JGR Solid Earth

RESEARCH ARTICLE

10.1029/2021JB023499

Special Section:
Machine learning for Solid Earth observation, modeling and understanding

Which Picker Fits My Data? A Quantitative Evaluation of Deep Learning Based Seismic Pickers

Jannes Münchmeyer[1,2], Jack Woollam[3], Andreas Rietbrock[3], Frederik Tilmann[1,4], Dietrich Lange[5], Thomas Bornstein[1], Tobias Diehl[6], Carlo Giunchi[7], Florian Haslinger[6], Dario Jozinović[8,9], Alberto Michelini[8], Joachim Saul[1], and Hugo Soto[1]

[1]Deutsches GeoForschungsZentrum GFZ, Potsdam, Germany, [2]Institut für Informatik, Humboldt-Universität zu Berlin, Berlin, Germany, [3]Geophysical Institute (GPI), Karlsruhe Institute of Technology, Karlsruhe, Germany, [4]Institut für geologische Wissenschaften, Freie Universität Berlin, Berlin, Germany, [5]GEOMAR Helmholtz Centre for Ocean Research Kiel, Kiel, Germany, [6]Swiss Seismological Service, ETH Zurich, Zurich, Switzerland, [7]Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Pisa, Pisa, Italy, [8]Istituto Nazionale di Geofisica e Vulcanologia, Roma, Italy, [9]Università degli Studi Roma Tre, Rome, Italy

Jannes Münchmeyer and Jack Woollam contributed equally.

Key Points:

- We conducted a large scale benchmark of machine learning pickers using six models and eight datasets
- Best overall performance is observed for EQTransformer, GPD and PhaseNet, with advantages for EQTransformer on teleseismic distances
- Models transfer well between different regions with similar distances, but not between regional and teleseismic distances

Supporting Information:

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

Correspondence to:

J. Münchmeyer,
munchmej@gfz-potsdam.de

Citation:

Münchmeyer, J., Woollam, J., Rietbrock, A., Tilmann, F., Lange, D., Bornstein, T., et al. (2022). Which picker fits my data? A quantitative evaluation of deep learning based seismic pickers. *Journal of Geophysical Research: Solid Earth*, *127*, e2021JB023499. https://doi.org/10.1029/2021JB023499

Received 26 OCT 2021
Accepted 22 DEC 2021

Abstract

Seismic event detection and phase picking are the base of many seismological workflows. In recent years, several publications demonstrated that deep learning approaches significantly outperform classical approaches, achieving human-like performance under certain circumstances. However, as studies differ in the datasets and evaluation tasks, it is unclear how the different approaches compare to each other. Furthermore, there are no systematic studies about model performance in cross-domain scenarios, that is, when applied to data with different characteristics. Here, we address these questions by conducting a large-scale benchmark. We compare six previously published deep learning models on eight data sets covering local to teleseismic distances and on three tasks: event detection, phase identification and onset time picking. Furthermore, we compare the results to a classical Baer-Kradolfer picker. Overall, we observe the best performance for EQTransformer, GPD and PhaseNet, with a small advantage for EQTransformer on teleseismic data. Furthermore, we conduct a cross-domain study, analyzing model performance on data sets they were not trained on. We show that trained models can be transferred between regions with only mild performance degradation, but models trained on regional data do not transfer well to teleseismic data. As deep learning for detection and picking is a rapidly evolving field, we ensured extensibility of our benchmark by building our code on standardized frameworks and making it openly accessible. This allows model developers to easily evaluate new models or performance on new data sets. Furthermore, we make all trained models available through the SeisBench framework, giving end-users an easy way to apply these models.

Plain Language Summary

The first step in many seismological workflows is identifying if a signal contains an earthquake, and at which time which type of seismic wave arrived. These steps are known as event detection, phase identification and phase picking. In recent years, machine learning methods, in particular deep learning methods have been developed, showing promising performance on these tasks. However, so far these models have not been compared systematically in a quantitative way. Here we evaluate the performance of six deep learning models on eight datasets. Additionally, we compare them to a traditional picking algorithm not using machine learning. From our results we identify that the models EQTransformer, GPD and PhaseNet perform best. As in many use cases no picker trained on the target region will be available, we further evaluated how well models are transferable across regions. We identified that transfer across regions works well as long as the distance ranges stay similar. To foster application of the results, we make all our trained models available through the SeisBench framework.

1. Introduction

Detecting events and picking seismic phases is at the core of many seismological workflows (Bormann, 2012). These tasks are required for both post-hoc and real-time analysis. Due to the well-defined nature of these tasks they represent one of the first fields of application of neural networks in seismology (Bergen et al., 2019; Kong et al., 2019). In recent years, several deep learning models for detection and phase picking have been published (e.g., Mousavi, Zhu, et al., 2019; Ross et al., 2018; Soto & Schurr, 2021; Woollam et al., 2019; Zhu & Beroza, 2019). Their excellent performance can largely be attributed to the very large training datasets, with millions of publicly available, manually annotated picks. A similar abundance of data has led to breakthroughs across domains (LeCun et al., 2015) in the last decade, as deep learning, even among machine learning methods, profits particularly from very large datasets (Sun et al., 2017).

The deep learning based seismic detection and picking methods published so far differ in multiple aspects: their architectures, their training datasets and their task definitions. These differences currently make it impossible to compare results across publications, in particular as most publications evaluate their model on a single data set only. Furthermore, it is often not possible to anticipate how the model will perform on new data that differ from the training data in some characteristics. Therefore, users seeking to apply deep learning for picking will have difficulties selecting the appropriate model for their task. Throughout this paper, we will refer to evaluation as "in-domain" when training and test sets come from the same data set, and as "cross-domain" otherwise.

This study aims to address these issues, by offering a comprehensive benchmark of deep learning methods for detection, picking and phase identification. In particular, we focus on single station methods, that is, methods that do not incorporate data from different seismic stations for their picking decision, as not sufficiently many multi-station picking methods have been published yet. We compare seven models (one classical automatic picking algorithm, six deep learning) on eight datasets to gain a detailed understanding of the models' advantages and disadvantages, when applied for particular tasks or types of data. We analyzed datasets of different sizes, from different regions, with different arrival-time picking procedures and include a mix of local, regional and teleseismic arrivals. The deep learning models differ in several points: architecture, with convolutional, recurrent and attention based networks; input representation, in time or frequency domain; output representation, as point or sequence labels; model size, from few layers to very deep models. We employ consistent training/development/testing splits and a comprehensive parameter selection strategy, to ensure a fair comparison. We built the benchmark on the SeisBench platform, which we introduce in a companion paper (Woollam et al., 2021), and pytorch lightning (Falcon et al., 2019). Using these frameworks, the benchmark itself is built in a modular way, allowing to add both new datasets and new models to the evaluation easily. By publishing all code for training and evaluating the models we hope to enable developers of future models to compare their performance to a wide range of known results with minimal effort.

For deep learning pickers there can be a gap between the development of novel methods and their widespread adoption by practitioners. SeisBench aims to close this gap by offering a unified and simple API, that is, a standardized programming interface, for applying deep learning models to seismological tasks (Woollam et al., 2021). As part of SeisBench, we make available the model coefficients for all models trained in the context of this benchmark study. Together with our analysis of these models, this enables practitioners to easily select and load the model that is best suited for their specific application scenario. As this paper is aimed at both machine learning researchers and users with less machine learning expertize, we strive to give a complete description of our evaluation methods, while also providing short explanations for the key ML terms used.

2. Data and Methods

2.1. Tasks and Evaluation Metrics

Multiple deep-learning models for event detection, phase identification and onset picking have been proposed. However, these models differ with respect to the length of the input waveform and the output specification. To make the models comparable, we defined three common tasks and define for each model how it is applied to the task. These tasks are used to evaluate the models. However, the models might use different data selection and optimization targets in the training phase. Note that the model training is not tailored to these tasks and is described below.

2.1.1. Task 1—Event Detection

Given a 30 s window of a seismic waveform, determine if it contains an event onset, that is, a first arrival. We exclude coda examples as it is unclear whether they should be labeled as event or noise.

We evaluate the first task using receiver operating characteristics (ROC) and the corresponding area under the curve (AUC). The ROC describes the true and false positive rates across all possible decision thresholds. We use the ROC because, depending on the application scenario, different trade-offs between false positive rate and true positive rate are required. For example, when using a simple pick association algorithm in a region with low seismic activity, a false positive rate below 0.01 might be required, while when using a hyperbolic pick association (Woollam et al., 2020) in a seismically active region, a false positive rate of 0.05 might be absolutely fine. This tuning can be achieved by using different decision thresholds. To complement the ROC with a single number that can easily be compared, we use the AUC, which gives an average performance across all possible thresholds. An AUC of 1 indicates a perfect model, an AUC of 0.5 a coin toss.

As a further statistic, we present some results using the F1 score. The F1 score is defined as the harmonic mean of precision, the fraction of correct detections among all detections, and recall, the fraction of detections among all possible detections. It is therefore a combined measure for both the sensitivity and the specificity of a model. The F1 score depends on a choice of decision threshold. Where F1 scores are used, we optimize the decision thresholds appropriately as described later in Section 2.5.

2.1.2. Task 2—Phase Identification

Given a 10 s window containing exactly one phase arrival, determine if it is a P or an S phase. We do not further differentiate among different P or S phases such as Pn and Pg.

In contrast to the detection task 1, where there is a clear assignment of positive (event) and negative (noise) class, task 2 is symmetric, that is, the assignment of P and S phases to positive and negative class is ambiguous. As AUC, ROC, and F1 score are all sensitive to the class assignment, they are therefore not suitable for task 2. Instead, we use the Matthews correlation coefficient (MCC). MCC is a symmetric metric with values between −1 (total disagreement) and 1 (full agreement). It is calculated as the correlation coefficient of the confusion matrix and is a well suited measure for binary classification performance even in case of class imbalance. As MCC is sensitive to the decision threshold, we optimize the threshold as described in Section 2.5.

2.1.3. Task 3—Onset Time Picking

Given a 10 s window containing exactly one phase arrival of known type (P or S), determine the onset time.

For evaluating task 3, we use the residuals, that is, the differences between ML-pick time and the reference pick time provided in the data set. We analyze the residual distribution using histograms for visual inspection, and three metrics: the fraction of samples with high residuals (>0.45 s for regional, >1.5 s for teleseismic), the root mean squared error (RMSE), and the mean absolute error (MAE). We evaluate both RMSE and MAE, to include one metric that is sensitive to outliers (RMSE) and one that is not (MAE).

For each of the eight datasets and the three tasks, we generate a set of evaluation targets. For task 1, we generate noise examples from noise traces, if present in the datasets, or otherwise use windows before the first annotated arrivals in the other traces. Each evaluation target consists of a three-component waveform window and the associated label. Models are allowed to use waveforms outside the provided window if they are available in the data set.

2.2. Datasets

We use eight datasets currently included with SeisBench for the benchmark. Among these datasets, six contain only data from events at local-to-regional distances (here used loosely for events with <10° epicentral distance): ETHZ (Woollam et al., 2021), INSTANCE (Michelini et al., 2021), Iquique (Woollam et al., 2021), LenDB (Magrini et al., 2020), SCEDC (Southern California Earthquake Center, 2013) and STEAD (Mousavi, Sheng, et al., 2019). The other two data sets primarily consist of data from events at teleseismic distances (>10° epicentral distance), although including some regional data as well: GEOFON (Woollam et al., 2021) and NEIC (Yeck & Patton, 2020). The data set sizes range from 13,400 traces (Iquique) to more than 8 million traces (SCEDC). For a more detailed description of the datasets, see (Woollam et al., 2021) and Table 1 therein.

All data sets except LenDB contain manually labeled P and S arrivals. Therefore, we exclude LenDB from tasks 2 to 3, phase identification and arrival time picking. We note that although these picks are manually labeled, their exact time is subject to filter selection and human judgment. Therefore minor discrepancies are to be expected even in case of multiple well-trained human analysts. Even more detailed phase identification, differentiating, for example, between Pn and Pg, are available for the ETHZ and GEOFON datasets. However, within this study, we do not take this fine-grained information into account. We exclude the NEIC and the Iquique datasets from evaluation for task 1, event detection, as they do not contain either noise examples or sufficiently long waveforms

Table 1
Description of the Models Studied

	BasicPhaseAE	CRED	DPP	EQT	GPD	PhaseNet
# Params	33,687	293,569	199,731/ 546,081/ 21,181	376,935	1,741,003	23,305
Type	U-Net	CNN-RNN	CNN/ RNN/ RNN	CNN-RNN-Attention	CNN	U-Net
Training set	N. Chile	N. California	N. Chile	STEAD	S. California	N. California
Orig. train size	8,800	440,000	65,700	1,100,000	3,375,000	623,000
Orig. weights	No	Yes	No	Yes	Yes	No
Reference	(Woollam et al., 2019)	(Mousavi, Zhu, et al., 2019)	(Soto & Schurr, 2021)	(Mousavi et al., 2020)	(Ross et al., 2018)	(Zhu & Beroza, 2019)

Note. The number of parameters refers to the total number of trainable parameters. Note that these numbers might deviate slightly from those published by the original authors due to differences in the underlying frameworks. For DeepPhasePick (DPP), information delimited by slashes indicate Detector/P-Picker/S-Picker networks. The training size corresponds to the approximate number of training examples used in the original studies. The row "Orig. weights" indicates whether original weights were published and are available in SeisBench. For PhaseNet, weights were published by the authors, but these weights are not integrated into SeisBench due to technical issues. No original weights are used within this benchmarking study.

before the arrival to use as noise. They are, however, used when training the models for evaluation of cross-domain performance.

All data sets have predefined splits into training, development and test sets that are available through SeisBench. We use these splits throughout all experiments. Splitting procedures vary between the data sets and are described in (Woollam et al., 2021). Please note that the development set is sometimes referred to as validation set in literature.

We resample all data sets to 100 Hz sampling rate if necessary, as this is the original sampling rate used for all models evaluated here. We note that model performance will be dependent on the sampling rate, but leave this aspect to future study. To fit the training data and models into 500 GB of main memory, we only train on 90% of the SCEDC training set (87% for EQTransformer), but use the full development and test set.

2.3. Models

We evaluate six models for detection and five of these as well for phase identification and onset picking.

BasicPhaseAE (Woollam et al., 2019) is a convolutional network for phase detection and onset picking. It uses a U-Net (Ronneberger et al., 2015) like structure. Input to BasicPhaseAE are 6 s waveforms at 100 Hz and the output are prediction curves for P and S phases and noise with the same length. BasicPhaseAE was designed to be trained on small datasets and therefore has few parameters to avoid overfitting. It was originally trained and evaluated on a data set of 11,000 P/S-pick pairs from the Iquique region in Northern Chile. For task 1, we use 1 minus the noise probability as the probability of a phase (P or S) being present. For task 2, we use the ratio of the peak of the P and S predictions. For task 3, we use the peak position of the relevant phase prediction.

CNN-RNN Earthquake Detector (CRED; Mousavi, Zhu, et al., 2019) is a pure detection network, that can not be used for phase identification or onset picking. CRED operates on spectrograms of 30 s waveforms at 100 Hz sampling rate. Internally, CRED uses convolutional neural network layers (CNN) and long short term memory units (LSTM). It outputs a prediction curve of 19 samples, indicating whether an earthquake was detected at different times in the signal. For training, earthquake detection labels are defined based on the P and S arrivals, that is, detections start at the P arrival and last for 2.4 times the P to S time. For all datasets without S picks or with teleseismic P arrival, we redefined the detection labels to start at the P arrival and last 20 s. CRED was originally trained on 550,000 event seismograms and 550,000 noise seismograms from Northern California. CRED is only tested for task 1, for which we use the peak of the detection.

DeepPhasePick (DPP; Soto & Schurr, 2021) is a collection of models for event detection and phase picking. For detection, it uses a CNN structure with depth-wise separable convolutions, which assigns probabilities for noise, P and S phases to 5 s waveform windows. Once a P or S arrival is detected, DPP applies the respective picking network. The picking networks consist of two bidirectional LSTM layers and a pointwise applied fully connected layer. For picking, the labels are encoded as step functions, with values zero before the onset and one afterwards. To determine the picking time from the prediction trace, the first prediction sample exceeding 0.5 is used. The threshold of 0.5 is taken from the original publication. The three networks, for detection, P picking and S picking are trained separately. For our study, we use the detection network in tasks 1 and 2. In task 3, we use the respective pick networks for P and S picks. We do not use the detection network for task 3, as the window selection for task 3 already gives a good prior on the pick position. We did not train DPP for S wave picking on GEOFON, due to the very low number of S picks in the data set. This issue is not present for the other models, as they are not exclusively trained on one type of arrival. DPP was originally trained on 25,647 P-phase, 25,647 noise, and 14,397 S-phase windows around the 1995 $M_w = 8.1$ Antofagasta and 2007 $M_w = 7.7$ Tocopilla earthquakes in Northern Chile. The original publication of DPP includes an extensive hyperparameter search, that is, an optimization for the model configuration, with a particular focus on the model architecture. As our data sets are considerably larger, we are not able to conduct such an optimization here. Therefore, we chose optimal hyperparameters from the published study, giving us the opportunity to evaluate their transferability to other tasks.

Earthquake transformer (EQTransformer; Mousavi et al., 2020) is a model for joint event detection, phase detection and onset picking. EQTransformer operates on 60 s waveform windows at 100 Hz sampling rate. The output of EQTransformer are three prediction traces of 60 s length at 100 Hz sampling rate, each denoting the probability of a detection, P and S wave at a time. Internally, EQTransformer uses a stack of CNNs, LSTMs and self-attention layers. In training, EQTransformer makes intensive use of data augmentations. Here, we implemented the same augmentations with the same probabilities p: addition of Gaussian noise ($p = 0.5$), insertion of gaps ($p = 0.2$), dropping of channels ($p = 0.3$). Furthermore, EQTransformer applies a cyclical shift in time to the traces to allow for arbitrary positions of the P and S picks within the window. We use this augmentation for training on all datasets where not at least 60 s before and after most picks are available, that is, where the pick cannot naturally occur at any time in the trace. These datasets are INSTANCE, LenDB, NEIC and STEAD. We use the same definition for the detection label as for CRED. EQTransformer was originally trained on STEAD. For task 1, we use the output of the detection prediction. For task 2, we use the ratio of the peak of the P and S predictions. For task 3, we use the peak position of the relevant phase prediction.

Generalized phase detection (GPD; Ross et al., 2018) is a phase identification model with a short input window of only 4 s at 100 Hz sampling rate. For the window, GPD gives one prediction as P, S or noise. Originally, GPD high-pass filters the input waveforms at 2 Hz. In contrast, here we use a high-pass filter at 0.5 Hz to take into account that our data sets contain events with lower frequency, in particular in the teleseismic case. When applying the trained model with a sliding window, the model can also be used for onset detection. In the original implementations, arrivals were guaranteed to be between seconds 1 and 3 of the input window. As this can not be guaranteed in our setup, we use a slight modification of the GPD target and loss function used for training the model. Instead of assigning a class, that is, noise, P or S, to a window, we assign a probability to each of the classes. Probabilities for P or S are 1 if the pick is in the center of the window and decline with a Gaussian kernel of width 0.5 s. To accommodate the modified label definition, we use a multi-class cross-entropy loss for training, similar to the original loss. For completeness, we provide full results with the original target definition and loss in the Tables S1–S4 in Supporting Information S1. For all tasks, we apply the trained model with a sliding window and a stride of 5 samples, that is, 0.05 s. While an even smaller stride might lead to a slight improvement in picking accuracy, it would also come at a considerably higher computational cost, for example, a stride of 1 would be five times as expensive. We consider stride 5 as a reasonable balance between accuracy and computational load. GPD was originally trained and evaluated on 4.5 million seismograms from Southern California with an even distribution between P arrivals, S arrivals and noise. For task 1, we use 1 minus the noise probability. For task 2, we use the ratio of the peak of the P and S predictions. For task 3, we use the peak position of the relevant phase prediction.

PhaseNet (Zhu & Beroza, 2019) is a U-Net based model for arrival time picking. Its inputs are 30 s waveforms at 100 Hz and its outputs are probability curves for P and S arrivals of identical length. Notably, PhaseNet does not have any "global" connections, that is, despite its 30 s long input windows, the effective receptive field is only approximately 4 s long. This means, that predictions at each time are based on relatively small parts of the input data. From its structure, PhaseNet is fairly similar to BasicPhaseAE, which has been published afterwards. In contrast to BasicPhaseAE, PhaseNet uses slightly larger filter sizes, has a lower total number of filters, and includes residual connections. PhaseNet was originally trained and evaluated on 779,514 waveforms with P and

S arrivals from Northern California. For task 1, we use 1 minus the noise probability. For task 2, we use the ratio of the peak of the P and S predictions. For task 3, we use the peak position of the relevant phase prediction.

2.4. Training

We implemented the benchmark using the SeisBench framework (Woollam et al., 2021). All data sets and models are available in SeisBench and we use SeisBench's data generation module for building training pipelines. As the length of available waveforms often exceeds the expected input lengths of the models, we selected windows according to the following schema. In 2/3 of the cases, we selected a window such that at least one pick is guaranteed to be within the window. In the remaining cases, we randomly select a window from the full trace, which can also contain picks. This strategy ensures that the training labels are not dominated by noise examples, in particular for models with short input windows. We apply the same window selection for generating training and development examples. Note that this window selection strategy is not to be confused with the window selection for the three evaluation tasks. The window selection here selects windows of appropriate length for training the models. The windows selected for the tasks are identical across all models and their length is independent of the specific model.

We did not conduct any resampling between P and S arrivals, as the number of available P and S picks are always within a factor of 4 of each other, that is, no massive label imbalance is present. Only for the GEOFON data set, the label imbalance is strong, with 100 times more P than S arrivals. However, as only around 2,800 S arrivals are available for this data set, we found the number insufficient for effective upsampling. This label imbalance for GEOFON will be taken into account during evaluation. For the DPP pickers, we only train on examples containing either P or S picks, as the picker assumes that exactly one pick is within the window.

We train the models using the Adam optimizer (Kingma & Ba, 2014). We trained each model for 100 epochs, but with an additional limit of 48 hr wall time. This wall time limit only terminated training of some models on the very large SCEDC data set, but the validation loss curves strongly suggested that the models had been fully trained already nonetheless. In total, training and evaluation of the models, including cross-domain evaluations, took ~4, 000 GPU hours and ~260, 000 CPU thread hours. A breakdown of the computational costs for the different models is contained in the discussion.

2.5. Threshold and Hyperparameter Selection

We train all models on the training parts of the datasets. We keep the training, development, and test splits identical across all experiments. For evaluation, we use the model with the lowest loss, that is, the metric scoring the quality of the models predictions in training, on the development set. For task 1 we evaluate the area under the receiver operating characteristic (ROC-AUC or short AUC), which is independent of the decision threshold between noise and event. However, we also show optimal configurations in terms of F1 score for reference. For task 2 we choose the decision threshold between P and S phase to optimize the MCC. For each of the tasks we select the optimal thresholds on the development sets. We select thresholds independently for each model and data set. The thresholds are documented in the supplement (Tables S5–S6 in Supporting Information S1). For cross-data set analysis, we select the model based on the loss on the development set of the source data set and select the threshold on the development set of the target data set. If not indicated otherwise, all results reported are from the test parts of the data sets.

The performance of models often crucially depends on the choice of hyperparameters, that is, parameters controlling the learning process. Those can be either related to the model, such as number of layers or kernel sizes, or to the optimization algorithm, such as the batch size or the learning rate. Extensive hyperparameter optimization if often computationally expensive, but can lead to much better model performance. Therefore, for a benchmark, a similar budget for hyperparameter optimization should be provided for each model. Here, we keep the model architectures fixed and only tune the optimization hyperparameters of the gradient descent algorithm.

For all models, we used a fixed batch size of 1,024 samples. We ran all experiments with learning rates, that is, the step size for the gradient descent optimization algorithm, of 10^{-2}, 10^{-3} and 10^{-4}. Learning rates were kept constant during the full training. We selected the best performing model based on the development set of the target using AUC score (task 1), MCC (task 2) or standard deviation (task 3), both for in-domain and cross-domain analysis. Due to the huge computational demand, we were not able to conduct a larger scale hyperparameter study. Nevertheless, we are confident the test results are reliable, as similar hyperparameters were used in the original publications and the Adam optimizer is known to require only low levels of hyperparameter tuning (Kingma & Ba, 2014).

2.6. Baseline

For P onset time picking we include a traditional picker, the Baer-Kradolfer picker (Baer & Kradolfer, 1987), as baseline. The Baer-Kradolfer picker depends on four parameters: a minimum required time to declare an event, a maximum time allowed below a threshold for event detection, and two thresholds. For details on the parameters, we refer to (Baer & Kradolfer, 1987) or (Kueperkoch et al., 2012). We set the second threshold to half of the first threshold to reduce the number of parameters. Furthermore, the Baer-Kradolfer picker expects a bandpass filtered signal, therefore we add two additional parameters to be tuned, the high- and low-pass frequencies of a causal Butterworth-bandpass-filter.

In contrast to the deep learning models, the parameters for the Baer-Kradolfer picker can not be optimized using gradient descent. Therefore, we optimize parameters using Gaussian optimization with the RMSE as fitness function. We use 25 initial points and 500 further evaluations of the fitness function. To reduce computational demand, we only evaluate the fitness on 2500 P picks from the development set. We use the same 2500 P picks for each evaluation of the fitness function. This does not severely limit model performance, as the number of parameters is very low, with only five parameters to select.

We do not include classical baselines for either detection or S wave picking. For detection, the classical workflow includes a picker with rather high false positive rate, followed by event association. As our data sets and experimental setup do not allow to run the association step, this approach could not be employed here. Furthermore, association will likely also improve the performance of the deep learning pickers. We do not include a classical S picking baseline, as they usually require additional event information, for example, the approximate event-station distance or the back-azimuth, and careful manual tuning. Classical S pickers often have considerably more parameters than classical P pickers that need to be adjusted. For example, the picker presented by (Diehl et al., 2009) has 14 parameters (see Diehl et al., 2009, Table 4), not including the parameters for frequency filtering or quality classification. Tuning these parameters is not feasible with simple optimization, but requires informed judgment for each individual data set.

3. Results

3.1. Task 1-Event Detection

Results from in-domain analysis are available in Figure 1. Note that even though some models were already trained on some of the data sets in the original publications, we retrained all models from random initialization for comparability. On average, EQTransformer shows the best performance (AUC 0.964), closely followed by PhaseNet (0.957), CRED (0.951), GPD (0.949) and DPP (0.943). Further behind is BasicPhaseAE (0.771). The considerably worse performance of BasicPhaseAE compared to PhaseNet is surprising, given their very similar architecture. However, this shows that the shorter input windows for BasicPhaseAE, together with the shorter filters and missing residual connections lead to considerably worse results. Overall, CRED and EQTransformer show similar performance to each other for all data sets, and also GPD and PhaseNet show similar performance to each other. This can be explained with the similar architectures: EQTransformer is an extended version of CRED, using attention structures in addition to the CNN and RNN structures. Similarly, the architectures of GPD and PhaseNet both use CNNs on a relatively short input window, and in that sense PhaseNet can be interpreted as a GPD-like network with sequence instead of point predictions.

While EQTransformer and CRED on average perform better than GPD, PhaseNet and DPP, this results exclusively from better performance on LenDB and GEOFON. In both cases, we argue that the longer receptive fields of these models allow for the better performance. For GEOFON, the lower frequency signals of teleseismic arrivals can likely be better captured with these longer receptive fields, therefore representing a genuine improvement. For DPP, which performs particularly badly on GEOFON, the reason might also be that its hyperparameters were explicitly tuned on a local seismic data set, that is, a data set with very different characteristics. Even though

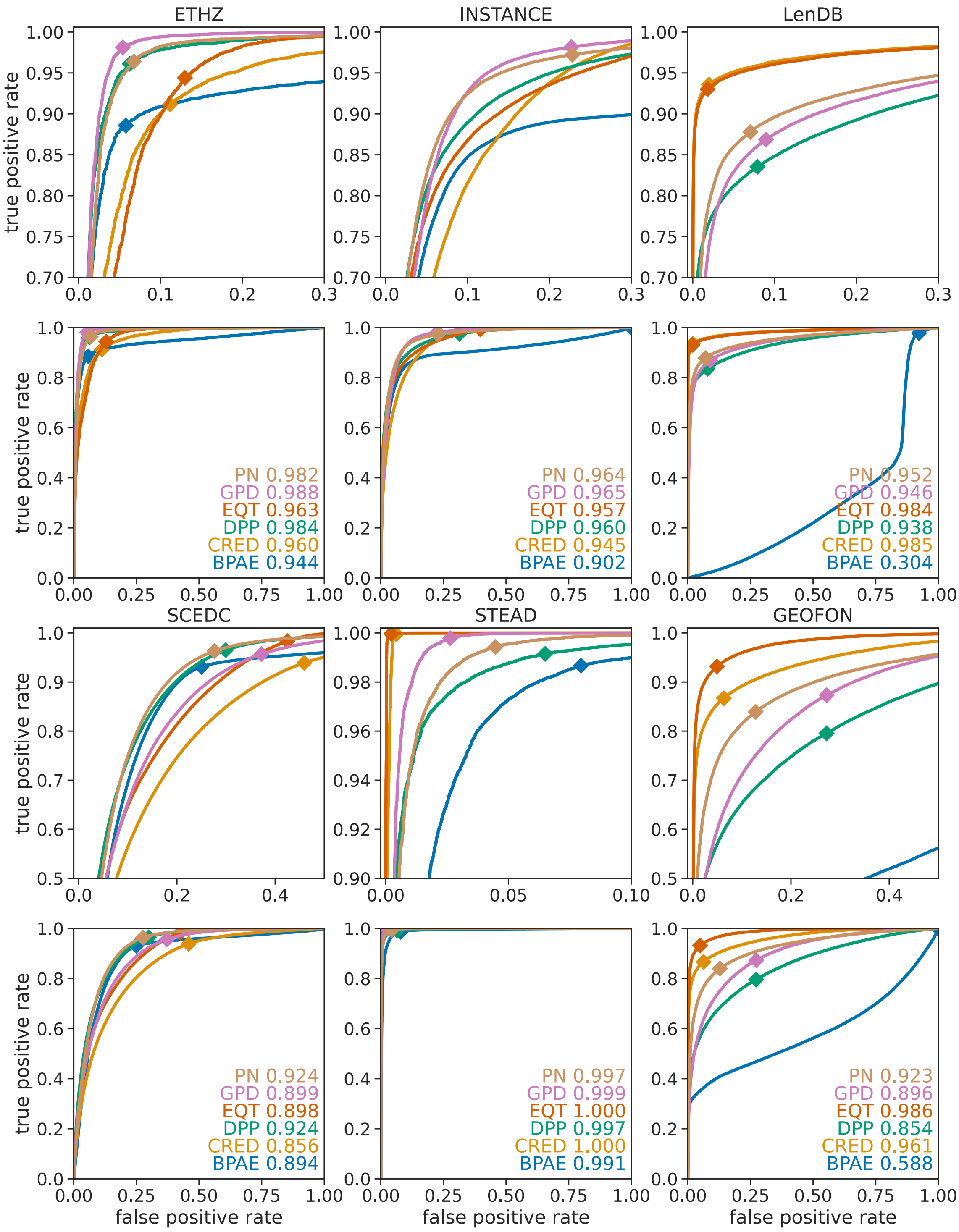

Figure 1. Receiver operating characteristics for detection results from in-domain experiments. Each curve shows one model. For each data set we provide two panels. The bottom panels show all datasets with equal scales, allowing to assess the full curves. The top panels show zoomed-in parts of the upper left corner, with zoom level selected individually for each data set to allow distinguishing the different models. Models were selected to maximize AUC score. Numbers in the corners indicate the test AUC scores. Markers indicate the point with the configuration associated with the highest F1 score.

the other models were also built for local or regional data, their hyperparameters were not tuned in a similarly systematic fashion as for DPP. In contrast to the GEOFON case, for LenDB, the global view of the input window gives CRED and EQTransformer the ability to learn the characteristics of the data set; the first arrival is always

Table 2
Phase Identification Results From In-Domain Experiments Given by Matthews Correlation Coefficient (MCC)

Model	BasicPhaseAE	DPP	EQTransformer	GPD	PhaseNet	∅
Data	MCC	MCC	MCC	MCC	MCC	MCC
ETHZ	0.77	0.89	0.97	0.92	0.91	0.89
INSTANCE	0.87	0.89	0.97	0.95	0.94	0.92
Iquique	0.81	0.91	0.99	0.98	0.96	0.93
SCEDC	0.84	0.82	0.96	0.93	0.91	0.89
STEAD	0.92	0.57	1.00	0.99	0.99	0.89
GEOFON	0.06	0.46	0.82	0.67	0.51	0.50
NEIC	0.70	0.76	0.96	0.84	0.81	0.81
∅	0.71	0.76	0.95	0.90	0.86	

Note. Averages are macro-averages, that is, the same weight is given to all models and data sets.

at a similar location within the 27 s input window, which gives them an (unfair) advantage over the other models. In addition, GPD, PhaseNet, and DPP suffer from the inaccurate pick times in LenDB, which result from using predicted arrival times for labeling. The characteristic training function for detection thus can start either too late or too early, meaning it is hard to minimize loss globally in training.

On INSTANCE, SCEDC, and STEAD no systematic differences between the models except BasicPhaseAE can be observed. However, on ETHZ GPD, PhaseNet, and DPP outperform CRED and EQTransformer by a small margin of ~0.02 points AUC score. The difference likely results from the definition of detections in the different models and the types of picks in the ETHZ data set. GPD, PhaseNet, and DPP simply calculate their detection score as one minus the noise probability. In contrast, CRED and EQTransformer provide explicit detection curves, that are fitted to predefined detection labels. Following the original publications, this detection label depends on P and S position. Therefore, at least for the regional datasets, detections were only declared if both P and S waves were annotated in the data set. In the ETHZ data set for a considerable number of traces either P or S annotations are missing, thus negatively affecting the performance of EQTransformer and CRED.

We observe a considerable variation of average scores on the different data sets. These differences can result both from the inherent difficulty of the task on a certain data set and the data set creation procedure. An example for the former would be that detection in a data set with a low average signal-to-noise ratio (SNR) will most likely be significantly harder than in one with high SNR, leading to below par average performance. Artifacts of the data set creation procedure could, for example, be different levels of quality control, or multiple annotators, leading to inconsistent annotations and thereby degraded performance. These effects also show that model performance can only be compared directly when all models are evaluated on the same datasets.

3.2. Task 2-Phase Identification

We evaluate task 2 using the MCC. In-domain results for all datasets and models are available in Table 2. For phase identification, EQTransformer shows the best results with an average MCC of 0.95, followed by GPD (0.90), PhaseNet (0.86), DPP (0.76) and BasicPhaseAE (0.71). These differences are considerably larger than for detection, indicating that the ability of EQTransformer to incorporate waveforms from a larger time window to understand the context indeed improves phase identification performance. In terms of data sets, phase identification is similarly hard for the five regional datasets (average MCC 0.90), more difficult for NEIC (0.81) and even more difficult for GEOFON (0.50). Again, the worse performance on the teleseismic data set most likely results from the lower frequency content of the arrivals. In addition, GEOFON only contains <3, 000 S picks in total, leading to a very small training set for these. BasicPhaseAE only exhibits marginally better performance than chance (0.06 compared to 0 for chance), indicating that the training set was insufficient to train the model.

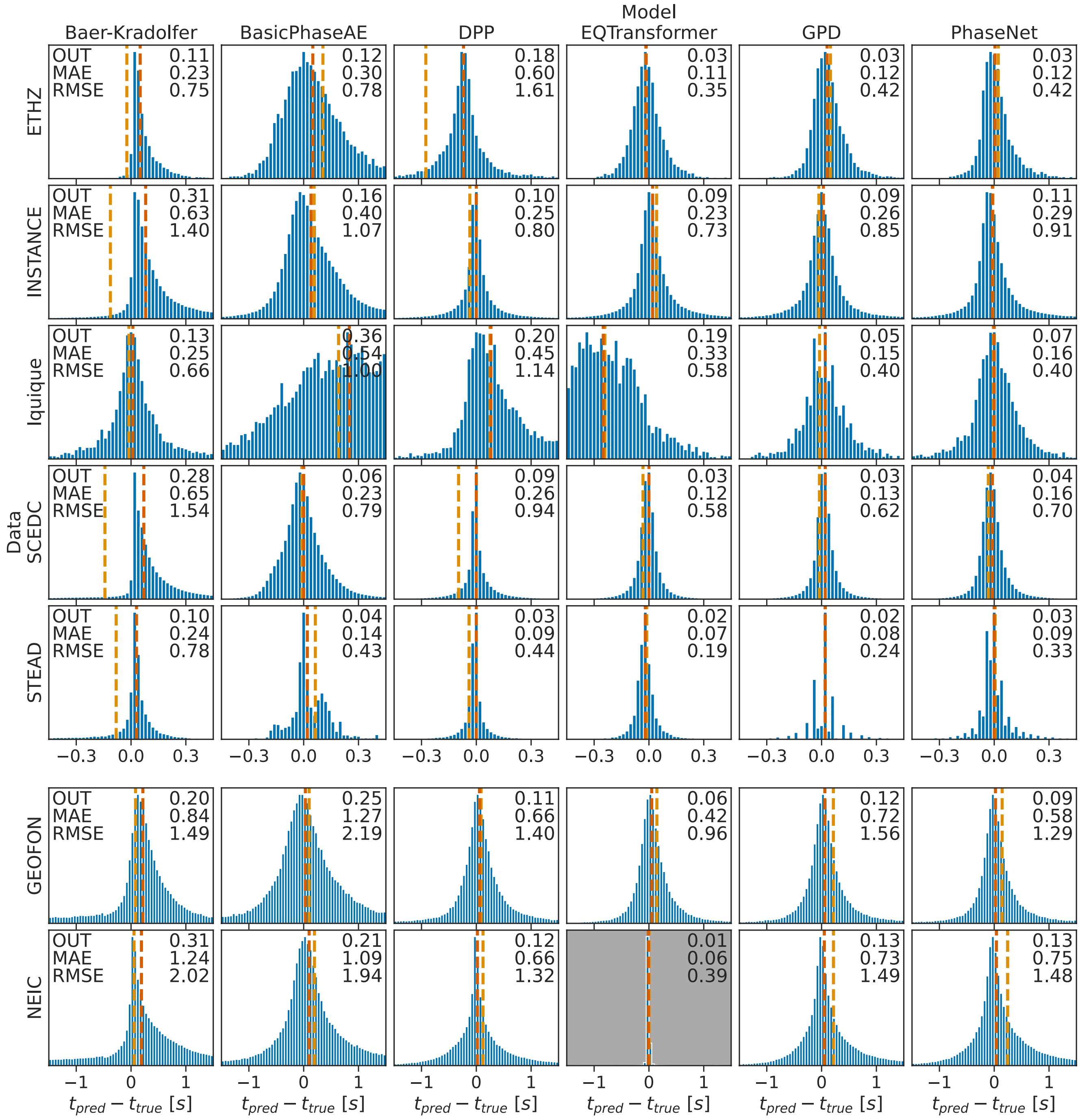

Figure 2. Histogram of P residuals from in-domain experiments. The numbers in the corner indicate the fraction of samples outside the plot boundaries, the mean absolute error (MAE) and the root mean squared error (RMSE). Vertical dashed lines show median (red) and mean (orange) of the residuals. For enhanced visibility, *y* axis scaling differs between all panels, therefore bar heights can not be compared across panels. Note also the different *x* axis scales for regional and teleseismic data sets. The EQTransformer on NEIC panel shows invalid results due to data constraints. For computation of the MAE, RMSE, mean and median of the Baer-Kradolfer picker we exclude picks within the first second of the window as these are mostly invalid. Due to the underlying obspy implementation, traces where no pick can be generated are picked early in the trace, leading to these picks. These picks are included for calculating the fraction of samples outside the plot boundaries.

3.3. Task 3-Onset Time Determination

In-domain P arrival picking results are shown in Figure 2, results for S arrival picking in Figure 3. On almost all datasets, for both P and S waves, EQTransformer performs best, although usually only with a small margin to GPD, PhaseNet and in some instances DPP. We note that the exceptional performance of EQTransformer on the

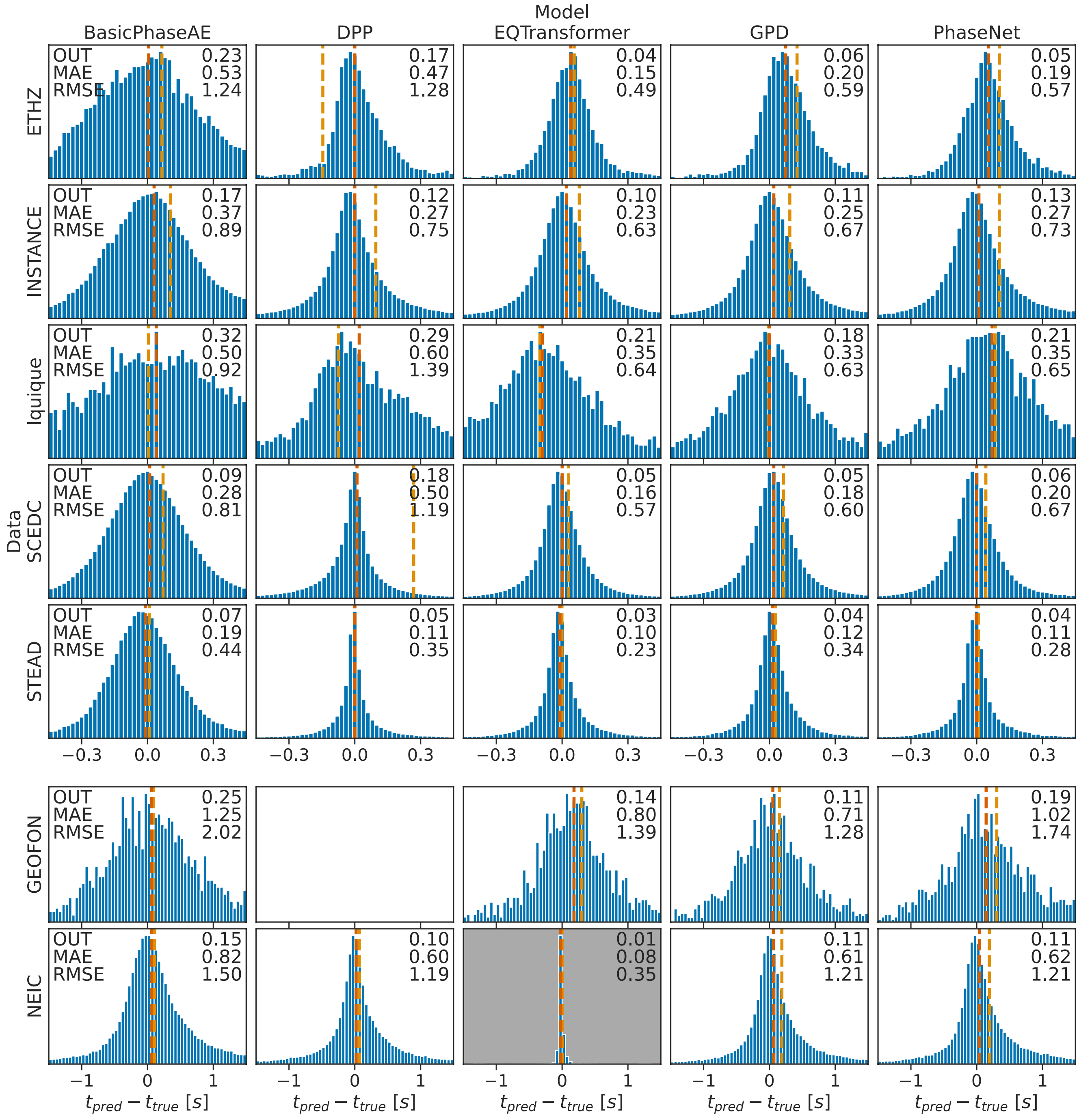

Figure 3. Histogram of S residuals from in-domain experiments. The numbers in the corner indicate the fraction of samples outside the plot boundaries (OUT), the mean absolute error and the root mean squared error. Vertical dashed lines show median (red) and mean (orange) of the residuals. For enhanced visibility, *y* axis scaling differs between all panels, therefore bar heights can not be compared across panels. Note also the different *x* axis scales for regional and teleseismic datasets. The EQTransformer on NEIC panel shows invalid results due to data constraints. Results for DPP on GEOFON are not available due to insufficient training data.

NEIC data set results from an artifact. In the NEIC traces, the picks are always at the same position within the 60 s input window. As EQTransformer has a global view of the full 60 s window, it does not need to learn to actually identify pick positions, but only to output a constant position. As this is not realistic for an actual application, and the other models can not reproduce this artifact due to their short input windows, this result needs to be excluded from the interpretation.

For both P and S waves, BasicPhaseAE performs worst on most datasets, with both RMSE and MAE often more than twice those of the best model. For P waves, DPP performs similarly to EQTransformer, GPD, and Phasenet on STEAD, GEOFON, INSTANCE, and NEIC, but considerably worse on ETHZ, Iquique, and SCEDC. DPP results on S waves mirror the P results, with competitive performance on INSTANCE, STEAD, and NEIC, but considerably worse performance than the best models on ETHZ, Iquique and SCEDC. From our observations, this behavior is likely caused by the unstable training of the LSTM in the DPP picker. The sequence length of 1,000 samples is fairly long for an LSTM and can lead to vanishing gradients. We observed that validation losses for DPP showed very high fluctuation over the training duration, with some training runs for some learning rates even failing to converge at all. As this is a random effect, it might be possible to improve performance to some extent by retraining.

Among EQTransformer, GPD, and PhaseNet, on the five regional seismic datasets, performance for both P and S waves is usually similar. On these datasets, for P waves, EQTransformer consistently shows 0.01 to 0.04 lower MAE than GPD and PhaseNet. For S waves, absolute performance differences are slightly larger, likely due to the higher absolute errors, but again EQTransformer shows the best performance on the regional data sets except Iquique.

On the teleseismic GEOFON data set, EQTransformer has considerably lower MAE for P waves (0.42 s) than GPD (0.72 s) and PhaseNet (0.58 s). Similar to task 1, this can likely be explained with the longer receptive field of EQTransformer being beneficial for the lower frequency content of teleseismic signals. However, this effect can not be observed for the S picks in GEOFON. Here, GPD performs best in terms of MAE (0.71 s), followed by EQTransformer (0.80 s) and PhaseNet (1.02 s). Due to the small number of S picks in the GEOFON data set (<3,000), these results are, however, not representative for the performance on teleseismic data in general. Rather these results show the generalization abilities of the models in a low training data scenario that is prone to overfitting.

Comparing the average performance of the models on the different data sets, taking into account only the three best models, there are consistent differences. For P waves, the lowest average MAE values occur for STEAD (0.08 s), ETHZ (0.12 s), SCEDC (0.14 s), Iquique (0.21 s), and INSTANCE (0.26 s). As for the performance differences in detection, this might partially be caused by differences in the data selection and labeling procedures. For example, the higher MAE for INSTANCE might partially result from the relatively large group of human annotators (~15–20 people) that contributed to the data set, leading to slightly lower consistency in the picks. Considerably higher MAE values were determined for GEOFON (0.57 s) and NEIC (0.74 s, excluding EQTransformer). These higher residuals can be explained with the teleseismic traces. For these, the onset times are often more challenging to pick due to their emergent onsets and lower frequency contents compared to mostly impulsive regional arrivals. Furthermore, the teleseismic arrivals in the GEOFON and NEIC datasets often exhibit worse signal to noise ratios.

The only regional data set with clear qualitative differences in the residuals compared to the others is the Iquique data set. This most likely results from the very small size of the data set (13,400 examples total). In particular, this affects BasicPhaseAE and EQTransformer, which both exhibit a systematic skew in the residual distribution. We investigated example predictions to understand this observation. In general, the predictive distributions of all models were wider than for the data sets with more training examples. For BasicPhaseAE we observe particularly large uncertainties. We speculate that the lack of residual connections in comparison to PhaseNet leads to a less efficient utilization of the training samples and thereby to an insufficiently optimized model. EQTransformer only shows mildly larger uncertainties compared to its counterparts trained on other datasets and, in contrast to the other models, the predictions are very smooth. This most likely results from the rather global view of EQTransformer compared to the localized approaches in GPD or PhaseNet. However, this also makes insufficiently trained EQTransformer instances more susceptible to systematic biases, rather than just wider uncertainties, as observed in the Iquique results. In conclusion, while the Iquique results might not be representative of model performance in favorable scenarios, they give valuable insights into the models' behavior in low training data cases.

For the S waves, average MAEs are approximately 25%–60% worse than the respective P residuals. We observe two exceptions with differing behavior. First, for INSTANCE, average S residuals are even slightly lower than the corresponding P residuals, leading to similar S residuals as for ETHZ or SCEDC. We think that this might be caused by a higher quality of the S picks compared to the P picks in INSTANCE (for an example of an inaccurate

P label, see Figure 6a). Second, for NEIC, average S residuals are even considerably lower than the P residuals. We speculate that this is an artifact of the data set creation: in teleseismic analysis, S waves are picked less regularly than P waves, which is also reflected in the lower number of S waves in the data set. However, this also means that S waves tend to be picked primarily in more favorable signal to noise conditions and at shorter distances, both of which lead to better defined pick onsets and in turn to lower residuals for the models.

We now analyze the residual histograms for P (Figure 2) and S arrivals (Figure 3). With the exception of the artifact for EQTransformer on NEIC, all residual distribution roughly resemble Laplacian densities with different widths, that is, distributions with a sharp mode and relatively heavy tails. Nearly all distributions are exactly centered on zero, that is, their mode is at zero. In some cases, the modes seem to be more centered than the mean error, indicating outliers to be systematically biased towards either too early or too late estimation. On all datasets, the residual distributions show no systematic differences between the best performing models, that is, no model exhibits, for example, a particular skew.

In contrast to the deep learning pickers, the classical Baer-Kradolfer picker shows considerably different features. First, the residual distribution is clearly non-symmetric, with considerably higher likelihood of the model picking slightly late than slightly early. This is expected, as the Baer-Kradolfer picker can not pick before the energetic onset. Similar behavior has also previously been reported, for example, in (Mousavi et al., 2020, Figure 10 in Supporting Information S1) or (Zhu & Beroza, 2019, Figure 6). Second, while the Baer-Kradolfer picker shows similarly low residuals as the best deep learning models for the majority of picks, it has a considerably higher fraction of outliers.

Given the latter finding, we compared the P residuals for the different pickers with respect to their SNR on STEAD (Figure S1 in Supporting Information S1). We use STEAD for this analysis, as it provides a large sample size, has high quality SNR annotations and good quality control for the labels. As expected, we observe that all pickers have decreasing residuals with increasing SNR. The majority of high residual picks results from low SNR examples. These results suggest that PhaseNet and DPP are more sensitive to SNR than GPD. For the classical Baer-Kradolfer picker, nearly all high residuals occur at lower SNR. This suggests that the biggest gain of deep learning pickers compared to a well tuned Baer-Kradolfer picker is their ability of picking successfully in low SNR conditions.

3.4. Cross-Domain Performance

So far, all presented results were in-domain results, that is, the models were trained and tested on data sets with mostly identical characteristics. However, in practice, one will often need to deviate from this principle and apply a model trained on one data set to different data. The performance of models in this cross-domain setup can be considerably different from the in-domain performance, because the characteristics of the target data might be different from those of the training data, but also because particularities in manual reference picking or selection might lead to unexpected biases in the trained models. To evaluate how the models fare in a cross-domain application, we perform a cross evaluation of the models, that is, we take each trained model and evaluate it on all test data sets on which it was not trained. Due to the vast number of results (# models × # source data sets × # target data sets = 336) for each task, we only report selected results in the main text (Figure 4, Figure 5). The full results for all tasks are available in the Tables S7–S12, Figures S2–S16 in Supporting Information S1.

The most obvious result from the cross-domain study is that, in general, cross-application works well if both datasets contain traces from the same distance range, although usually worse than in-domain application, but completely fails when applying regional models to teleseismic data. For example, in task 1, no model trained on a regional data set reaches an AUC score above 0.75 for detection on GEOFON, which is considerably below the in-domain performance of 0.85–0.99. When trained on NEIC, CRED (F1 0.86), EQTransformer (F1 0.78), GPD (F1 0.82), and PhaseNet (F1 0.82) achieve good F1 scores on GEOFON, mostly comparable with the in-domain performance. The opposite case, applying teleseismic models to regional data, also shows considerably worse performance than between regional datasets, but performance degradation is not as bad as from regional to teleseismic. However, this might at least be partially caused by the fact that both the GEOFON and NEIC data set contain at least some examples of regional picks. Notably, for determining pick onset times, regional to teleseismic application and vice versa work, at least to some extent, with residual distributions being wide, but clearly centered around 0.

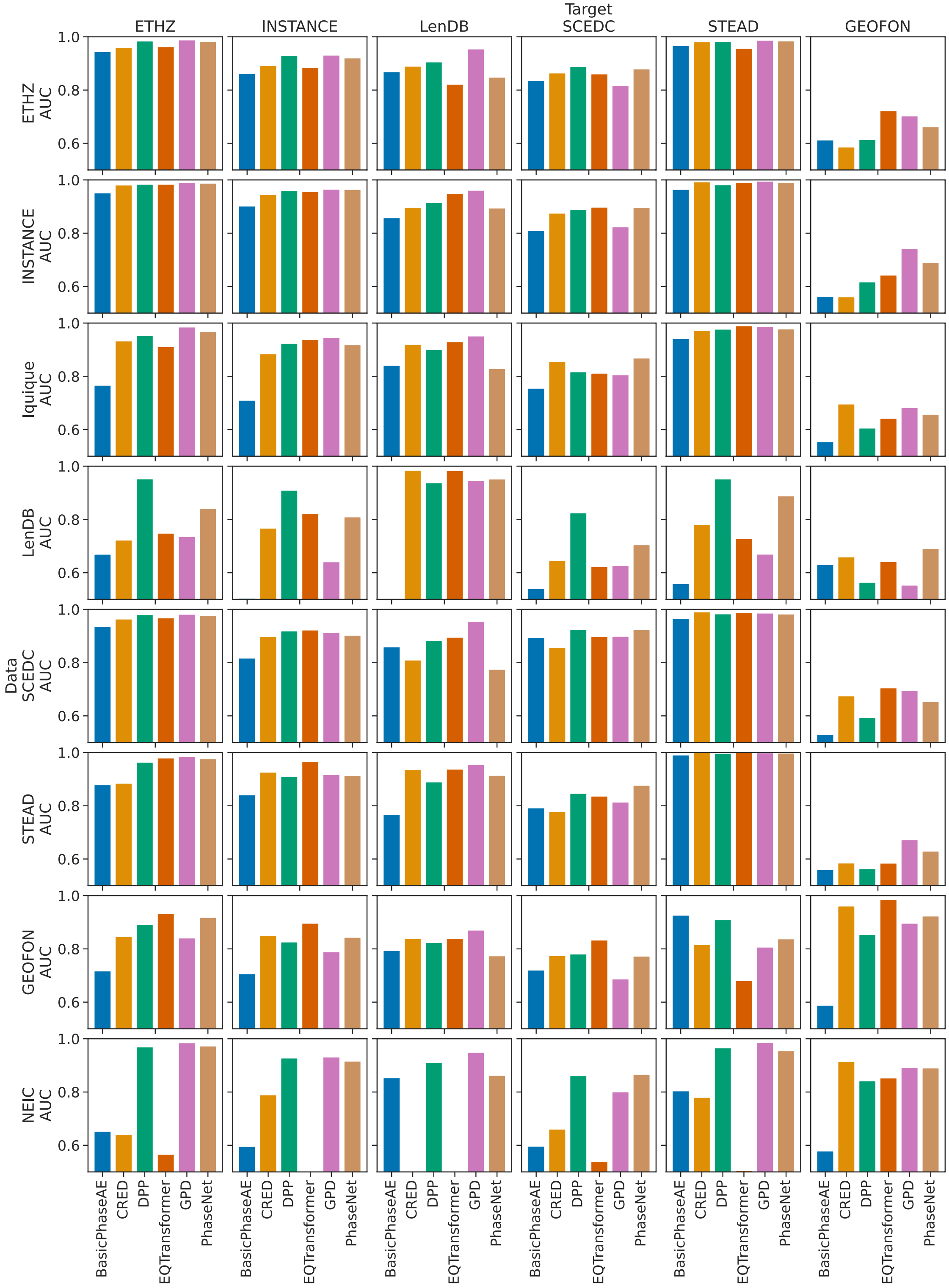

Figure 4. Area under the curve (AUC) scores for detection results from cross-domain experiments. Each panel shows one combination of training (row) and evaluation (column) data set, each bar one model. Models were selected to maximize AUC score on the evaluation data set. Note that the bars start at 0.5 instead of 0 as 0.5 is the AUC of a random model. Bars not shown have AUC values below 0.5. We include Iquique and NEIC as training but not as evaluation sets, as they do not have noise examples for the evaluations, but models can still be trained for detection on these datasets. A figure with the ROC curves and the numerical AUC values can be found in the supplement (Figure S2 in Supporting Information S1).

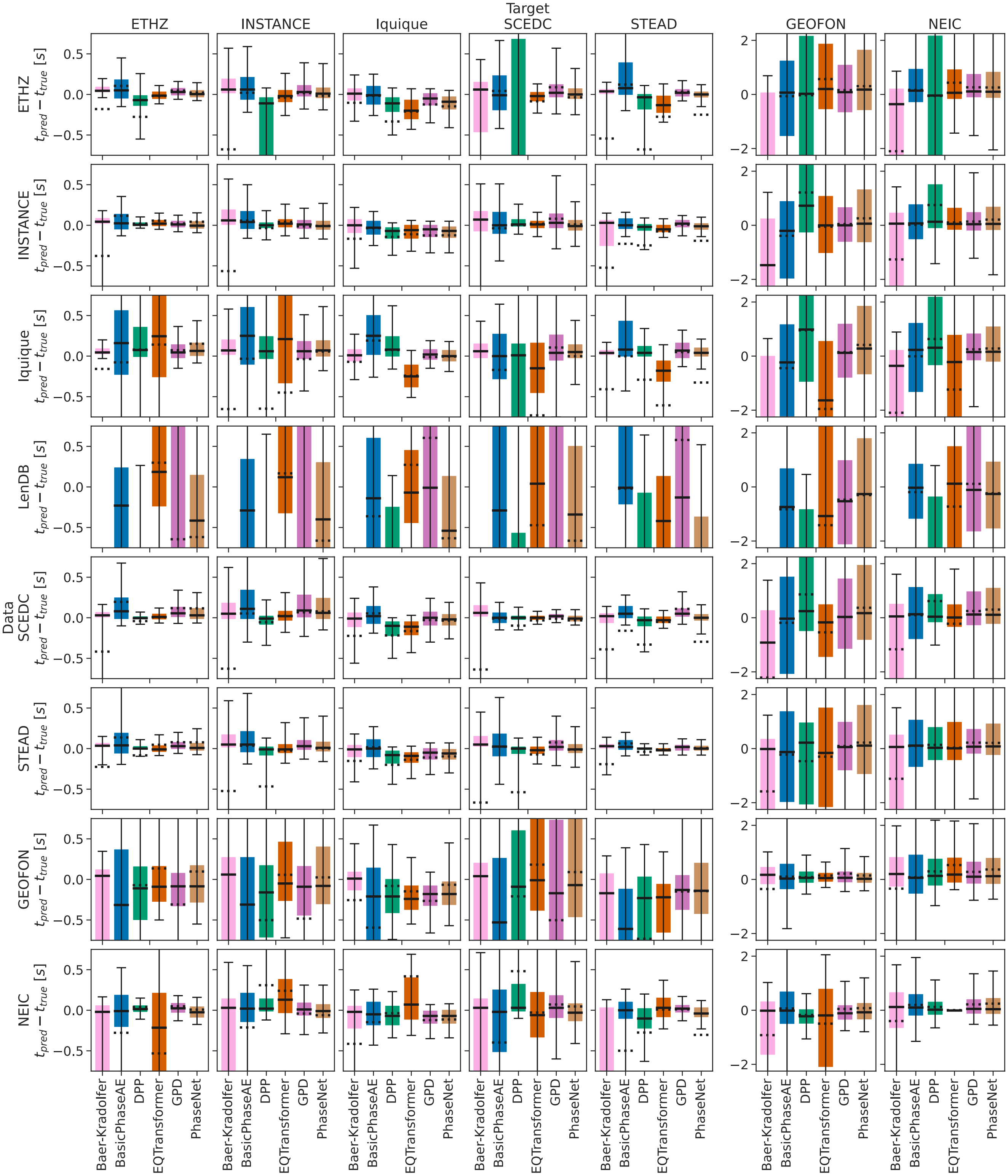

Figure 5. Distribution of P pick residuals from cross-domain experiments. Each panel shows one combination of training (row) and evaluation (column) data set, each bar one model. The solid bars show the interquartile range, the whiskers range from the 10th to the 90th percentile. The solid lines indicate medians, the dashed lines indicate means. Note that we include LenDB as a training data set, but not as an evaluation data set, because learning to pick on the predicted arrival times might be possible, while they do not serve as a sufficient reference for evaluating picking performance. An analogous plot for S pick residuals is available in the Figure S3 in Supporting Information S1.

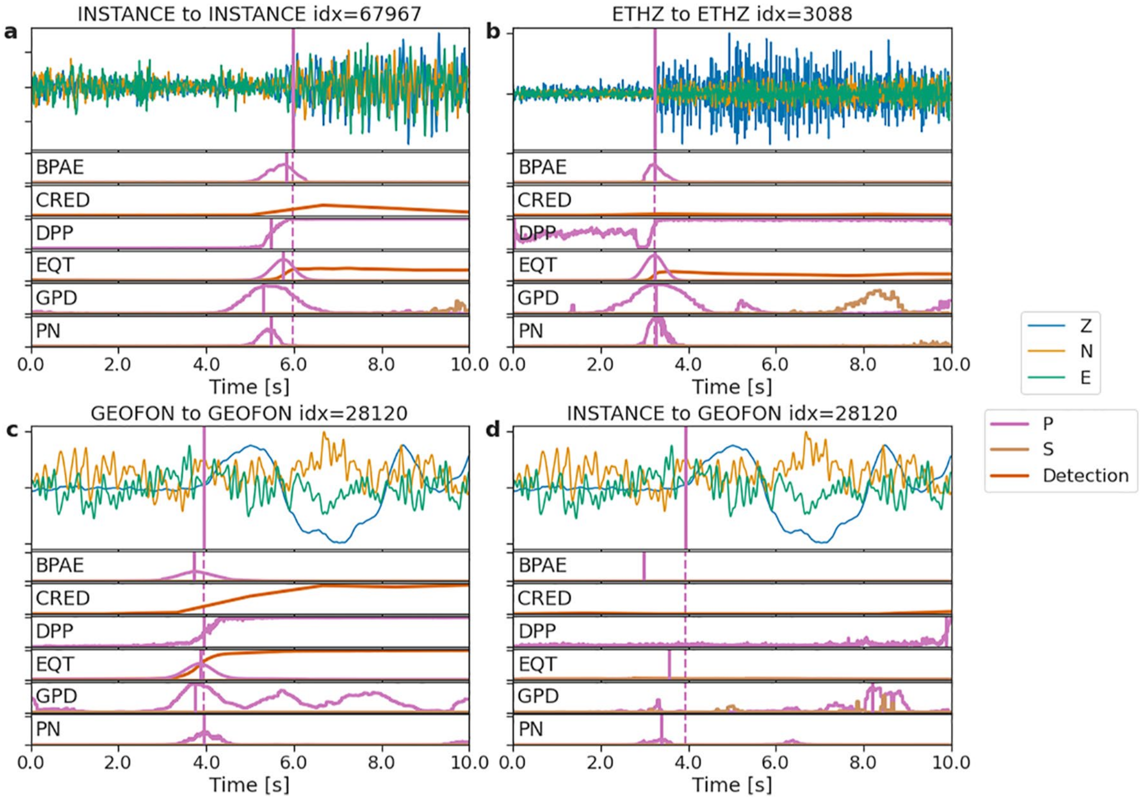

Figure 6. Example predictions for waveforms from different data set. (a) INSTANCE (~50 km epicentral distance, $M_L = 2.0$). Waveforms were highpass filtered at 2 Hz for better visibility of the onset time. (b) ETHZ (~65 km, $M_L = 2.2$). (c) GEOFON (~7, 300 km, $M_w = 5.7$). (d) Same waveforms as in c (GEOFON), but annotated with models trained on the INSTANCE data set. In all plots, vertical lines indicate pick times. The manual pick times are indicated by solid vertical lines in the waveforms plot and dashed lines in the prediction plots. Note that the ML-predicted pick for DPP in panel b is at 0.0 s.

The models trained on two of the regional datasets, Iquique and LenDB, show worse cross-domain performance than the ones trained on the other datasets. For Iquique, we suspect that this results from the small training set, leading to less well defined model parameters and lower generalization ability. For LenDB, this most likely results from the picks being obtained from travel-time calculation with a velocity model instead of manually labeled phase arrival times and the short input windows, leading to a data bias in the learned models. Notably, the DPP model strongly outperforms the other models for detection, when trained on LenDB. This is most likely caused by the label definition for detection with DPP: it only considers if a P/S pick is contained, but does not incorporate its position, making it less affected by the inaccurate pick positions in the data set. A data bias is also visible for EQTransformer trained on NEIC. As mentioned above, picks in the NEIC data set are always at the same position in the 60 s waveform traces, which can be recognized by EQTransformer. Applications of EQTransformer trained with NEIC data therefore perform considerably worse than other models with the same combination. However, the performance is still significantly better than would be expected in case of a constant pick location, indicating that the data augmentation employed in the training of EQTransformer indeed enables it to partially mitigate this issue.

For detection between different regional datasets (excluding LenDB and Iquique), PhaseNet performs best (AUC 0.947), closely followed by EQTransformer (0.941), GPD (0.937), and DPP (0.934). CRED shows similar performance to EQTransformer in most cases, but shows considerably worse performance in a few cases, in particular when trained on STEAD. For phase identification in the same setup, EQTransformer works best (MCC 0.95), outperforming PhaseNet (0.82), and GPD (0.76).

When evaluating picking performance, a clear feature is that all models incur an elevated level of picks with large differences (>0.45 s/1.5 s) in cross-domain application compared to in-domain application. The fraction of picks with such large differences often goes up to 10% even for datasets with generally good results in cross-application, for example, STEAD and INSTANCE. We suspect that this might be caused by differences in annotation practice, that is, some datasets might tend to miss weak earlier phase arrivals or might not exclude examples with overlapping events. For P arrival picking at regional distances, in most cases EQTransformer performs best. However, this result is not fully consistent, with several cases of GPD outperforming EQTransformer, sometimes even considerably, for example, from ETHZ or INSTANCE to STEAD. PhaseNet performs slightly worse than EQTransformer in most cases, but for some combinations works even considerably worse, for example, from SCEDC to ETHZ. However, we are not able to identify a systematic pattern when a specific model shows particularly good or bad cross-domain performance among the regional datasets.

In some cases, the cross-domain application also reveals biases in the data. For example, the P picks from all models trained on NEIC and applied to GEOFON are systematically too early. Conversely, trained on GEOFON and applied to NEIC, the arrivals are picked too late. This indicates that the two agencies exercise different judgment when picking arrivals.

On average, models trained on INSTANCE perform best in all tasks. However, differences for detection performance are usually minor with models achieving similarly good detection results when trained on STEAD, SCEDC and to some extent ETHZ, as when trained on INSTANCE. For P wave picking, performance of models trained on INSTANCE is usually better than on other datasets, even though training on STEAD also yields good performance. We note that this is in contrast to the in-domain performance, where models consistently showed worse performance on INSTANCE than on STEAD. Possibly the better performance for models trained on INSTANCE can be explained with the higher average number of waveforms per event (21 for INSTANCE, 2 for STEAD), leading to more diverse waveforms for each event. However, other reasons could dominate as well. For example, the evaluation datasets might on average be more similar to INSTANCE than STEAD, leading to the performance difference. Models trained on ETHZ and SCEDC often show worse detection performance, but perform nearly on par with models trained on INSTANCE for picking. The overall good performance of INSTANCE and STEAD can likely be explained with a combination of the quality, the size and the diversity of the datasets. Both datasets contain more than one million P picks and more than 700,000 S picks, giving plenty of training examples for the models. While SCEDC contains even more picks, the higher diversity in the picks in the other datasets likely leads to the better performance of models trained on INSTANCE and STEAD. For STEAD, this diversity is achieved by including picks from different regions. For INSTANCE, the diversity results from the complex tectonic setting of Italy, giving rise to both crustal seismicity and subduction events. These results indicate that for training a transferable model, it is highly desirable to include diverse picks.

4. Discussion

4.1. Model Comparison on Waveform Examples

For further insights into the models, we present several examples for which we compare the predictions from the different models (Figure 6, a–c in-domain, d cross-domain). All examples are from the test sets. Figure 6a shows predictions around a P pick from the INSTANCE data set. In this example, all models pick 0.5 s before the annotated onset. Indeed, when inspecting the waveforms after a highpass filter at 2 Hz, the pick seems to be annotated roughly 0.5 s too late in the original data set. This highlights the ever possible imperfections in the training datasets deriving from oversights of the analysts as in this case. On the other hand, it also illustrates how the deep learning models are able to learn a more consistent picking than present in the data set, because they can not reproduce differences between human annotators.

Figure 6b shows predictions around a P pick from the ETHZ data set. BasicPhaseAE, EQTransformer, GPD, and PhaseNet all correctly identify the P pick and determine the onset time to within 0.1 s. DPP fails to correctly identify the onset time. While the prediction curve correctly jumps from 0 to 1 around the pick, it already exceeds 0.5 within the first seconds, leading to an early pick. Predictions curves are smooth for EQTransformer, while predictions from the other models are considerably more rough. This presumably results from the long-range relationships modeled in EQTransformer but not accounted for in the other models. Except for GPD, no model detects additional potential picks within the trace. However, we observed that GPD operates best when choosing a very high detection threshold, such that the secondary picks would be ignored in practice.

Figures 6c and 6d both show the same teleseismic P arrival from the GEOFON data set. However, while Figure 6c shows the predictions for models also trained on GEOFON, Figure 6d shows predictions for models trained on INSTANCE, that is, without teleseismic arrivals in the training data. When trained on the GEOFON data, all models correctly detect the pick and its time with an error below 0.2 s. Notably, even in this scenario GPD detects multiple secondary P picks, indicating difficulties in differentiating the onset of the low frequency signal (~0.25 Hz) from its later wiggles. In contrast to the models trained on GEOFON, the models trained on INSTANCE consistently miss the arrival and return mostly arbitrary onset times. Only GPD produces detections, but as mentioned above, GPD detections should only be treated as true picks for very high confidence scores, which are not reached in this example. This example confirms the conclusion from the quantitative cross-domain analysis that cross-domain application only works well within the same distance range.

4.2. Cross-Domain Application With Adjusted Sampling Rate

As reported above, models trained on regional data perform poorly when applied to teleseismic examples. As a key reason, we identified the lower frequency content of the teleseismic arrivals. To further validate this hypothesis and to test a mitigation strategy, we analyzed rescaled versions of trained models. To this end, we apply the models trained on waveforms with 100 Hz sampling rate to waveforms with a considerably lower sampling rate. As the models do not know about timing, but only relative sample position, this effectively downscales the frequency ranges the models are looking for. We applied models trained on ETHZ, INSTANCE, Iquique, SCEDC, and STEAD to the GEOFON and NEIC data sets. We tried sampling rates of 20 and 40 Hz, that is, downscaling by factors of 5 and 2.5. For each combination of model, target and source data set, we selected the combination of learning rate and target sampling rate that showed best development scores and report the test results. As before, we only report selected results in the main text, but full results are available in the supplement (Tables S13–S15, Figures S17–S20 in Supporting Information S1).

For event detection (task 1) on GEOFON, resampling improves performance considerably, with AUC values up to 0.908 (GPD trained on INSTANCE). This is even above the in-domain score of GPD on GEOFON. While the non-resampled models achieve AUC scores often only slightly above a trivial classifier, the resampled models consistently outperform the trivial classifier. This also indicates that it might be reasonable to directly train the models on 20 or 40 Hz teleseismic data to achieve better performance. We did not conduct this test here, but leave it for future study. However, we note that CRED (0.961) and EQTransformer (0.986) still achieve better in-domain results. Still, this is close to the optimal cross-domain performance on GEOFON (0.915), achieved with CRED trained on NEIC. As NEIC is not applicable to task 1, we can only report the GEOFON results.

Similarly to detection, for P wave picking (task 3) we observe substantial improvements for EQTransformer, GPD, and PhaseNet. Best performance on GEOFON is achieved with PhaseNet trained on INSTANCE (MAE 1.01 s). For NEIC, the same model achieves an MAE of 0.90 s. For GEOFON, this is considerably inferior to the best model trained on NEIC without resampling (MAE 0.77 s) and the best in-domain performance (MAE 0.66 s). For NEIC, this score is superior to the best cross-domain model (MAE 0.95 s), but again does not outperform the optimal in-domain model (MAE 0.73 s). The error distributions are similar to the original error distributions, in particular they are centered around zero in most cases and they have heavy tails. The fraction with large residuals >1.5 s exceeds 20% or even 30% in most cases.

In contrast to the P wave case, we do not see an improvement for S wave onset determination with the resampled models. As reported above, the original models already performed considerably better for S wave detection than for P wave detection in both in- and cross-domain analysis. We explained this with the selection procedure, leading mostly to S picks with good signal-to-noise ratio and at moderate distances. For the same reason, we expect the typical amplitude spectra of regional and teleseismic datasets to be more similar for S arrivals than for P arrivals. Therefore, resampling does not yield performance improvements for S waves.

4.3. Computational Demand

A major consideration for deep learning models is their computational demand. We trained all models on identical machines, always using one Nvidia A100 GPU with 40 GB GPU memory. Except for the LSTM of the DPP picker and for evaluating GPD with the sliding window approach, we never got close to using the full memory, with all models staying well below 10 GB at a batch size of 1,024 samples. We note that larger batch sizes could have lead to better performance, however, we decided not to experiment with larger batch sizes as they were not employed in the original publications. Furthermore, large batch sizes often require the use of specific optimizers, for example, LARS (You et al., 2017), which tend to be less robust than the Adam optimizer.

To quantify the performance of the models, we measured run times for training and evaluation on INSTANCE. We chose INSTANCE for two reasons: it is large enough to ensure run times are not dominated by the overhead of epoch starts and ends, and it naturally comes at a sampling rate of 100 Hz and therefore does not require resampling on the fly, which could lead to CPU saturation. In our measurements, we did not include overheads from data preloading and model setup. We focus our analysis on throughput in training and evaluation. However, we note that for training performance, this only gives a rough guidance, as convergence speeds might differ between the models. For incorporating this aspect, models need to be compared provided a fixed compute budget. As for most applications the inference time is of bigger concern than the training time, we do not conduct this analysis here.

For training, the fastest models were BasicPhaseAE and the DPP detection network, both with 7,500 samples per second. The other models achieved, in decreasing order, PhaseNet (~6,300), GPD (~5,700), CRED (~4,900), DPP picker networks (~3,100), and EQTransformer (~2,600). For evaluation, again BasicPhaseAE achieved the highest throughput with ~6,700 sample per second. The other models achieved, in decreasing order, CRED (~4,900), DPP detection network (~4,800), PhaseNet (~4,800), DPP picker networks (~4,700), EQTransformer (~3,000) and GPD (~64). The very poor throughput of GPD in evaluation results from its sliding window approach. While all other models give prediction curves, GPD only gives point predictions and therefore needs to be applied repeatedly for each trace at regular intervals. We chose a stride of 5 samples, that is, applied GPD every 5 samples, as a good balance between accuracy and runtime. However, GPD is still slower than the next slowest model by a factor ~50. Overall, performance differences are within a factor of ~2, except for the evaluation of GPD. However, we note that performance differences might be considerably different on other hardware, in particular systems without GPUs and older hardware.

To provide guidance on model performance in annotation we applied the models to 24 hr of 100 Hz waveform data from a single station using SeisBench on Google Colaboratory. On CPU, annotation took 8 s for EQTransformer, 1 s for PhaseNet, 1 s for CRED, 2 s for BasicPhaseAE, and 267 s for GPD. On GPU, annotation took 1 s for EQTransformer, 1 s for PhaseNet, 1 s for CRED, 1 s for BasicPhaseAE, and 29 s for GPD. Note that we did not measure subsecond differences. The results nonetheless show the substantial improvements for large models on GPUs.

While the provided numbers give an indication of the model performance on our hardware/Google colaboratory, they do not immediately imply which resource is limiting the performance, that is, if the models are CPU bound, GPU bound or if a memory bus saturates. From observations of computing resources utilization during training and evaluation, we are confident that EQTransformer, the DPP pickers and CRED are GPU limited. The same holds true for GPD in evaluation. For the remaining models, we experienced GPU loads considerably below 100%, indicating a limitation on CPU or memory bus side.

5. Open Questions

The benchmark presented here gives an overview of the performance of a variety of models on different datasets. Nonetheless, it can by no means be considered exhaustive. In this section we discuss its limitations and open questions to be answered in follow up studies.

While our study analyzed both in- and cross-domain performance, it exclusively focused on event based analysis, that is, we only analyzed the performance on pre-selected windows. We chose this approach, as it is a good first-order proxy for the performance in practical applications and as it permits a thorough quantitative evaluation. Furthermore, it is closely related to a practical application scenario: post-processing. For example, NEIC applies deep learning models to the outputs of their STA/LTA pickers to refine pick times and estimate phase type and event-station distance (Yeck et al., 2021).

A different use case would be the application of these models to continuous data. While our results still give some guidance for this case, the different properties of this use case need to be taken into account. For example, in a continuous setup the false positive rate needs to be significantly lower than in post-processing, as only a fraction of all windows will usually contain arrivals, leading to a strongly biased prior distribution. Furthermore, assumptions used in the benchmark might become incorrect, for example, windows might contain multiple picks, in particular in dense aftershock sequences. A detailed analysis of the performance of the models applied to continuous data therefore should be conducted in a follow up study.

Another application case for deep learning pickers is real-time identification of earthquake arrivals. The results from this study only considered the post hoc performance of the models, not taking into account how early they would be able to identify an event onset. This aspect needs to be studied explicitly, before applying the models in a real-time/early warning scenario.

While the data sets studied here encompassed several world regions and also teleseismic arrivals, there also exist different classes of seismic signals. For example, recent years have seen a surge of data from nodal seismometers at local distances that were not covered within this study. Similarly, machine learning could be applied to induced seismicity, mine blasts, or volcanic signals. While some studies in this direction exist (Chai et al., 2020; Dong et al., 2020; Lapins et al., 2021), there is not yet a comprehensive study. To facilitate such a study, comprehensive benchmark datasets with rich metadata need to be assembled.

The majority of data used in this study were recorded on short-period or broadband instruments. However, a variety of recording devices are used in different scenarios nowadays, including geophones, ocean bottom seismometers, accelerometers or even distributed acoustic sensing. The results from our study do not allow to make reliable conclusion on the performance and characteristics on these types of recordings. To this end, further studies with targeted benchmark data sets are required. Similarly, we primarily analyzed model performance on three component recordings. However, several instruments either possess only single components, or have additional sensors such as pressure sensors or rotational sensors. The influence of these components should be assessed in future studies.

One of the objectives of this study was to analyze cross-domain performance of models, that is, the performance of a model trained on one data set and applied to another one. This scenario corresponds to a typical practical use case: waveform data is collected and should then be automatically analyzed. However, often at least some labeled data are available for a data set, for example, an analyst has picked a certain timeframe of a deployment. In this case, transfer learning can be applied to improve performance. In transfer learning, instead of training models from random initialization, one only fine-tunes a model pretrained on a large data set to a usually small target data set. This often leads to significantly better performance than either directly training on the target data set or direct application of a model trained on the large data set. Transfer learning has lately been shown to be beneficial in several seismological tasks (Chai et al., 2020; Jozinović et al., 2021; Münchmeyer et al., 2021).

While the general idea of transfer learning is simple, a wide variety of transfer learning schemes exist. Furthermore, it is as yet unclear which datasets are most suitable for pretraining models. These questions need to be analyzed in future work. As one result from our study we publish all obtained model weights. These can serve as a basis for transfer learning, both for future studies of transfer learning approaches and for practitioners seeking to tune models for their data. We expect this to be particularly beneficial when training on catalogs of limited size (<10,000 events).

6. Conclusions & Recommendations

In this study, we conducted a quantitative comparison of six deep learning based models for earthquake detection, phase identification and onset time determination. Using eight data sets—six of local and regional distance recordings and two mainly at teleseismic distances—we evaluated both in- and cross-domain performance. In conclusion, we found EQTransformer, GPD and PhaseNet to be the best performing models. Among these three models, EQTransformer shows considerably better performance for teleseismic data, likely due to its longer receptive field. GPD, while showing excellent performance, only achieves poor throughput in evaluation, making it only applicable for small data sets or with large computational resources being available. PhaseNet achieves similar performance to GPD, while providing significantly higher throughput. CRED and DPP also achieve very good detection performance. However, CRED, in contrast to the other models, is limited to detection, which makes it less appealing, in particular considering its close architectural similarity to EQTransformer. DPP is performing well on detection but shows considerably poorer performance for onset time determination.

The results of our study do not only represent a model comparison, but also give guidance which training datasets should be used in a cross-domain application, for example, when the pretrained weights provided with this study through SeisBench are used on new data. The most important factor is using a training data set from the appropriate distance range. Furthermore, the data set should be diverse in the waveforms contained, large, and have high quality annotations. For local to regional data, models trained on STEAD and INSTANCE generally showed best performance. Combined with the model discussion above, we would recommend using PhaseNet or EQTransformer for picking and detection on these data sets. For teleseismic targets, models trained on teleseismic data sets should be used. As our comparison only used two teleseismic data set, we can not give a clear recommendation here. However, caution needs to be exercised when using EQTransformer trained on NEIC due to the fixed pick positions. If only detections, but no pick locations are required, CRED trained on the datasets mentioned above is a viable alternative.

Our evaluation highlights the value of curated benchmark data sets. In particular, only testing on a variety of such sets allows reliable conclusions on the relative performance of the models. The study also shows that model performance can generally not be compared across data sets, because performance on a data set largely depends on its characteristics, such as the data selection, quality control or internal consistency. Notably, the influence of training and evaluation data can manifest differently for each task. For example, the absolute performance on task 1 (detection) is largely dependent on the data set used for evaluation, while the performance for task 3 (onset time determination) is primarily defined by the training data set.

Besides the performance evaluation presented in this paper, our study also yielded a rich collection of trained models. We make trained model weights for all combinations of data sets and models publicly available through the SeisBench framework. These models can be used directly by practitioners wanting to automatically pick their data, but they can also be used for further evaluation as discussed above. For the appropriate choice of models, this study should give a good guideline. For each model, we also provide suggested decision thresholds through SeisBench. Even though the optimal threshold will depend on the application scenario, the provided values give an orientation for threshold selection.

Data Availability Statement

SeisBench is available at https://doi.org/10.5281/zenodo.5568813 and https://github.com/seisbench/seisbench. The benchmarking code is available at https://doi.org/10.5281/zenodo.5795612 and https://github.com/seisbench/pick-benchmark. All data used can be accessed through SeisBench.

Acknowledgments
This work was supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition. J. Münchmeyer acknowledges the support of the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS). The authors thank the Impuls-und Vernetzungsfonds of the HGF to support the REPORT-DL project under the grant agreement ZT-I-PF-5-53. This work was also partially supported by the project INGV Pianeta Dinamico 2021 Tema 8 SOME (CUP D53J1900017001) funded by Italian Ministry of University and Research "Fondo finalizzato al rilancio degli investimenti delle amministrazioni centrali dello Stato e allo sviluppo del Paese, legge 145/2018." The authors thank Daniel Trugman and Mostafa Mousavi for their insightful reviews that helped to improve the manuscript. Open access funding enabled and organized by Projekt DEAL.

References

Baer, M., & Kradolfer, U. (1987). An automatic phase picker for local and teleseismic events. *Bulletin of the Seismological Society of America*, *77*(4), 1437–1445. https://doi.org/10.1785/bssa0770041437

Bergen, K. J., Johnson, P. A., Maarten, V., & Beroza, G. C. (2019). Machine learning for data-driven discovery in solid earth geoscience. *Science*, *363*(6433). https://doi.org/10.1126/science.aau0323

Bormann, P. (2012). *New manual of seismological observatory practice (nmsop-2)*. IASPEI, GeoForschungsZentrum. https://doi.org/10.2312/GFZ.NMSOP-2

Chai, C., Maceira, M., Santos-Villalobos, H. J., Venkatakrishnan, S. V., Schoenball, M., Zhu, W., & Team, E. C. (2020). Using a deep neural network and transfer learning to bridge scales for seismic phase picking. *Geophysical Research Letters*, *47*(16), e2020GL088651. https://doi.org/10.1029/2020gl088651

Diehl, T., Deichmann, N., Kissling, E., & Husen, S. (2009). Automatic s-wave picker for local earthquake tomography. *Bulletin of the Seismological Society of America*, *99*(3), 1906–1920. https://doi.org/10.1785/0120080019

Dong, L.-j., Tang, Z., Li, X.-b., Chen, Y.-c., & Xue, J.-c. (2020). Discrimination of mining microseismic events and blasts using convolutional neural networks and original waveform. *Journal of Central South University*, *27*(10), 3078–3089. https://doi.org/10.1007/s11771-020-4530-8

Falcon, W., Borovec, J., Wälchli, A., Eggert, N., Schock, J., Jordan, J., & Bakhtin, A. (2019). *Pytorchlightning/pytorch-lightning*. Zenodo. https://doi.org/10.5281/zenodo.3530844

Jozinović, D., Lomax, A., Štajduhar, I., & Michelini, A. (2021). *Transfer learning: Improving neural network based prediction of earthquake ground shaking for an area with insufficient training data*. arXiv preprint arXiv:2105.05075.

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980.

Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P. (2019). Machine learning in seismology: Turning data into insights. *Seismological Research Letters*, *90*(1), 3–14. https://doi.org/10.1785/0220180259

Kueperkoch, L., Meier, T., & Diehl, T. (2012). *Automated event and phase identification. New manual of seismological observatory practice 2. (NMSOP2)*. Retrieved from https://gfzpublic.gfz-potsdam.de/pubman/item/item_43230

Lapins, S., Goitom, B., Kendall, J.-M., Werner, M. J., Cashman, K. V., & Hammond, J. O. (2021). A little data goes a long way: Automating seismic phase arrival picking at nabro volcano with transfer learning. *Journal of Geophysical Research: Solid Earth*, *126*(7), e2021JB021910. https://doi.org/10.1029/2021jb021910

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Magrini, F., Jozinović, D., Cammarano, F., Michelini, A., & Boschi, L. (2020). Local earthquakes detection: A benchmark dataset of 3-component seismograms built on a global scale. *Artificial Intelligence in Geosciences*, *1*, 1–10. https://doi.org/10.1016/j.aiig.2020.04.001

Michelini, A., Cianetti, S., Gaviano, S., Giunchi, C., Jozinovic, D., & Lauciani, V. (2021). Instance–the Italian seismic dataset for machine learning. *Earth System Science Data Discussions*, *13*, 1–47.

Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C. (2020). Earthquake transformer—An attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, *11*(1), 1–12. https://doi.org/10.1038/s41467-020-17591-w

Mousavi, S. M., Sheng, Y., Zhu, W., & Beroza, G. C. (2019). Stanford earthquake dataset (stead): A global data set of seismic signals for ai. *IEEE Access*, *7*, 179464–179476. https://doi.org/10.1109/access.2019.2947848

Mousavi, S. M., Zhu, W., Sheng, Y., & Beroza, G. C. (2019). Cred: A deep residual network of convolutional and recurrent units for earthquake signal detection. *Scientific Reports*, *9*(1), 1–14. https://doi.org/10.1038/s41598-019-45748-1

Münchmeyer, J., Bindi, D., Leser, U., & Tilmann, F. (2021). The transformer earthquake alerting model: A new versatile approach to earthquake early warning. *Geophysical Journal International*, *225*(1), 646–656. https://doi.org/10.1093/gji/ggaa609

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). https://doi.org/10.1007/978-3-319-24574-4_28

Ross, Z. E., Meier, M.-A., Hauksson, E., & Heaton, T. H. (2018). Generalized seismic phase detection with deep learning. *Bulletin of the Seismological Society of America*, *108*(5A), 2894–2901. https://doi.org/10.1785/0120180080

Soto, H., & Schurr, B. (2021). DeepPhasePick: A method for detecting and picking seismic phases from local earthquakes based on highly optimized convolutional and recurrent deep neural networks. *Geophysical Journal International*, *227*(2), 1268–1294. https://doi.org/10.1093/gji/ggab266

Southern California Earthquake Center. (2013). *SCEDC*. https://doi.org/10.7909/C3WD3xH1

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the ieee international conference on computer vision* (pp. 843–852). https://doi.org/10.1109/iccv.2017.97

Woollam, J., Münchmeyer, J., Tilmann, F., Rietbrock, A., Lange, D., Bornstein, T., & Soto, H. (2021). Seisbench-A toolbox for machine learning in seismology. *Seismological Research Letters*, *81*(3), Retrieved from https://arxiv.org/abs/2111.00786

Woollam, J., Rietbrock, A., Bueno, A., & De Angelis, S. (2019). Convolutional neural network for seismic phase classification, performance demonstration over a local seismic network. *Seismological Research Letters*, *90*(2A), 491–502. https://doi.org/10.1785/0220180312

Woollam, J., Rietbrock, A., Leitloff, J., & Hinz, S. (2020). Hex: Hyperbolic event extractor, a seismic phase associator for highly active seismic regions. *Seismological Society of America*, *91*(5), 2769–2778. https://doi.org/10.1785/0220200037

Yeck, W. L., & Patton, J. (2020). *Waveform data and metadata used to national earthquake information center deep-learning models*. U.S. Geological Survey. Retrieved from https://www.sciencebase.gov/catalog/item/5ed528ff82ce2832f047eee6

Yeck, W. L., Patton, J. M., Ross, Z. E., Hayes, G. P., Guy, M. R., Ambruz, N. B., et al. (2021). Leveraging deep learning in global 24/7 real-time earthquake monitoring at the national earthquake information center. *Seismological Society of America*, *92*(1), 469–480. https://doi.org/10.1785/0220200178

You, Y., Gitman, I., & Ginsburg, B. (2017). *Large batch training of convolutional networks*. arXiv preprint arXiv:1708.03888.

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