LTA curve (Fig. 3c). The performance of the DetNet initially locates beneath the STA/LTA curve because the training starts from a random initial point. The performance curve quickly moves to the top-right corner and gradually moves to the top-left corner, demonstrating that the training initially classifies all data as events achieving both high true- and false-positive rates. Then noise data are gradually excluded from the event class, reducing false positives. The final training stages (after 15,000 iterations) stabilized at 2%–3% of the false-positive rate and 94%–97% of the true-positive rate (event detection rate), which appears to be the optimized performance of the DetNet. This event detection rate is significantly higher than that (∼60%) of the STA/LTA performance. The



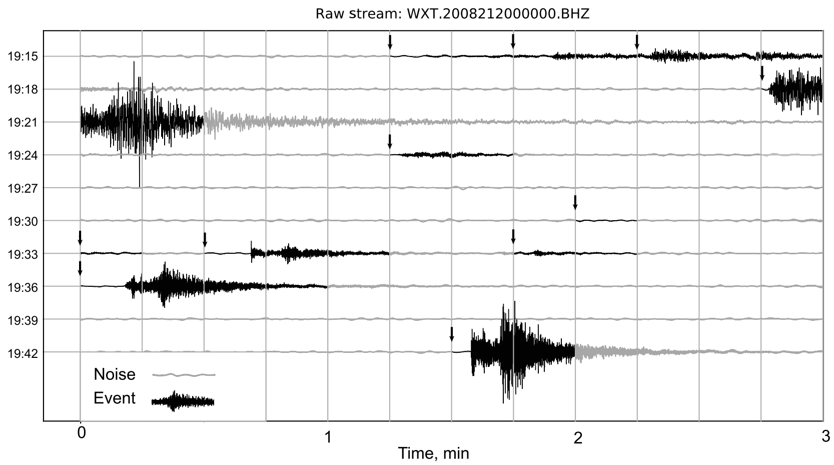
▲ **Figure 3.** (a,b) The training accuracy and loss function curves are plotted in blue and brown, respectively. The validation accuracy of the DetNet for event and noise is plotted in orange and green, respectively. The validation accuracy for the PpkNet is plotted in orange. (c) The performance of the DetNet is plotted as a colored curve with the color denoting different training iterations. The modified receiver operating characteristic (ROC) curve of a short-term average/long-term average (STA/LTA) algorithm is plotted as a black curve as a reference. The modified ROC curve at 13,780th iteration is plotted as a black dashed curve. (d) A zoom-in view of the black box in (c). The steady-state performance (after 30,000 training) is marked as a dashed ellipse.

modified ROC curve of the 13,780th iteration also performs much better than that of the STA/LTA curve. As a performance example of the DetNet, a segment of continuous waveform in the test dataset is labeled by the DetNet and plotted in Figure 4. It shows earthquake signals with different durations and amplitudes are correctly labeled by the DetNet. An example of PpkNet outputs is plotted in Figure 5 for events at different epicentral distances. It shows that the correct picking of *P* and *S* phases is not significantly and systematically influenced by epicentral distance or the onset of the *P*-wave when both phases are presented in the time window. Even when *P* arrivals at the second half of the time window (as in the third line of Fig. 5), which is not included in the training set, the PpkNet gives reasonable results.

Tests on Detection and Picking Performance

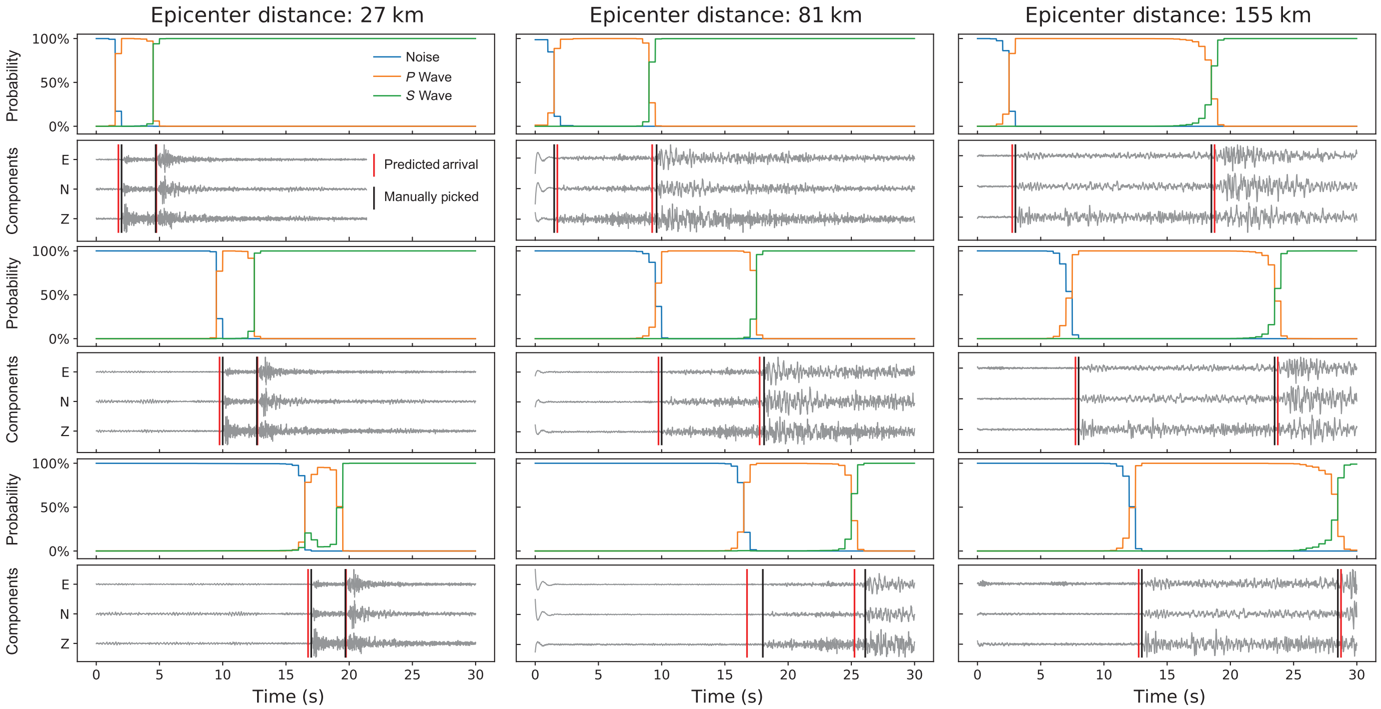
The performance of the optimized DetNet is evaluated by true-positive rate, as defined earlier. We applied the DetNet on continuous waveforms in the testing data and calculated the histogram for the true-positive rates of detection for events at different epicentral distances. The DetNet achieved similar true-positive rate for each epicentral distance range and different station (Fig. 6a) and high true-negative rates (noise detection rate $>$ 97%), which ensures a low false-positive (false detection) rate. The event detection rate appears to be correlated with the SNR (Fig. 6c). In summary, about 95% of manually labeled events can be detected by the DetNet.

We tested the performance of the PpkNet by cutting off time windows around the manual picks in the testing days and

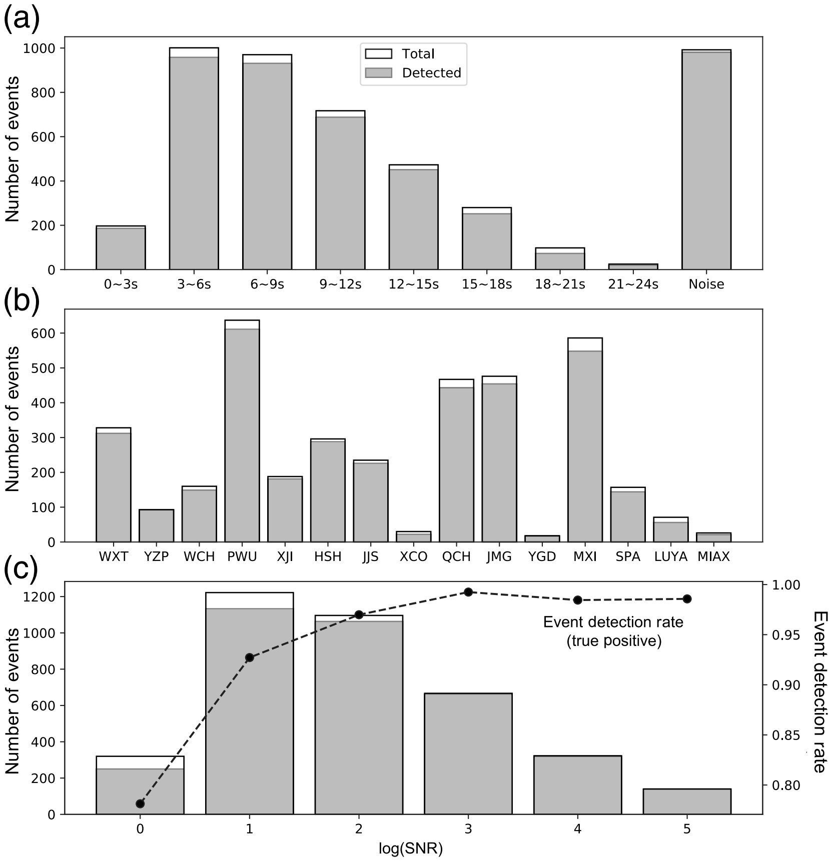


▲ **Figure 4.** A sample of event detection in continuous waveform. Waveform segments detected as event and noise are plotted as black and gray colors, respectively. The initial times of event segments are pointed out by the black arrows.

apply the PpkNet to these time windows. Phase-picking errors are defined by the time differences between the PpkNet and manual picks. We also used two traditional algorithms to pick *P* and *S* phases to compare with the PpkNet outputs. We used STA/LTA to pick *P* arrivals from *Z*-component records and to pick *S* arrival from polarized-motion-filtered three-component waveforms (Ross and Ben-Zion, 2014). We conducted three sets of operation by (1) applying traditional algorithms directly to the whole time window (30 s), (2) applying the PpkNet to the whole window, and (3) applying traditional algorithm to a refined time window (±2 s of the PpkNet picks) to evaluate the accuracy and stability of the PpkNet. Detailed description of phase-picking performance comparison is provided in the Ⓔ supplemental content. The algorithm performances are summarized in Table 1, which clearly shows the traditional algorithms produce high false detection (error > 5 s) rates when applying to the whole time window. This artifact is caused by identifying impulsive noise as earthquake signals, which could be more significant when applying traditional algorithms to continuous waveforms (Yue *et al.*, 2018). The PpkNet produces high detection accuracy for both *P* and *S* phases, demonstrating that it is a stable and accurate phase-picking algorithm. When applying a traditional algorithm to a refined time window near the PpkNet picks (PpkNet + Algorithm), the *P*-phase picking accuracy is not significantly improved, but *S*-wave picking performance is reduced. Figure 7 shows that the traditional algorithm not only has large errors of *S*-phase picking but also produces systematic arrival-time shifts for signals with different duration. This error introduces systematic location errors for



▲ **Figure 5.** Examples of PpkNet performance for different epicentral distances are plotted in each column, respectively. Probability functions and waveforms of early, middle, and late *P*-wave arrivals are plotted in each row, respectively. In each set of panels, probability functions of noise, *P* wave, and *S* wave are plotted as blue, orange, and green curves in the upper layer. Original waveforms and arrival times are plotted in the lower layer. The labeled and predicted arrival times are marked as black and red lines, respectively. E, N, and Z refers to east, north, and vertical components, respectively.



▲ **Figure 6.** (a) Event and noise detection performances are evaluated for signals with different duration. Number of manually picked and DetNet detected events and noises are plotted as white and gray bars. (b) Event detection performance is evaluated for each station using the same setting of (a). (c) Event detection performance is evaluated for segments with different signal-to-noise ratios (SNRs). Event detection rate is plotted with a dashed curve using the axis on the right.

events from differential epicentral distances. Such comparison clearly shows the PpkNet outperforms traditional algorithms in *S*-wave picking accuracy as also picking stability for both *P* and *S* waves.

The PpkNet picking accuracy is comparable for *P* and *S* phases, although manually picked *P* arrivals are more accurate (~0.1 s) than *S* waves, which is much smaller than the PpkNet. The similar PpkNet pick uncertainty for *P* and *S* waves may be caused by the accuracy upper limit (0.25 s) introduced by waveform segmentation (1 s) and stride (0.5 s). Further tests need to be made if finer segments could improve the *P*-wave picking accuracy.

CONCLUSION

In this article, we introduced a hybrid algorithm using both CNNs and RNNs to detect events and pick phases from continuous waveforms. The hybrid algorithm outperforms traditional algorithms in picking stability (low false detection) and high picking accuracy. This hybrid approach appears to be a promising direction to replace human picking from continuous waveforms. It is also possible to improve the performance of real-time techniques, for example, the earthquake early warning system, which relies on traditional picking approaches.

DATA AND RESOURCES

Waveform data are provided by the Data Management Center (DMC) of the China Earthquake Administration (CEA; Zheng *et al.*, 2010). Related software package is accessible on the GitHub via https://github.com/YijianZhou/CDRP_TF (last accessed February 2019).

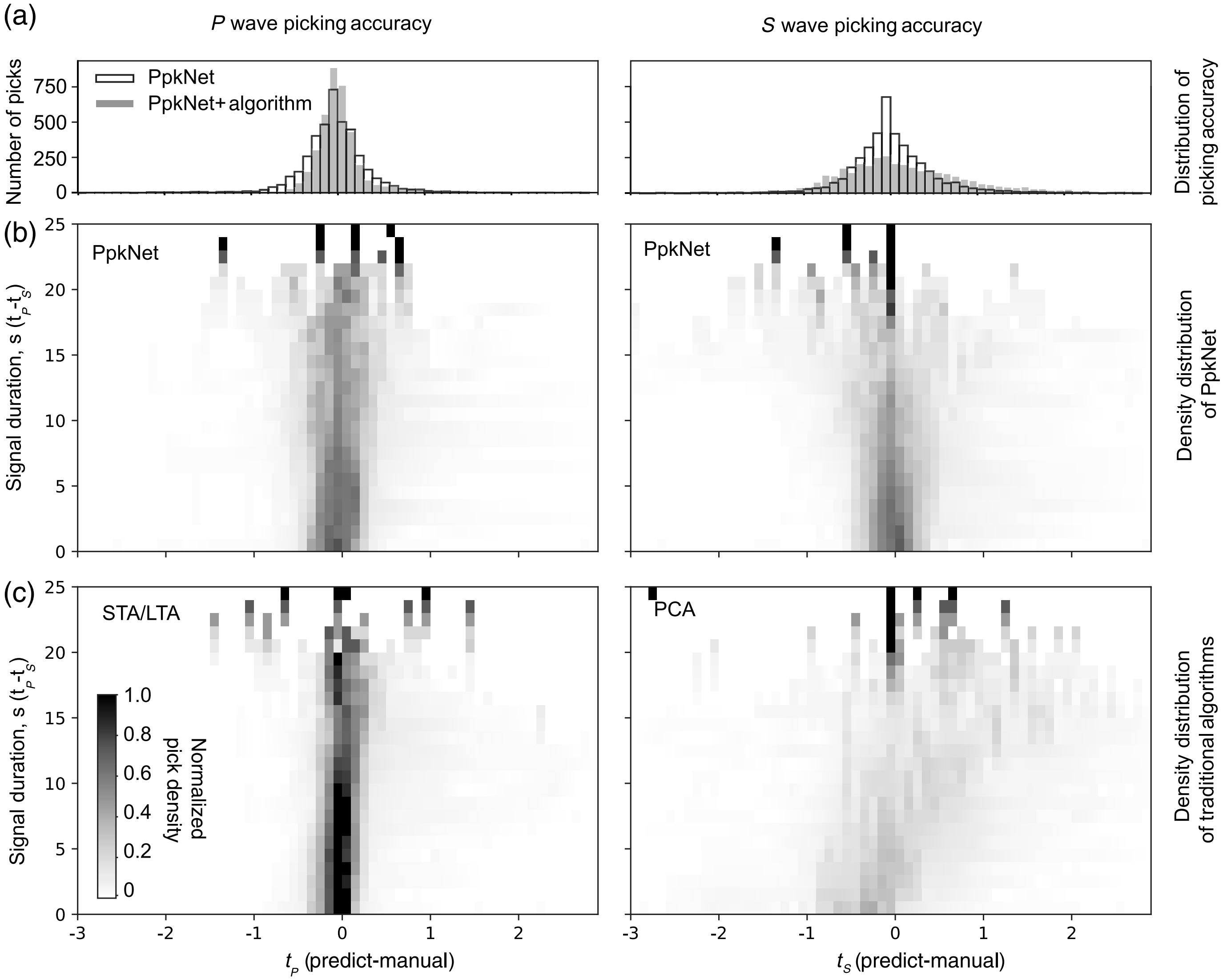
ACKNOWLEDGMENTS

Deep-learning neural network training uses TensorFlow package (Abadi *et al.*, 2016). Seismic data preprocessing uses Seismic Analysis Code (SAC). The authors thank Richard Allen for providing this opportunity of collaboration and substantial suggestion. This research is supported jointly by the National Key R&D Program of China (Grant Number 2018YFC1503401), China Earthquake Science Experiment Project (Grant Number 2016 CESE 0104), and National Science Foundation of China (NSFC) projects (Grant Number 41674047).

Table 1
Comparison of Pickers

	P-Wave Arrival-Time Error (Mean ± St. Dev., s)	*S*-Wave Arrival-Time Error (Mean ± St. Dev., s)	*P*-Wave False Detection (Error > 5 s) (%)	*S*-Wave False Detection (Error > 5 s) (%)
Traditional algorithms	0.10 ± 0.63	−0.33 ± 1.67	6.4	15.8
PpkNet	−0.03 ± 0.48	0.03 ± 0.56	0.26	0.7
PpkNet + algorithm	0.06 ± 0.47	0.14 ± 0.91	0.20	0.68

PpkNet, phase-picking network.



▲ **Figure 7.** Distribution of *P*- and *S*-phase picking errors are plotted in each column, respectively. (a) Picking error histograms of PpkNet and traditional algorithms are plotted as blank and gray filled bars, respectively. (b,c) Probability density in the detection error and signal duration domain is plotted in a gray color scale. Phase-picking performance of PpkNet and traditional algorithms are plotted in (b) and (c), respectively. PCA, principal component analysis.

REFERENCES

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.* (2016). Tensorflow: A system for large-scale machine learning, paper presented at the *Proc. of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2016)*, Savannah, Georgia, 2–4 November 2016.

Allen, R. V. (1978). Automatic earthquake recognition and timing from single traces, *Bull. Seismol. Soc. Am.* **68,** no. 5, 1521–1532.

Brown, J. R., G. C. Beroza, and D. R. Shelly (2008). An autocorrelation method to detect low frequency earthquakes within tremor, *Geophys. Res. Lett.* **35,** no. 16, doi: 10.1029/2008GL034560.

Cho, K., B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation, *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing*, Doha, Qatar, October, 1724–1734.

Chung, J., C. Gulcehre, K. Cho, and Y. Bengio (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling, available at https://arxiv.org/abs/1412.3555 (last accessed February 2019).

Gibbons, S. J., and F. Ringdal (2006). The detection of low magnitude seismic events using array-based waveform correlation, *Geophys. J. Int.* **165,** no. 1, 149–166, doi: 10.1111/j.1365-246X.2006.02865.x.

Glorot, X., and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks, paper presented at the *Proc. of the Thirteenth International Conf. on Artificial Intelligence and Statistics*, Sardinia, Italy, 249–256.

Ioffe, S., and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift, available at https://arxiv.org/abs/1502.03167 (last accessed February 2019).

Joswig, M. (1990). Pattern recognition for earthquake detection, *Bull. Seismol. Soc. Am.* **80,** no. 1, 170–186.

Kingma, D. P., and J. L. Ba (2015). Adam: A method for stochastic optimization, available at https://arxiv.org/abs/1412.6980 (last accessed February 2019).

Kong, Q., D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, and P. Gerstoft (2018). Machine learning in seismology: Turning data into insights, *Seismol. Res. Lett.* **90,** no. 1, 3–14.

LeCun, Y., Y. Bengio, and G. Hinton (2015). Deep learning, *Nature* **521,** no. 7553, 436.

Mousavi, S. M., W. Zhu, Y. Sheng, and G. C. Beroza (2018). CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection, available at https://arxiv.org/abs/1810.01965 (last accessed February 2019).

Nair, V., and G. E. Hinton (2010). Rectified linear units improve restricted Boltzmann machines, paper presented at the *Proc. of the 27th International Conf. on Machine Learning (ICML-10)*, Haifa, Israel, 21–24 June 2010, 807–814.

Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance, paper presented at the *Proc. of the Twenty-First International Conf. on Machine Learning*, Banff, Alberta, Canada, 4–8 July 2004, 78 pp.

Perol, T., M. Gharbi, and M. Denolle (2018). Convolutional neural network for earthquake detection and location, *Sci. Adv.* **4,** no. 2, doi: 10.1126/sciadv.1700578.

Prechelt, L. (1998). Early stopping—But when? in *Neural Networks: Tricks of the Trade*, G. B. Orr and K. R. Müller (Editors), Springer-Verlag, London, United Kingdom, 55–69.

Ross, Z. E., and Y. Ben-Zion (2014). Automatic picking of direct P, S seismic phases and fault zone head waves, *Geophys. J. Int.* **199,** no. 1, 368–381.

Ross, Z. E., M. A. Meier, and E. Hauksson (2018). *P*-wave arrival picking and first-motion polarity determination with deep learning, *J. Geophys. Res.* **123,** no. 6, 5120–5129.

Turin, G. (1960). An introduction to matched filters, *IRE Trans. Inform. Theory* **6,** no. 3, 311–329.

Withers, M., R. Aster, C. Young, J. Beiriger, M. Harris, S. Moore, and J. Trujillo (1998). A comparison of select trigger algorithms for automated global seismic phase and event detection, *Bull. Seismol. Soc. Am.* **88,** no. 1, 95–106.

Yoon, C. E., O. O'Reilly, K. J. Bergen, and G. C. Beroza (2015). Earthquake detection through computationally efficient similarity search, *Sci. Adv.* **1,** no. 11, e1501057.

Yue, H., Y. Zhou, S. Zhou, Y. Huang, M. Li, L. Zhou, and Z. Liu (2018). The 2017 Jiuzhaigou earthquake aftershock-monitoring experimental network: Network design and signal enhancement algorithm, *Seismol. Res. Lett.* **89,** no. 5, 1671–1679.

Zheng, X.-F., Z.-X. Yao, J.-H. Liang, and J. Zheng (2010). The role played and opportunities provided by IGP DMC of China National Seismic Network in Wenchuan earthquake disaster relief and researches, *Bull. Seismol. Soc. Am.* **100,** no. 5B, 2866–2872, doi: 10.1785/0120090257.

Zhu, W., and G. C. Beroza (2019). PhaseNet: A deep-neural-network-based seismic arrival-time picking method, *Geophys. J. Int.* **216,** no. 1, 261–273, doi: 10.1093/gji/ggy423.

Yijian Zhou
Han Yue
Shiyong Zhou
School of Earth and Space Sciences
Peking University
5 Yiheyuan Road
Beijing 100871, China
yue.han@pku.edu.cn

Qingkai Kong
Berkeley Seismological Laboratory
University of California, Berkeley
209 McCone Hall
Berkeley, California 94720 U.S.A.

Published Online 10 April 2019