DynaPicker: Dynamic Convolutional Neural Networks for Seismic Phase Classification and Arrival-time Picking

Wei Li, Jonas Köhler, Megha Chakraborty, Georg Rümpker, Horst Stöcker, Nishtha Srivastava

Abstract—Seismic phase picking is at the core of earthquake monitoring. To date, the advancements in earthquake data collection have led to an exponential increase in the volume of available seismic waveform data. This highly necessitates a reliable solution for seismic phase classification. Advanced technologies, such as convolutional neural networks, have been widely introduced to create earthquake catalogs from continuous waveforms produced using conventional methods. However, their performance is restricted by the static convolution kernels. To cope with this challenge, a dynamic convolutional neural network-based framework, termed DynaPicker, is proposed in this study for detecting seismic body wave phases that allows a dynamic inference with the deep learning architecture. We demonstrate the performance of our framework on two tasks: seismic phase classification and arrival-time picking, both tested on four open-source seismic datasets. The results of our experiments show that DynaPicker can yield a testing accuracy of 98.82% in seismic phase identification. We demonstrate the robustness of this method’s ability in classifying seismic phases, even when the low-magnitude seismic data is polluted by noise. Moreover, DynaPicker can be extended to deal with input data of varying lengths for seismic phase detection. Given continuous seismic data, DynaPicker can correctly identify more seismic events and produce a lower arrival time picking error compared to the baseline methods.

Index Terms—Dynamic convolutional neural network, Seismic phase identification, Seismic phase picking

I. INTRODUCTION

Seismic phase picking, which plays an essential role in earthquake location identification and body-wave travel time tomography, is often performed manually. In order to achieve adequately automated seismic phase picking, many conventional approaches have been studied over the past few decades. Common algorithms developed for seismic phase picking include short-time average/long-time average (STA/LTA) [1] and Akaike information criterion (AIC) [2]. The STA/LTA is mathematically formulated as the ratio of the average amplitude over a short time window to the average amplitude over a long time window. In STA/LTA, an event is detected when the ratio is greater than the defined threshold. The AIC solution is subject to the assumption that the seismogram can be split into auto-regressive (AR) segments, where the minimum AIC value is usually defined as the arrival time. However, neither STA/LTA nor AIC can achieve satisfactory performance for low signal-to-ratio (SNR) signals.

The past decades have witnessed a sharp increase in the amount of available seismic data owing to the advancement of seismic equipment and the expansion of seismic monitoring networks. This has increased the demand for a robust seismic phase picking method to deal with large volumes of seismic data. Deep learning has the merit of facilitating the processing of large amounts of data and extracting its features which makes it successful in diverse areas, especially in image processing [4]. The implementation of seismic phase picking can be considered similar to object identification in computer vision. Thus, the use of deep learning has been widely embraced in first-motion polarities identification of earthquake waveforms [5], seismic event detection [6]–[9], earthquake magnitude classification and estimation [10]–[12], and seismic phase picking [13]–[17]. The authors in the work [18] stated that seismic phase picking approaches can be roughly divided into two main streams, continuous seismic waveform-based and small window-format-based methods. The former is to process continuous seismic waveforms like earthquake-length windows of fixed duration with more complex triggers. The output of this type of model is the probability distribution over the fixed window length. The latter is to split the seismic waveform into small windows (e.g., 4 – 6 s [13]), where only one centered pick or noise is included. Then, each window is identified as one of three classes: P-wave, S-wave, and noise. Stepnov et al., (2021) [18] concluded that for the former scenarios those models can work well when scanning archives, whereas it is not suitable for real-time processing because of the restriction imposed by the required input window length. On the contrary, considering that the ground motion data are constantly received in small chunks, processing small windows of the long waveform data renders it more suitable for real-time seismic applications [18]. As a result, the length of the long waveform can be formed by sequentially adding the successive chunk to the previous continuous data, and each chunk could be directly fed into the pre-trained model for class identification.

Most deep learning-based seismic phase classification model architectures largely rely on convolutional neural networks (CNN). CNN is capable of extracting meaningful fea-
Fig. 1. Schematic diagram for the proposed Dynamic Convolution Decomposition (DCD) based model. The model architecture represented on the left side includes a convolutional layer, 1D-DCD layers, and the classifier. The 1D-DCD block displayed on the right side is the backbone of the 1D-DCD layer, which is adapted from the work of Li et al. [3] and converted into the 1D case in this study. In a 1D-DCD block, the input \( x \) first goes through a dynamic branch to generate \( \Lambda(x) \) and \( \Phi(x) \), and then to produce the convolution matrix \( W(x) \) using equation (3).


tures from the input data, which enables the neural network to achieve a good performance. However, most of the prevalent CNN-based models perform inference using static convolution kernels, which may limit their representation power, efficiency, and ability for interpretation. To cope with this challenge, dynamic convolution [19] is proposed by aggregating parallel convolution kernels via attention mechanism [20]. Compared to static models, which have fixed computational graphs and parameters at the inference stage, dynamic networks can adapt their structures or parameters to different inputs, leading to notable advantages in terms of accuracy, computational efficiency, adaptiveness, etc. [21]. However, it is challenging to jointly optimize the attention score and the static kernels in dynamic convolution. To mitigate the joint optimization difficulty, Li et al. [3] revisited it from the matrix decomposition perspective by reducing the dimension of the latent space.

In this work, we pioneer a novel deep learning-based solution, titled DynaPicker, for seismic body wave phase classification. Furthermore, the phase classifier trained on the short-window data is used to estimate the arrival times of the P-wave and S-wave on the continuous waveform in a long-time scale. In DynaPicker, the 1D dynamic convolution decomposition (DCD) adapted from the work of [3] is used as the backbone of the solution (see Fig. 1 for illustration).

In order to complete seismic body wave phase classification, and phase onset time picking, the main steps in this study are included as follows. First, the impact of different input data lengths on the performance of seismic phase detection and arrival time picking are studied on the subset of the STanford EArthquake Dataset (STEAD) [22]. Then, the SCEDC dataset [23] without specific phase arrival-time labeling collected by the Southern California Seismic Network is used to train and test the model in seismic phase identification. Finally, the pre-trained model is further applied to several open-source seismic datasets to evaluate the model performance in phase arrival-time picking performance. To that aim, in this study, the STEAD dataset [22], the Italian seismic dataset for machine learning (INSTANCE) [24], and the dataset across the Iquique region of northern Chile (Iquique) [25] are used to verify the model performance in seismic phase picking.

The main contributions of this work are summarized as follows: (a) The proposed method for seismic phase identification is capable of reliably detecting P- and S-waves of even very small earthquakes, e.g., the local magnitude of the SCEDC dataset ranges from \(-0.81\) to \(5.7\ M_L\). (b) The results tested on the data of varying lengths indicate that DynaPicker can be adaptive to different lengths of input data for seismic phase identification. Meanwhile, it is proved that Dynapicker is robust to classify seismic phases even when the seismic data is polluted by noise. (c) The testing data and the training data used for seismic phase identification and phase picking have no overlap, which proves that DynaPicker is capable of generalizing entire waveforms and metadata archives from different regions. (d) The proposed model can achieve a superior picking performance over the baseline methods.
Fig. 2. Visualization of arrival-time picking using DynaPicker for a given normalized seismic waveform. Here, the subfigure c shows only one channel of a real seismogram from the STEAD dataset [22]. The figure presents the model performance for different input window lengths of 2s, 4s, and 6s; the windows are shifted by 10 samples at a time (for further details on this refer to the methodology section). The subsequent windows are denoted by different colors and shown explicitly in subfigures b and d. Note that we only show specific windows around P- and S-arrivals in subfigures b and d, respectively, as they are most relevant for the corresponding picks. a and e show the predicted probability of P-phase and S-phase arrivals, respectively for the entire waveform. Each window visualized in subfigure b, is mapped to a vertical line of the corresponding color in subfigure a at the window index $w_i$ representing that window. Similarly, each window visualized in subfigure d, is mapped to a vertical line of the corresponding color in subfigure e at the window index representing that window. The blue and pink dashed vertical lines in subfigure c represent the true P-phase and S-phase arrival times (provided in the metadata for the dataset), respectively; analogously, the solid dashed and dotted blue vertical lines in subfigure a indicate the window indices corresponding to the predicted P-arrival for models trained on 4s, 2s and 6s windows respectively and the solid, dashed and dotted pink vertical lines in subfigure e indicate the window indices corresponding to the predicted S-arrival for models trained on 4s, 2s and 6s windows respectively. The P- and S-arrival samples are considered to be at the center of the picked windows.
II. METHODOLOGY

In this study, we develop a 1D-DCD-based seismic phase classifier to handle seismic time series data. Our model takes a window of the normalized three-channel seismic waveform as input and predicts its label as P-phase, S-phase, or noise. Then, the pre-trained model is employed to automatically pick the arrival time of real-time continuous seismic data. Fig. 1 schematically visualizes the proposed model architecture which consists of convolutional layers, batch normalization, dropout, DCD-based layers, and a 1D dynamic classifier adapted from the work [3].

A. Dynamic convolution decomposition (DCD)

Dynamic convolution achieves a significant performance improvement over convolutional neural networks (CNNs) by adaptively aggregating K static convolution kernels [19], [26]. As shown in the paper written by Li et al [3], based on an input-dependent attention mechanism, dynamic convolution succeeds in aggregating multiple convolution kernels into a convolution weight matrix, which can be described as equation (1) and (2).

\[ W(x) = \sum_{k=1}^{K} \pi_k(x) W_k \]  
\[ s.t. \ 0 \leq \pi_k(x) \leq 1, \sum_{k=1}^{K} \pi_k(x) = 1 \]

where the attention scores \( \{ \pi_k(x) \} \) are used to linearly aggregate the \( K \) convolution kernels \( \{ W_k(x) \} \).

However, the vanilla dynamic convolution suffers from two main limitations: firstly, the use of \( K \) kernels will lead to the lack of compactness; secondly, it is challenging to jointly optimize the attention scores \( \{ \pi_k(x) \} \) and static kernels \( \{ W_k \} \) [3].

To address the above-mentioned issues, Li et al. [3] revisited dynamic convolution from a matrix decomposition viewpoint. They further proposed dynamic channel fusion to replace dynamic attention over channel group to reduce the dimension of the latent space, and mitigate the difficulty of the joint optimization problem. Fig. 1 gives an illustration of a DCD layer. The general formulation of dynamic convolution using dynamic channel fusion is given [3]:

\[ W(x) = \Lambda(x)W_0 + P\Phi(x)Q^T \]

where \( \Lambda(x) \) represents a \( C \times C \) diagonal matrix (\( C \) denotes the number of channels), and \( W_0 \) denotes the static kernel. In the matrix \( \Lambda(x) \), the element \( \lambda_{i,i}(x) \) is a function of the input \( x \). The matrix \( \Phi(x) \) of size \( L \times L \) fuses channels in the latent space \( \mathbb{R}^L \) associated with the dimensionality \( L \) dynamically. The two static matrices \( Q \in \mathbb{R}^{C \times L} \) and \( P \in \mathbb{R}^{L \times L} \) are used to compress the input \( x \) into a low dimensional space and expand the channel number to the output space, respectively. More details can be found in the paper by Li et al. [3].

B. Seismic Phase Classifier Network architecture

As presented in Fig. 1, the first convolutional layer is applied to process a three-channel window of seismic data in the time domain, to generate a feature representation. Then, a batch normalization layer (BN) is used to accelerate the training process and provide the stability for the network followed by an activation function using Rectified Linear Unit (ReLU) [27]. Finally, a max-pooling block [28] is added to reduce the size of the feature map, which is followed by a Dropout layer [29] to avoid overfitting. The second part of the framework is comprised of several DCD-based layers, which are used to leverage favorable properties that are absent in static models. The right part of Fig. 1 shows the diagram of the 1D-DCD block, where a dynamic branch is used to produce coefficients for dynamic channel-wise attention \( \Lambda(x) \) of size \( C \times C \) and dynamic channel fusion \( \Phi(x) \) of size \( L \times L \) [3]. In the dynamic branch, the average pooling is first applied to the input \( x \) and then is followed by two fully connected (FC) layers associated with an activation layer between them. For the two used FC layers, the former aims to reduce the number of channels, and the latter tries to expand them into \( C + L^2 \) outputs. Similar to a static convolution, a DCD layer also includes a batch normalization and an activation (e.g. ReLU) layer followed by a dropout layer. Finally, the dynamic classifier uses this information to map the high-level features to a discrete probability over 3 categories (P-wave, S-wave, and noise wave). The dynamic classifier is also based on a 1D-DCD block.

It is worth noting that the model introduced in this study can be easily adapted to address inputs with different window sizes by simultaneously adjusting the sizes of the first layer and the dynamic classifier layer, respectively. With the goal to verify the model robustness, the impact of different length data on seismic phase identification is investigated in the following section. The pre-trained model is extensively applied to pick arrival time for real-time seismic data. The process of the arrival time picking for real-time seismic data using different window sizes when feeding the same continuous seismic waveform is schematically visualized in Fig. 2.

C. Phase arrival-time estimation for continuous seismic data

To achieve seismic phase picking, the following steps are included, where the main steps are the same as in GPD [13] and CapsPhase [30]. The pipeline for phase arrival time picking on continuous seismic data using the pre-trained phase classifier is visualized in Fig. 3.

- First, each waveform is filtered using the bandpass filter. For instance, the data from the STEAD dataset is filtered within the frequency range 2-20Hz, following the CapsPhase [30].
- Then, the waveform is resampled at 100Hz followed by normalization using the absolute maximum amplitude. For example, for the STEAD dataset, each waveform has a size of \( 6000 \times 3 \) after pre-processing.
- Afterwards, the data of filtered are divided into several windows. Each window contains a 4s-three-component seismogram (400 samples since the sampling rate is 100
Hz), while the window strides with ten samples such that the number of overlapping samples between neighbor windows is 390 samples. Therefore, the total number of windows is as follows.

\[
N_{\text{win}} = \frac{L_{\text{total}} - L_{\text{win}}}{n_{\text{shift}}} + 1 \quad (4)
\]

where \(L_{\text{total}}\) and \(L_{\text{win}}\) denote the length of the original waveform after sampling, and the length of the window (e.g., 400 samples in this study), respectively. \(n_{\text{shift}}\) is the number of the shift between windows in samples and in this work, it is empirically set as 10 same as CapsPhase [30].

- Then, the pre-trained classifier is utilized to predict three sequences of probabilities for each window associated with P-phase, S-phase, and noise respectively. Following the work [19], a temperature softmax function [31] is used in this study to smooth the output probability as follows:

\[
\delta_k = \frac{\exp(z_k/T)}{\sum_j \exp(z_j/T)} \quad (5)
\]

where \(z_k\) is the output of the classifier layer, and \(T\) is the temperature. The original softmax function is a special case when \(T = 1\). As \(T\) increases, the output is less sparse. In this study, the value of \(T\) is experimentally set to 4.

- Finally, the arrival-time detection is declared using the following equation:

\[
t_{(P/S)} = 0.01 \times (\text{Win}_{\text{index}} \times n_{\text{shift}} + n_c) + t_{\text{star}} \quad (6)
\]

where \(\text{Win}_{\text{index}}\) denotes the window index of the largest probability, and \(t_{\text{star}}\) is the trace starting time. \(n_c\) denotes the added constant that is 0.5*window length for generalization since in the SCEDC dataset [23], the P-wave and S-wave windows are centered around the arrival time.

III. DATA AND LABELING

Within this work, the dataset provided by Southern California Earthquake Data Center (SCEDC) [23] is used for model training and testing in seismic phase identification. The magnitude range of the data is \(-0.81 < M_L < 5.7\). This dataset is comprised of 4.5 million three-component seismic signals with a duration of 4s including 1.5 million P-phase picks, 1.5 million S-phase picks, and 1.5 million noise windows. The P-wave and S-wave windows are centered on the arrival pick, while each noise window is captured by starting 5s before each P-wave arrival. Finally, the absolute maximum amplitude discovered on the three components is used to normalize each three-component seismic record. In this study, 90% of the seismograms from the SCEDC dataset [23] are used for model training, and 5% of seismograms are employed to test the model performance. Furthermore, we compare the seismic phase classification performance to a capsule neural network-based seismic data classification approach, termed CapsPhase [30], and our previous work, 1D-ResNet [9].

To achieve seismic phase identification, DynaPicker takes a window of three-channel waveform seismogram data as input, and then for each input, the model predicts the probabilities corresponding to each class (P-wave, S-wave, or noise). This model has three output labels: zero for the P-wave window, one for the S-wave window, and two for the noise window.

In order to further evaluate the model performance in phase arrival-time picking, several subsets of three open-source public seismic datasets namely the STEAD dataset [22], the INSTANCE dataset [24], and the Iquique dataset [25] are used. Each waveform in the first two datasets is either 1 or 2 minutes long. They can be viewed as good generalization tests of our proposed method. DynaPicker is compared to the generalized phase detection (GPD) framework [13] based
on convolutional neural networks, CapsPhase [30] based on capsule neural network [32], and AR picker [33] to evaluate the performance of phase arrival-time picking for real-time seismic data.

IV. EVALUATION METRICS FOR SEISMIC PHASE CLASSIFICATION

In this article, noise labels are not treated differently from phase labels, so classifying a noise window correctly has the same weight as confirming a phase window. The seismic phase detector can be viewed as a three-class classifier that decides whether a given time window contains a seismic phase (P or S), or only noise. Here, the "noise" windows do not contain P- or S-phases. We can evaluate a deep-learning model by processing labeled testing data where the true output is known. The accuracy defined below is a simple measure of a classifier’s performance:

\[ \text{Accuracy} = \frac{N_C}{N_T} \]  

where \( N_C \) denotes the number of correctly labeled samples and \( N_T \) represents the total number of testing samples.

To evaluate the detector's effectiveness, a confusion matrix [34] is adopted to reflect the classification result, and then precision and recall can be defined as follows:

\[ \text{Precision} = \frac{TP}{TP + FP} \]  
\[ \text{Recall} = \frac{TP}{TP + FN} \]

The F1-score is computed from the harmonic mean of precision and recall for each class:

\[ F1_{\text{score}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \]

where TN, FN, FP, and TP are the number of true negative, false negative, false positive, and true positive, respectively.

V. EXPERIMENTS AND RESULTS

A. Seismic phase classifier training

In this study, for dynamic convolution decomposition units, all the weight and filter matrices are initialized with a normal initializer and bias vectors set to zeros. For optimization, we use the ADAM [35] algorithm, which keeps track of first- and second-order moments of the gradients and was invariant to any diagonal rescaling of the gradients. We used a learning rate of \( 10^{-3} \) and trained the DynaPicker for 50 epochs same as CapsPhase [30]. In this work, DynaPicker was implemented in Pytorch [36] and all the training was performed on a NVIDIA A100 GPU. The model was trained using a cross-entropy loss function with the ADAM optimization algorithm, in mini-batches of 480 records. We used a dropout rate of 0.2 for all dropout layers.

B. Investigation on different length input data

Here, we investigate the impact of different input data lengths on the performance of seismic phase detection and arrival-time picking using the STEAD dataset. The details of arrival-time picking using a pre-trained phase classifier can be found in the following subsections and the Methods section.

1) Different length of the input data on phase classification: To that end, we select 58,018 earthquake waveforms from the STEAD dataset [22] and create three datasets within different durations (2s, 4s, and 6s). There is a total of 174,054 waveforms including P-wave, S-wave, and noise wave in each dataset. In this experiment, all data are re-sampled at 100 Hz and each three-component waveform is normalized by the absolute maximum amplitude observed on any of the three components. Similar to the SCEDC dataset [23], P-wave and S-wave windows are centered on the respective arrival-time picks. Meanwhile, noise windows are captured from pure noise waveforms. Note that these three datasets are comprised by the same events, and only the window length is different.

Then, each dataset is split into a training dataset (90%) and a testing dataset (10%). The overall testing accuracy for different length input data is estimated to be 95.52%, 97.99%, and 98.02% in line with 2s, 4s, and 6s respectively. The result demonstrates that DynaPicker can work well with the input of different time duration.

The confusion matrices corresponding to the input data with different duration are shown in Fig. 4. We can observe that the developed model reaches a high detection accuracy for each class, especially in noise window detection as shown in Fig. 4, where noise waveform is more easily distinguishable from P and S arrivals than they are from each other in the cases for 4s and 6s data.

In the end, the testing results indicate that our model can be adaptive to different lengths of input data. At the same time, our model achieves a compatible performance in seismic phase picking even with low-volume training data. Meanwhile, the model is compared to EPick [17], a simple neural network that incorporates an attention mechanism into a U-shaped neural network. Here, the pre-trained and saved model of EPick is directly used without retraining. Besides, there is no overlap between the training data used for seismic phase identification and the data adopted in testing the model performance in phase picking. The details for arrival time picking can be found in the Methods section. The testing results are summarized in Table I. Firstly, we can observe that EPick achieves the best performance in phase picking over DynaPicker by using different window sizes. The potential reason is that EPick is pre-trained on the data labeled with the specific phase arrival time from the STEAD dataset. Secondly, a larger window size reduces the amount of the P-phase with an error lower than 0.5s. Thirdly, in the case where the window size is 4s, the number of the S-phase with an error lower than 0.5s is larger than in the other two cases e.g., 2s and 6s.
(a)
(b)
(c)

Fig. 4. Confusion matrices for seismic phase classification given different length input data: (a) 2s, (b) 4s, and (c) 6s.

| Method              | No. of undetected events | No. of abs(e) ≤ 0.5s for P-pick | μ_P   | σ_P   | No. of abs(e) ≤ 0.5s for S-pick | μ_S   | σ_S   | Ref.                  |
|---------------------|--------------------------|---------------------------------|-------|-------|---------------------------------|-------|-------|-----------------------|
| DynaPicker (2s)     | 0                        | 8236                            | 0.027 | 0.146 | 3014                            | -0.048| 0.200 | This study            |
| DynaPicker (4s)     | 0                        | 3734                            | 0.011 | 0.136 | 3855                            | -0.120| 0.182 | This study            |
| DynaPicker (6s)     | 0                        | 2819                            | 0.058 | 0.218 | 1733                            | -0.127| 0.224 | This study            |
| EPick               | 0                        | 9873                            | -0.002| 0.052 | 9663                            | 0.002 | 0.122 | [17]                  |

μ_P and σ_P are the mean and standard deviation of errors (ground truth − prediction) in seconds respectively for P phase picking. μ_S and σ_S are the mean and standard deviation of errors (ground truth − prediction) in seconds respectively for S phase picking.

C. Seismic phase classification on 4s SCEDC Dataset

As discussed in the previous subsections, the proposed model, DynaPicker, can be adapted to the data with different lengths and achieves compatible performance.

Here, DynaPicker is further retrained and tested on the SCEDC dataset [23] collected by the Southern California Seismic Network (SCSN). Then, we compared our model with CapsPhase [30] and our previous work, 1D-ResNet [9] with the same test set. The testing accuracy of DynaPicker is 98.82%, which is slightly greater than CapsPhase [30] (98.66%) and 1D-ResNet [9] (98.66%).

Then, different evaluation metrics, like the Precision, Recall, and F1-score for DynaPicker, CapsPhase [30], and 1D-ResNet [9] are summarized in Table II. As one can see from Table II, compared with the baseline methods, DynaPicker can achieve superior performance in terms of the F1-score. For precision and recall, DynaPicker also achieves a comparable performance.

Finally, in order to investigate the model performance, when facing more noisy data, the same subset selected from the STEAD dataset used in 1D-ResNet [9] is utilized. Here, the signal-to-noise ratio (SNR) of the selected data before adding noise ranges from 0 to 70 dB, and the SNR is the mean value of SNR over three components for each signal. The magnitude of the data ranges from 1.0 to 3.0. To study the impact of different noise levels on model performance, the subset is masked by the Gaussian noise (similar to the method used in EQTransformer [15]) with mean μ = 0 and standard deviation δ = 0.01, 0.05, 0.1, and 0.15, respectively. Afterward, these noisy data are fed to the pre-trained phase classifier to test the model performance. The testing accuracies of different models are summarized in Table III below. The results in Table III show that (a) large noise reduces the model performance; (b) DynaPicker outperforms over CapsPhase and 1D-ResNet; (c) DynaPicker is robust in identifying seismic phases when the seismic data is polluted by noise.

D. Seismic arrival-time picking for continuous seismic records

We next demonstrate the applicability of our model to pick the seismic phase arrival time for continuous seismic data in the time domain. The main parameters related to phase arrival-time picking are studied in Appendix. Within this work, DynaPicker is implemented for seismic phase identification given short-window seismic waveforms same as GPD and CapsPhase. Hence, DynaPicker is firstly compared with GPD and CapsPhase on both the STEAD dataset and the INSTANCE dataset. Secondly, we compare DynaPicker with one of the state-of-the-art seismic phase pickers, EQTransformer [15] on the Iquique dataset [25]. The reason is that on one hand, EQTransformer is a multi-task deep learning model designed for earthquake detection and seismic phase picking, which is trained on the STEAD dataset labeled with specific phase arrival time. On the other hand, the original INSTANCE paper [24] reported that EQTransformer is used in picking the first arrivals of P- and S-waves. Therefore, in this study, the subset of the Iquique dataset [25] is further applied to achieve a fair comparison between DynaPicker and EQTransformer.

1) Application to the STEAD dataset: We randomly select 20,000 earthquake waveforms from the STEAD dataset out of
TABLE II
RESULTS OF EVALUATION METRICS ON THE TEST DATASET [23] FOR PHASE CLASSIFICATION.

| Category | Model       | Precision | Recall | F1-score | Ref.          |
|----------|-------------|-----------|--------|----------|---------------|
| P-phase  | DynaPicker  | 99.15%    | 98.54% | 98.84%   | This study    |
|          | CapsPhase   | 98.93%    | 98.45% | 98.69%   | [30]          |
|          | 1D-ResNet   | 98.64%    | 98.64% | 98.76%   | [9]           |
| S-phase  | DynaPicker  | 98.87%    | 99.04% | 99.96%   | This study    |
|          | CapsPhase   | 98.89%    | 98.63% | 98.76%   | [30]          |
|          | 1D-ResNet   | 98.72%    | 98.94% | 98.83%   | [9]           |
| Noise    | DynaPicker  | 98.43%    | 98.86% | 98.65%   | This study    |
|          | CapsPhase   | 98.17%    | 98.90% | 98.54%   | [30]          |
|          | 1D-ResNet   | 98.52%    | 98.54% | 98.53%   | [9]           |

The saved model of CapsPhase is directly used here without retraining and, unlike the original CapsPhase [30], the output threshold for each class is not used in this work since it reduces the CapsPhase performance in the testing phase. Bold values represent the best performance.

TABLE III
TESTING RESULTS OF DIFFERENT NOISE LEVELS FOR PHASE IDENTIFICATION ON THE STEAD DATASET.

| Noise level | 0.01 | 0.05 | 0.1  | 0.15 |
|-------------|------|------|------|------|
| Capsphase [30] | 95.28% | 95.43% | 92.80% | 88.90% |
| 1D-ResNet [9]    | 96.30% | 96.46% | 93.22% | 89.16% |
| DynaPicker       | 96.88% | 96.73% | 94.49% | 91.26% |

The best-saved model of CapsPhase is directly used here without retraining and unlike the original CapsPhase paper [30], the output threshold for each class is not used in this work since it reduces the CapsPhase performance in the testing phase. Bold values represent the best performance.

which 10,000 have a time difference greater than 4s between P-wave arrival and S-wave arrival, while for the remainder this difference is lower than 4s and the epicentral distances are less than or equal to 35km. Here we use 4s as the threshold for waveform selection since the SCEDC dataset [23] with the duration of 4s is used to train and test the model performance on seismic phase classification. To study the impact of the time difference between P and S picks, the events of different time differences are used to verify our model’s robustness in seismic arrival time picking for continuous seismic data.

As presented in GPD [13] and CapsPhase [30], a triggering method is used to locate arrival picks by setting a threshold. However, the picking performance is impacted by the threshold. To overcome this drawback we use the window index with the largest probability to locate the P and S picks as this empirically yields the best results.

Table IV summarizes the testing result of arrival-time picking. From Table IV, we can see that (a) our model succeeds in correctly detecting all seismic events, while about 174 and 115 seismic events cannot be detected by GPD [13] and CapsPhase [30] for the earthquake signal with \( S_{arrival} - P_{arrival} > 4s \), and about 338 and 139 seismic events cannot be detected by GPD [13] and CapsPhase [30] for the earthquake signal with \( S_{arrival} - P_{arrival} < 4s \); (b) compared with GPD [13], CapsPhase [30], and AR picker [33], most of the error between the located P-wave or S-wave picks and the ground truth are within 0.5 s when using DynaPicker. We use 0.5 s for our analysis following Capsphase [30]; (c) DynaPicker is robust for seismic events of different time differences between P and S picks. In summary, our proposed model outperforms the baseline methods.

2) Application to the INSTANCE dataset: We also evaluate the picking performance of our model using the INSTANCE dataset [24] and compare the picking error with the benchmark methods. This dataset includes about 1.2 million three-component waveforms from about 50,000 earthquake events recorded by the Italian National Seismic Network. In the metadata, the recorded earthquake list ranges from January 2005 to January 2020, and the magnitude of the earthquake events ranges from 0.0 to 6.5. All the recorded seismic waveforms have a duration of 120s and are sampled at 100Hz. We randomly select 20,000 earthquake waveforms from the INSTANCE dataset [24], out of which 10,000 have a time difference greater than 4s between P-wave arrival and S-wave arrival, while for the remainder this difference is lower than 4s and similarly, the epicentral distances are less than or equal to 35km.

As summarized in Table V, we can observe that the proposed model outperforms the baseline methods. On one hand, the proposed model succeeds in identifying the true label corresponding to each input, which means all seismic events are detected compared with the used baseline methods. On the other hand, our model achieves a lower arrival-time picking error, and it is robust for different time differences between P and S picks. In particular, our model achieves the lowest mean error in S-phase arrival-time picking for both cases.

3) Application to the Iquique dataset: The Iquique dataset is comprised of locally recorded seismic arrivals throughout northern Chile, and is used in several previous studies [25], [37], [38] to train a deep learning-based picker. It contains about 11,000 manually picked P-/S- phase pairs, where all the seismic waveform units are recorded in counts. In this study, 10,000 P-/S-phase pairs are used, and DynaPicker is further compared with the advanced deep learning model Earthquake transformer (EQTransformer) [15], to evaluate onset picking. Afterward, the same data processing procedures are also conducted to achieve the 4s window data. In particular, it is worth noting that neither DynaPicker nor EQTransformer are retrained on the Iquique dataset.

First, the confusion matrices for P- and S- arrival picking results of the experiment using DynaPicker and EQTransformer are shown in Fig. 5a and 5b. We find that out of the selected 10,000 signals, EQTransformer misses 243 events, which means that for these misclassified earthquake events, no arrival
Detects multiple picks including one incorrectly detected P-phase in the bottom subplot of Fig. 5c. In Fig. 5d, EQTransformer is of high probability, and S-phase is also detected as shown in the bottom subplot of Fig. 5d. However, the estimated P-phase by DynaPicker is with a low probability, and the S-phase is missing, while DynaPicker also picks multiple P-phase and S-phases. It can be observed that in Fig. 5c the detected P-phase and S-phase arrival time pairs.

Then, two examples from the Iquique dataset using EQTransformer and DynaPicker are displayed in Fig. 5c and 5d, respectively. The picking result of EQTransformer is shown in Fig. 5c and DynaPicker, only the sample with the largest probability is recognized as P- or S-phases. It can be observed that in Fig. 5c the detected P- and S-phases are the mean and standard deviation of errors (ground truth – prediction) in seconds respectively for P phase picking. For both P and S waves, EQTransformer performs slightly better than DynaPicker in terms of both the root mean squared error (RMSE) and the mean absolute error (MAE).

Finally, the absolute error between deep learning-based model predicted picks (e.g., EQTransformer and DynaPicker) and manual picks that are below 0.5s are taken into account. For both P and S waves, EQTransformer performs slightly better than DynaPicker in terms of both the root mean squared error (RMSE) and the mean absolute error (MAE). Here, the MAE and RMSE of both P and S waves using EQ-
Fig. 5. Visualizations of the testing result on the Iquique dataset. (a) and (b) are the confusion matrices from in-domain experiments for DynaPicker and EQTransformer, respectively. Here, the pre-trained model of EQTransformer is directly used without retraining and adopted from Seisbench [37], where DynaPicker is able to detect all earthquake events compared with EQTransformer. (c) and (d) plots the EQTransformer and DynaPicker predictions on two waveform examples from the Iquique dataset. In (c) and (d), the upper three subplots are the three-component seismic waveforms where the vertical red and blue lines correspond to the ground truth arrival time of P- and S-phases from the dataset metadata, respectively, and the bottom subplots display the predicted probability for P- and S-phases by using EQTransformer and DynaPicker, respectively, where the dashed vertical lines in red and blue depict the locations of the maximal predicted probabilities of P- and S-phases, respectively. For EQTransformer, in (c) only the P-pick is detected at a low probability, whereas the S-pick is missing, and in (d) multiple picks are predicted, especially one incorrectly P-phase is detected at a high probability. For DynaPicker, both the true P- and S-phases are detected with a higher probability compared with EQTransformer.

Transformer are $MAE(P) = 0.091s$, $RMSE(P) = 0.095s$ and $MAE(S) = 0.159s$, and $RMSE(S) = 0.126s$. And the MAE and RMSE of both P and S waves using DynaPicker are $MAE(P) = 0.127s$, $RMSE(P) = 0.128s$ and $MAE(S) = 0.198s$, and $RMSE(S) = 0.137s$. However, it is worth noting, that the original EQTransformer is trained on labeled arrival-time seismic data of the STEAD dataset, while DynaPicker is only trained on the short-window SCEDC dataset without phase arrival-time labeling.

4) Brief Summary: On one hand, the evaluations performed on the STEAD and INSTANCE datasets strongly verify that DynaPicker achieves a promising arrival-time picking performance compared with GPD [13], CapsPhase [30], and AR picker [33], and they also indicate that our model has good generalizability. On the other hand, DynaPicker achieves a comparable phase picking result on the Iquique dataset compared to EQTransformer [15].

VI. CONCLUSION

In this study, we present a DCD-based network for seismic phase classification. Our network consists of a normal convolution layer and DCD layers. Our model is trained using the seismic data collected by the Southern California Seismic Network, and it shows good results when it is tested using earthquake waveforms recorded worldwide, indicating a good generalization ability. Extensive experiments demonstrate that this model yields superior performance over several baseline methods. The results from our work add to the literature of supervised learning-based methods for seismic phase classification and demonstrate that with appropriate considerations
regarding overfitting and generalization, such methods can improve seismological processing workflows, not just for large catalogs, but also for varying datasets.

**DATA AVAILABILITY**

The seismic dataset of the Southern California Earthquake Data Center used in this study can be accessed at https://scedc.caltech.edu/data/deeplearning.html. The STEAD data can be downloaded from https://github.com/smousavi05/STEAD, and the INSTANCE dataset is freely available at http://www.pi.ingv.it/instance/. The details about how to download and use the Iquique dataset can be found in SeisBench [37].

**ACKNOWLEDGMENTS**

This work is supported by the “K Ignawuchswissenschaftlerinnen” - grant SAI 01IS20059 by the Bundesministerium für Bildung und Forschung - BMBF. Calculations were performed at the Frankfurt Institute for Advanced Studies’ GPU cluster, funded by BMBF for the project Seismologie und Artifizielle Intelligenz (SAI). We thank the authors of Seisbench [37] for their help to use the saved model of EQTransformer [15] in the Pytorch version, and also thank Dr. Omar M. Saad for his help in using the pre-trained model of CapsPhase [30].

**REFERENCES**

[1] R. V. Allen, “Automatic earthquake recognition and timing from single traces,” Bulletin of the seismological society of America, vol. 68, no. 5, pp. 1521–1532, 1978.
[2] M. Leonard and B. Kennett, “Multi-component autoregressive techniques for the analysis of seismograms,” Physics of the Earth and Planetary Interiors, vol. 113, no. 1–4, pp. 247–263, 1999.
[3] Y. Li, Y. Chen, X. Dai, M. Liu, D. Chen, Y. Yu, L. Yuan, Z. Liu, M. Chen, and N. Vasconcelos, “Revisiting dynamic convolution via matrix decomposition,” arXiv preprint arXiv:2103.08756, 2021.
[4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436–444, 2015.
[5] M. Chakraborty, C. Q. Cartaya, W. Li, J. Faber, G. Rüpcker, H. Stöcker, and N. Srivastava, “Polarcap—a deep learning approach for first motion polarity classification of earthquake waveforms,” Artificial Intelligence in Geosciences, vol. 3, pp. 46–52, 2022.
[6] T. Perol, M. Gharbi, and M. Denolle, “Convolutional neural network for earthquake detection and location,” Science Advances, vol. 4, no. 2, p. e1700578, 2018.
[7] S. M. Mousavi, W. Zhu, Y. Sheng, and G. C. Beroza, “Cred: A deep residual network of convolutional and recurrent units for earthquake signal detection,” Scientific reports, vol. 9, no. 1, pp. 1–14, 2019.
[8] D. Fenner, G. Rüpcker, W. Li, M. Chakraborty, J. Faber, J. Kohler, H. Stöcker, and N. Srivastava, “Automated seismo-volcanic event detection applied to stromboli (italy),” Frontiers in Earth Science, p. 267, 2022.
[9] W. Li, Y. Sha, K. Zhou, J. Faber, G. Ruempker, H. Stoecker, and N. Srivastava, “A study on small magnitude seismic phase identification using 1d deep residual neural network,” Artificial Intelligence in Geosciences, vol. 3, pp. 115–122, 2022.
[10] M. Chakraborty, G. Rüpcker, H. Stöcker, W. Li, J. Faber, D. Fenner, K. Zhou, and N. Srivastava, “Real time magnitude classification of earthquake waveforms using deep learning,” in EGU General Assembly Conference Abstracts, pp. EGU21–15941, 2021.
[11] M. Chakraborty, D. Fenner, W. Li, J. Faber, K. Zhou, G. Ruempker, H. Stoecker, and N. Srivastava, “Creime: A convolutional recurrent model for earthquake identification and magnitude estimation,” Journal of Geophysical Research: Solid Earth, vol. 127, no. 7, p. e2022JB024595, 2022.
[12] M. Chakraborty, J. Faber, L. Ellisworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, “Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking,” Nature communications, vol. 11, no. 1, pp. 1–12, 2020.
[13] Z. E. Ross, M.-A. Meier, E. Hauksson, and T. H. Heaton, “Generalized seismic phase detection with deep learning,” Bulletin of the Seismological Society of America, vol. 108, no. 5A, pp. 2894–2901, 2018.
[14] W. Zhu and G. C. Beroza, “Phasenet: a deep-neural-network-based seismic arrival-time picking method,” Geophysical Journal International, vol. 216, no. 1, pp. 261–273, 2019.
[15] S. M. Mousavi, W. L. Ellisworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, “Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking,” Nature communications, vol. 11, no. 1, pp. 1–12, 2020.
[16] W. Li, G. Rüpcker, H. Stöcker, M. Chakraborty, D. Fenner, J. Faber, K. Zhou, J. Steinheimer, and N. Srivastava, “Mca-unet: Multi-class attention-aware u-net for seismic phase picking,” in EGU General Assembly Conference Abstracts, pp. EGU21–15841, 2021.
[17] W. Li, M. Chakraborty, D. Fenner, J. Faber, K. Zhou, G. Ruempker, H. Stoecker, and N. Srivastava, “Epice: Attention-based multi-scale unet for earthquake detection and seismic phase picking,” Frontiers Earth Science, p. 2075, 2022.
[18] A. Stepnov, V. Chernykh, and A. Konovalov, “The seismo-performer: A novel machine learning approach for general and efficient seismic phase recognition from local earthquakes in real time,” Sensors, vol. 21, no. 18, p. 6290, 2021.
[19] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, “Dynamic convolution: Attention over convolution kernels,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11103–11109, 2020.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.
[21] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, “Dynamic neural networks: A survey,” arXiv preprint arXiv:2102.04906, 2021.
[22] S. M. Mousavi, Y. Sheng, W. Zhu, and G. C. Beroza, “Stanford earthquake dataset (stead): A global data set of seismic signals for ai,” IEEE Access, vol. 7, pp. 179464–179476, 2019.
[23] S. C. E. D. Center, “Southern california earthquake data center (2013),” California Institute of Technology, Dataset, 2013. doi:10.7909/C3W3hx11.
[24] A. Micheli, S. Cianetti, S. Gaviano, C. Giunchi, V. D. Jozinovic, and L. V. Lancei, “Instance–the italian seismic dataset for machine learning,” Earth System Science Data Discussions, pp. 1–47, 2021.
[25] J. Woollam, A. Rietbrock, A. Bueno, and S. De Angelis, “Convolutional neural network for seismic phase classification, performance demonstration over a local seismic network,” Seismological Research Letters, vol. 90, no. 2A, pp. 491–502, 2019.
[26] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, “Condeoconv: Conditionally parameterized convolutions for efficient inference,” Advances in Neural Information Processing Systems, vol. 32, 2019.
[27] A. F. Agarap, “Deep learning using rectified linear units (relu),” arXiv preprint arXiv:1803.08375, 2018.
[28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in International Conference on Learning Representations, 2015.
[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.
[30] O. M. Saad and Y. Chen, “Capsphase: Capsule neural network for seismic phase classification and picking,” IEEE Transactions on Geoscience and Remote Sensing, 2021.
[31] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning, MIT press, 2016. http://www.deeplearningbook.org.
[32] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” Advances in neural information processing systems, vol. 30, 2017.
[33] T. Akazawa, “A technique for automatic detection of onset time of p-and s-phases in strong motion records,” in Proc. of the 13th World Conf. on Earthquake Engineering, vol. 786, p. 786, Vancouver, Canada, 2004.
[34] S. V. Stehman, “Selecting and interpreting measures of thematic classification accuracy,” Remote sensing of Environment, vol. 62, no. 1, pp. 77–89, 1997.
[35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.
[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., “Pytorch: An imperative style, high-performance deep learning library,” Advances in Neural Information Processing Systems, vol. 32, 2019.
Magnitude
10
0
1
2
3
4
5
6
Frequency
100
101
102
103
0 1 2 3 4 5 6
Magnitude
Source_depth [km]
10
0
1
2
3
4
5
6
Frequency
100
101
102
103
0 25 50 75 100 125 150 175 200
Source_depth [km]
Source_distance [km]
Frequency
10
0
1
2
3
4
5
6
Frequency
10
10
0
1
2
3
4
5
6
Time residual [s]
10
0
1
2
3
4
5
6
Frequency
10
10
0
1
2
3
4
5
6
Frequency
10
10
0
1
2
3
4
5
6
Fig. 6. Distribution of (a) earthquake magnitudes, (b) earthquake source depths, (c) earthquake source distances, and (d) time difference between P-phase and S-phase arrival-time.

[37] J. Woollam, J. Münchmeyer, F. Tilmann, A. Rietbrock, D. Lange, T. Bornstein, T. Diehl, C. Giunchi, F. Haslinger, D. Jozinović, et al., “Seisbench—a toolbox for machine learning in seismology,” Seismological Society of America, vol. 93, no. 3, pp. 1695–1709, 2022.

[38] J. Münchmeyer, J. Woollam, A. Rietbrock, F. Tilmann, D. Lange, T. Bornstein, T. Diehl, C. Giunchi, F. Haslinger, D. Jozinović, et al., “Which picker fits my data? a quantitative evaluation of deep learning based seismic pickers,” Journal of Geophysical Research: Solid Earth, p. e2021JB023499, 2022.

APPENDIX

In this part, the impact of different temperatures in the softmax function (see equation (5) for illustration), and different shift numbers (n_{shift}) on the model performance of phase arrival-time picking for continuous seismic waves are investigated. Here, 10,000 samples are randomly selected from the STEAD dataset [22]. The distribution of earthquake magnitudes, earthquake source depth, earthquake source distance, and time difference between P-phase and S-phase arrival-time are displayed in Fig. 6.

Impact of different temperatures. In this part, the impact of different temperatures in the used softmax function on the model performance of phase arrival-time picking for the continuous seismic wave are investigated as summarised in Table VI, in which n_{shift} is fixed as 10. From Table VI, we can conclude that the temperature T is empirically set to 4.

Impact of shift numbers. This part studies the effect of different shift numbers (n_{shift}) on seismic onset arrival time estimation, where the temperature T is set to 4. From Table VII, we can conclude that the results of n_{shift} = 5 and n_{shift} = 10 are close, while according to equation (4), lower shift number increases the numbers of the sliding window,
TABLE VI

| $T$  | No. of undetected events | No. of abs(e) ≤ 0.5s for P-pick | $\mu_P$ | $\sigma_P$ | No. of abs(e) ≤ 0.5s for S-pick | $\mu_S$ | $\sigma_S$ |
|------|--------------------------|---------------------------------|---------|-----------|---------------------------------|---------|-----------|
| 1    | 0                        | 8926                            | 0.018   | 0.132     | 7168                            | -0.003  | 0.199     |
| 4    | 0                        | 9032                            | 0.005   | 0.125     | 6984                            | 0.002   | 0.196     |
| 10   | 0                        | 9063                            | 0.0008  | 0.123     | 6857                            | 0.004   | 0.196     |
| 20   | 0                        | 9084                            | -0.001  | 0.122     | 6797                            | 0.004   | 0.196     |

$\mu_P$ and $\sigma_P$ are the mean and standard deviation of errors (ground truth — prediction) in seconds respectively for P phase picking. $\mu_S$ and $\sigma_S$ are the mean and standard deviation of errors (ground truth — prediction) in seconds respectively for S phase picking.

TABLE VII

| $n_{shift}$ | No. of undetected events | No. of abs(e) ≤ 0.5s for P-pick | $\mu_P$ | $\sigma_P$ | No. of abs(e) ≤ 0.5s for S-pick | $\mu_S$ | $\sigma_S$ |
|-------------|--------------------------|---------------------------------|---------|-----------|---------------------------------|---------|-----------|
| 5           | 0                        | 9050                            | -0.005  | 0.120     | 7032                            | -0.004  | 0.195     |
| 10          | 0                        | 9032                            | 0.005   | 0.125     | 6984                            | 0.002   | 0.196     |
| 20          | 0                        | 8947                            | 0.016   | 0.119     | 6975                            | 0.003   | 0.201     |
| 100         | 0                        | 8652                            | 0.012   | 0.111     | 5803                            | 0.024   | 0.248     |

$\mu_P$ and $\sigma_P$ are the mean and standard deviation of errors (ground truth — prediction) in seconds respectively for P phase picking. $\mu_S$ and $\sigma_S$ are the mean and standard deviation of errors (ground truth — prediction) in seconds respectively for S phase picking.

and more time is used to locate the arrival-time. Hence, in this study, the shift number $n_{shift}$ is set as 10.