Dear Yen Joe Tan, Mostafa and Reviewer C,

Thank you very much for your time and comments regarding our submission.

Our replies to your comments, suggestions and corrections are in this blue text style.

We have replied to all reviewer comments, made changes to the text as part of these replies, and corrected the resolution of the figures.

The main additional analysis we did was to run the case of using both 3-component waveforms and characteristic functions (8 channels) as input, as suggested by the reviewers. Wwe found that running the 8 channels input makes almost no difference in the results, as described below in our replies and in the revised text.

Best regards,

The authors.

Reviewer B:

Review of “Effects on a Deep-Learning, Seismic Arrival-Time Picker of Domain-Knowledge Based Preprocessing of Input Seismograms” by Lomax et al

The paper explores the effects of domain-knowledge input on a deep-learning seismic arrival picker. It discusses two methods for converting raw seismograms into picker characteristic functions (CFs) and analyzes the impact of domain-knowledge preprocessing on the performance of a deep-learning phase picker. It concludes that DL-pickers may have already learned to extract the most critical features from raw seismic data as they could achieve a very similar performance. This is an interesting study exploring the properties of DL-phase pickers from another angle and I think would be of the interest of the seismological community. Hence I recommend it for publication. However, I do have some comments and suggestions that may help improve the manuscript.

Major comments:

The primary motivation for using characteristic functions (CFs) in this study is to highlight earthquake signals, thereby facilitating their identification and enhancing robustness. Convolutional neural networks (CNNs), on the other hand, can automatically extract task-specific features from raw seismic signals through an interactive network training process. The observation of faster network convergence during training when CFs were used instead of raw data indicates the effectiveness of the incorporated domain knowledge.

However, considering that CFs undergo further post-processing throughout the CNN layers, there may not be much left for the network to learn, potentially limiting the scope for performance improvement. It is noteworthy that U-Net is a fully convolutional network whose primary strength lies in its hierarchical feature extraction capability.

We agree this is an important consideration. However, as we already noted and discussed in the manuscript, the network still needs to detect, classify and otherwise quantify features of potential phase arrivals. This is equivalent to the pick detection, timing, uncertainty estimation, first-motion determination, etc performed by a classic automated picker. Arguably, these are the most difficult and important tasks performed by any picker algorithm.

We also note that our primary goal is not to develop and test a new, improved picker, but to compare two ML picking procedures which only differ in pre-processing while otherwise introducing the fewest differences possible between the pickers.

We have added additional mention and discussion on these points in the Discussion (lines 765-769).

In light of this, it would be more meaningful to compare the performances of the following setups:

1) PhaseNet with raw seismic input versus a simple 2-layer CNN (for dimensionality reduction) and a 2-layer fully connected (FC) network with CFs as inputs. This comparison

would illustrate the effectiveness of engineered features (i.e., CFs) in comparison with automatically extracted features by PhaseNet.

In retrospect, after performing and documenting our study, we agree strongly that such a study is warranted and would be of interest. We already mentioned in the Discussion "Additional study might investigate the usage of “simpler” and “shallower” model-architectures than that of PhaseNet, while still feeding the DKPN input or similar." However, we see two fundamental problems with (greatly) extending the current study with a modified architecture:

1) The basic premise of our study is that we change only the input of an existing ML picker, and otherwise leave the picker unmodified: "we examine changes in the performance of a deep-learning picker when its raw seismogram input is modified using seismological domain-knowledge from classical pickers", "We otherwise make no change to the PhaseNet model", see our Figure 1. Making only this change helps allow meaningful comparison, analysis and understanding of changes in performance relative to the existing picker, as there is a rapid increase in complexity and potential causes for differences in performance as more changes are included. In particular, if we also change the architecture of the picker, then we are really developing a new picker model and just using the existing picker as a reference, without much likelihood of focussed or deeper understanding of which change (inputs? architecture? number of layers? etc) lead to which performance changes.

2) If we did develop a new picker architecture which uses CF inputs, we would still need to demonstrate that the new architecture does not perform the same or better with raw seismic inputs. This demonstration would effectively be equivalent to our current study, in which we leverage all the advantages of an existing, well established and widely used picker architecture.

2) PhaseNet with raw seismic signals and CFs

See our response to Reviewer C's Detailed comment 2).

to determine whether incorporating domain knowledge assists the network in reducing its issues. PhaseNet is known to have a high false positive rate when applied to continuous data, which may not be evident in benchmark datasets that lack noise examples.

We did spend some time investigating the original Phase Net and DKPN on continuous data and noted some problems. After this investigation, when we had nearly completed our DKPN study, paper(s) on the problems of PhaseNet with continuous data were published (e.g. Park et al. 2023, also see Park et al. 2024). We decided that including application to continuous data would add excessive time and material to our study, and would require addressing the known but unresolved problems (which most likely will also occur for DKPN, but maybe not, or less) that are not core to our study. In addition, we tried to follow as closely as possible the configuration, training and testing in the original PhaseNet paper, where there is only a toy example of continuous data and the statement that for application to continuous data PhaseNet should be retrained with more non-seismic signals including noise spikes.

Park, Y., Beroza, G. C., & Ellsworth, W. L. (2023). A Mitigation Strategy for the Prediction Inconsistency of Neural Phase Pickers. Seismological Research Letters. https://doi.org/10.1785/0220230003

Park, Y., Delbridge, B.G., Shelly, D.R., 2024. Making Phase-Picking Neural Networks More Consistent and Interpretable. The Seismic Record 4, 72–80. https://doi.org/10.1785/0320230054

Moreover I think your study may benefit from an ablation test. For this you can reduce the number of layers in PhaseNet when applied to CFs (this means the domain knowledge might make the problem easier enough already that now could be solved with fewer number of neurons), test performance using each CF individually (to see which plays a bigger role), etc.

See our response to your point 1) above.

Minor comments:

[From the annotated PDF.]

Page 3 - Substitute the “Non-technical summary” with

Seismic energy onset detection procedures on seismograms are critical for earthquake and environmental monitoring, earthquake and tsunami early-warning, and for fundamental research in seismology and earthquake hazard. High-performance onset detectors mainly use sophisticated algorithms, which are trained on large, unlabeled, seismogram datasets. However, there is a long history of rule-based, automated onset pickers in earthquake seismology that efficiently exploit various characteristics of seismogram waveforms. Here we use classical, seismological phase-based transforms to transform seismogram waveforms before input to an established machine-learning onset detector. We compare this extended detector with the original detector using synthetic learning datasets and application to various test seismograms. We find that the extended detector shows improved performance when applied to seismograms with different characteristics from those used for training, and can allow use of smaller datasets during training. The results show that the established machine-learning detector performs well irrespective of the characteristics of the input data, but that such transformations can improve performance and efficiency in some cases.

Thanks! We incorporated many of your suggestions, but skipped some, mainly to preserve the simplicity of and to repeat words used to replace technical terms.

Page 4 - Introduction - First paragraph (line 3)

“for earthquake and environmental monitoring, earthquake and tsunami early-warning, and for basic research of earthquakes and their hazard”

How about arrival-time-based tomography and subsurface characterization?

Thanks, done.

Page 6 - Line 13

“...wavelet analyses (Anant & Dowla, 1997; Zhang et al., 2003).”

This is another wavelet-based picker: Automatic microseismic denoising and onset detection using the synchrosqueezed continuous wavelet transform

Thanks, added. Quite interesting!

Page 6 - Line 22

“the 3-component characteristic functions of the multi-band FilterPicker, plus the instantaneous modulus”

This also decomposes the 3c time series into multiple time-series, and thus expanding the dimensionality of the input, am I right?

Correct. We have mentioned this increased dimensionality in the Introduction and Discussion.

Page 6 - Line 23

“inclination of the waveforms from particle-motion analysis.”

the filter bank provides a controlled decomposed view of the input data that could make it easier for the network to identify discriminative patterns in the noise, signal, and different phases. But what is the logic behind the use of particle-motions? to help differentiating P from S? if so it may help readers by stating it more clearly here.

We have added more explanation in the section "2.3 Instantaneous modulus and inclination" (lines 268-274).

Page 7 - Line 2

“computing time for training and application”

So this includes the waveform processioning. I guess the main computational costs come from polarization computation, is that right?

Actually the longer pre-processing times mainly arise from the characteristic function calculations and higher level Python array manipulations. The inclination and modulus calculations are relatively fast since more fully implemented array-wise in numpy.

Page 8 - LIne 14

“band-pass filtered seismograms.”

I am just wondering if using DWT will have similar functionality here while perhaps being more efficient!?

The band-pass filtering in FilterPicker is implemented as a bank of recursive, time-domain filters so is very fast and easy to implement for real-time streams. And the whole set of filters is a sample of rows of a spectrogram, so perhaps a "domain knowledge" study using a spectrogram or DWT image as input would be of interest.
The study of Njirjak et al., (2022) explores ML earthquake detection using different time–frequency representations (TFRs) and shows that the choice of TFR leads to significantly different results for several different ML architectures. We added a second reference to this paper in the Introduction.

Njirjak, M., Otović, E., Jozinović, D., Lerga, J., Mauša, G., Michelini, A., & Štajduhar, I. (2022). The Choice of Time–Frequency Representations of Non-Stationary Signals Affects Machine Learning Model Accuracy: A Case Study on Earthquake Detection from LEN-DB Data. Mathematics, 10(6), 965. https://doi.org/10.3390/math10060965.

PAGE 8 - Line 22

“simplified time-frequency, spectrogram representation of the raw waveforms.”

Do you think this would also have some sort of normalization effect and hence could help the training and perhaps the generalization to other dataset?

Yes, good point. With regards to normalisation, the amplitude of the CF's depends primarily on the impulsiveness of an arrival and secondarily on the signal-to-noise level, but not directly on the absolute amplitude of the arrival. And the FilterPicker CF is the maximum over bands. So there is a normalisation, or more specifically a greater sensitivity to quality of an energy onset than to raw amplitude. For this reason we add the modulus waveform channel, to replace the loss of information on absolute amplitude.
These features of the CF's may also aid in generalisation, since amplitude and frequency characteristics of the waveforms, which may vary between datasets and study areas, are suppressed in favour of perhaps more general features of energy onsets on the waveforms.

We have added mention of these points in describing Figure 2 in the text and in the Discussion (lines 733-734).

Reviewer C:

Overview

The authors have presented an interesting experiment by using characteristic functions as the input to a deep neural network. They compared the CF-based models and waveform-based models trained on data sets of different sizes by conducting comprehensive tests. They found that the model trained on characteristic functions can outperform the same network trained on raw waveforms when there are insufficient training data (a few thousand examples). The model trained on characteristic functions also show slightly better performance in cross-domain tests. The workflow is reasonable and the test results are informative. The authors have also discussed the limitations of their method. The insights provided in this paper can be beneficial to future studies of model development.

I have some comments that I would like the authors to address before this paper can be accepted for publication.

Detailed comments

1. Will the conversion from raw waveforms to CFs result in loss of information? If so, will it potentially impair the model performance?

   Yes there must be a loss of information in general, as the CF only captures certain characteristics of the waveforms. Indeed, a premise of the study is that the CF is a logical choice to amplify some of the most important characteristics for phase picking, and a purpose of the study was to investigate the effect of the conversion to CF's on model performance.

   However, we added inclination and modulus channels for the reason of preserving certain information that might be important for phase picking and lost in the characteristic function.

   We spent much time testing various combinations of processed channels and settled on 3 CF's, inclination and modulus.

   We have added mention of this issue in the section "2.3 Instantaneous modulus and inclination" (lines 268-274).

2. Why not use both 3-component waveforms and characteristic functions (8 channels) as input?

   We have run this case and found that running the 8 channels input makes almost no difference in the results (e.g. mean and median F1 scores, see Reviewer Figures R1-3), except for a degradation of results for S with the smallest training dataset

NANO2 for all test datasets. Additionally, the 8 channel input leads to increased spread of the upper/lower limits of the mean for the NANO2 and MICRO training datasets. We suppose that this reduction in performance is due to the raw waveforms adding little or no information relevant to picking over the 5 channels of CF's plus inclination and modulus waveforms. The train-validation loss-curves with 8 channels (Reviewer Figure R4) are similar to those with 5 channels (Supplementary Figure S4).

We added a note of this test in the Discussion (lines 729-736).

Maybe this can overcome some drawbacks of DKPN, such as the stabilization issue?

There would still be the stabilisation issue (needing a certain length of signal before the arrivals) for the CF channels, so this would not be a complete solution. In addition, we do not think the stabilisation issue is important in practice, since picking is typically done on long, continuous traces with sliding windows where the stabilisation only occurs at the beginning of the trace data. This is identically the case for FilterPicker and other STA/LTA pickers with continuous data.

3. Is DKPN applicable to seismograms with only one or two components, for which the modulus and inclination cannot be accurately calculated.

   In the current implementation it can not. We could easily add a code to handle empty/missing channels as zeroes. Though the CFs will be altered.
   In any event, we see no fundamental reason why DKPN is not applicable in this case (there will only be one or two CF's that are non zero and the inclination may not be available) as long as the training data has a statistically representative sample of missing traces for a given target study, the trained network should be applicable, and likely would perform as well as PhaseNet trained on the same sample of traces.

4. Do the parameters of FilterPicker (e.g. long-term window) affect the performance of the trained model? How do you determine these parameters?

   This is an issue carried over from FilterPicker and all STA/LTA type pickers. We basically followed the guidelines from the FilterPicker paper with some trial-and-error over a limited range of typical values for broadband, local and regional event picking. We added a note on this in the manuscript in section 2.2 "Modified FilterPicker characteristic functions" (lines 253-256).

5. You did not include noise traces in the training set (page 16, first paragraph). Can noise traces make a big difference to the training and the final models?

We did not include noise traces primarily because we are following as closely as possible the procedures from the original PhaseNet paper (Zhu and Beroza, 2019). This is already stated in the manuscript in section 2.5 "Seismogram waveform datasets": "INSTANCE includes pure noise waveforms, which, following Zhu & Beroza (2019) we do not use for training."

Also, there are effectively "noise" signals before and after the P and S phases in the event waveforms, if these get picked it adds false positives to the statistics and analysis.

6. Did you include noise traces in the validation set and test set? Since continuous waveforms can contain a large number of noise segments, including noise examples in the test set is necessary for reflecting the true performance statistics of a picker.

   As above, we never used pure noise trace windows in order to follow closely the procedures of Zhu and Beroza, 2019

7. *Page 5, the last sentence of the first paragraph*
   Detections from classical pickers have also been fed into ... →
   Detections from classical pickers can also be fed into ...

   Thanks. Done.

8. *Page 6*
   We train sets of PhaseNet and DKPN models using different size datasets extracted from the INSTANCE waveform collection... → You can consider rephrasing this sentence to make it more straightforward, for example “We train PhaseNet and DKPN on waveforms from the INSTANCE dataset (Michelini et al., 2021). The training is run on 7 subsets with different sizes, leading to a set of model variants”.

   Thanks. Done.

9. *Page 7 Subsection 2.1*
   It is not necessary to introduce too many details here, because most users of the model may not care about the details, while model developers may be familiar with these. For example, the sentence “*Deep-neural-networks transform input data, represented as a layer of nodes, through numerous, simple, non-linear modules into increasingly abstract layers which pre- serve only essential information in the data needed for a target regression or classification task (LeCun et al., 2015)*” is too abstract for beginners and uninformative for experts. It should be sufficient to say “A

trained deep-neural-network can be interpreted as a very high- dimensional approximation function composed of many, local mappings of input to output".

Actually, besides seeming quite interesting and informative to some of us authors, the point "preserve only essential information in the data needed for a target regression or classification task" is a prime motivation for our study. We have reworded some of this subsection to better make this point and tighten up the presentation.

10. *Page 7, subsection 2.1*

"... with each stage implemented with 1-D convolutional and other operators."

What are the "other operators"? Do you mean skip connections and transposed convolution? Please be more specific, or remove this statement.

Changed to: "4 stages of down-sampling and reduction in number of nodes based on 1-D convolution followed by 4 stages of near-symmetric up-sampling and expansion based on 1-D deconvolution"

11. *Page 9, subsection 2.3*

How do you deal with one-component waveforms? If there are not 3 components, then modulus and inclination may be incorrect.

Besides, if only Z component is available, the expression $\sqrt{Z}$ becomes numerically $(N2+E2)$

singular because $N2 + E2 = 0$. Do you just use the limit of $\tan{-1}\ \sqrt{Z} = \pi/2$ when $(N2+E2)$

$N2 + E2$ is near zero?

We always input 3 components. The two horizontal components can be unavailable (zero or constant). See also our response to your comment 3 above.

12. *Page 11*

Figure 2 is too long. Is it possible to put the 4 subfigures into the same row? In addition, com- pared with other figures (e.g. Figures 3-4), Figure 2 is less important. Is it really necessary to show all the four subfigures?

This figure illustrates much for understanding DKPN and PN and is referenced as such in the text, and serves as a solid record for features a reader may wonder about which we do not cite or discuss. This is the only figure in the main paper showing data along with all processed input and output for DKPN and PN. We have reduced the figure panel sizes so all 4 panels fit on one page.

13. Capitalize the words in Figures 1, 4, 7, 9. For example, in Figure 1, “nearly raw 3-component seismograms” → “Nearly raw 3-component seismograms”, “standard PhaseNet output” → “Standard PhaseNet output”.

Done where feasible.

14. *Page 16*
You have constructed 7 training sets in different sizes (NANO3, NANO2, NANO, MICRO, tiny, small, MEDIUM, LARGE), but only results for NANO2, MICRO and MEDIUM are shown. Why not present the results for the other training sets, such as the LARGE?

The primary reasons are for clarity of not showing too many cases, and because the main evolutions and features of picker behaviour with different training sizes can be shown with less than all cases. We do not present or discuss LARGE because in general the pickers appeared fully trained by the MEDIUM training dataset.

We extended an existing sentence explaining this decision in section 2.5 "Seismogram waveform datasets".

15. You can show how the model performance metrics (e.g. the highest F1 score) vary with the size of the training set (refer to Figure 4 in *Lapins, S., Goitom, B., Kendall, J.-M., Werner, M. J., Cashman, K. V., & Hammond, J. O. S. (2021). A little data goes a long way: Automating seismic phase arrival picking at Nabro volcano with transfer learning. Journal of Geophysi- cal Research: Solid Earth, 126, e2021JB021910. https://doi.org/10.1029/2021JB021910*).

This is shown to some degree in Figs 3, 5, 7 and, as our goal is in comparing DKPN with PhaseNet and not illustrating and validating a new picker, we did not see a need for such a comparison in additional figures or discussion.

16. *Page 17, first paragraph*
“For machine learning training in general, an ample length of background before arrivals is also needed for random window-shift, data augmentation to enhance generalization in the trained model, and, most importantly, to avoid that first arrivals are near the same window position in all or most training samples. ”

An ample length of background noise is not a necessary requirement for random window- shift, because you can pad the trace with zeroes at the beginning. In seisbench, this can be done by setting the strategy=‘pad’ in the window classes (see https://github.com/seisbench/seisbench/blob/7119db6aa3a95b1e9a6e4427baaa0e775cc0e1e0/seisbench/ generate/windows.py#L76-L131 and https://github.com/seisbench/pick-benchmark/blob/74ba1965b1dd5e770a8358ed83e339a01460e86b/benchmark/models.py#L472-L476)

This would be a problem for the DKPN CF's, as they would (and should!) respond to the change from zeros to real signals - any padding, including with most synthetic noise, introduces a change in trace statistics and a discontinuity at the join with any real data. It also seems likely that there would be related problems for any ML pickers, unless they are trained on and applied to windows with similar statistical rates of occurrence and length of preceding zeros. But even then, would not the ML picker necessarily be learning something about the padding and become less sensitive to certain forms of true phase onset? For ML, real noise from each channel may work, but this may not work for DKPN CF's, as there is still a discontinuity for some orders. Interesting!

17. *Page 17, first paragraph*

"Lack of sufficient background data before arrivals can impair classical methods like STA/LTA in comparisons with machine-learning pickers."

Can it be addressed by padding zeros or random noise at the beginning?

STA/LTA methods are building a statistical representation of the true noise from the preceding background. This representation clearly cannot be obtained from zeros, and probably not from random noise unless this noise somehow matches very closely the true noise from the point of view of the STA/LTA processing algorithm. See also our response to your comment 16 just above.

18. *Page 17, last sentence in the first paragraph* "... and very little may be available after."

What is the meaning of this sentence?

Changed to: "very little data may be available after an arrival onset before reaching the last received data"

19. *Page 17, the first sentence of the second paragraph*

"We present a set of tests with different evaluation datasets to illustrate and compare the performance of PhaseNet and DKPN."

It could be more concise, such as "compare the performances of PhaseNet and DKPN on different test sets". In addition, this sentence seems not relevant to Section 2.6 "Dataset and model configuration, processing, training and comparison".

Thanks. Modified.

20. *Page 17*

"We configure and preprocess datasets and models, train the models and compare PhaseNet and DKPN P and S arrival predictions, statistics and metrics using open Python codes we developed for this study (see Data and code availability)". The statement "using open Python codes we developed for this study" can be removed here because it has been mentioned in the section of "Data and code availability".

Thanks. Modified.

21. *Page 17*

"this length should be greater than the number of sample points (*NFPS*)" What does the sub- script NPS represent?

FilterPicker stabilization (FPS). Added this definition.

22. *Page 17*

"3. ... 50% training, 5% validation and 45% remainder for drawing test samples)."

First, it seems that you did not use all the 45% of waveforms for test. Instead, you randomly choose subsets consisting 5000 samples, right? You have 300,000 waveforms in total. 45% represents 135,000 testing samples. 7 subsets, each consisting of 5000 samples, include 35,000 sample. Then there are 100,000 samples that have not been used. Why not use all the 135,000 testing samples in your tests? Are 5000 samples enough to reflect the performances of the models?

We saw no evidence that 5000 samples are insufficient to develop representative statistics of the test performance. More samples greatly increases the overall time needed to perform all tests.

Second, what do you use the validation set for? It seems that it is not used in the rest of the paper.

We use the validation set during the training stages, to avoid overfitting and defining the criteria of early stopping. Train-validation loss-curves are shown in File S4 in the supplementary material.

We added explicit mention of use of the training and validation datasets in section 2.6 "Dataset and model configuration, processing, training and comparison" (lines 420-436)

Third, have you included any noise traces in the test set?

No, we do not use noise traces in any part of this study, as mentioned in replies to previous comments

23. *Page 18, the training workflow*

What are your hyperparameters, e.g. learning rate, batch size?

Parameter values are reported in supplementary material, Table S1. After an initial, educated guess and some try and error testing, we settled with a batch-size of 64 and a learning rate of 0.001. The early-stop criteria was responsible for the number of epochs adopted.

Have you tuned the hyperparameters?

We performed limited, trial-and error exploration of different values.

24. *Page 18*

"... where the loss must decrease by at least a fixed improvement value"

Do you mean the training loss or the validation loss?

Corrected.

25. *Train models using an early-stopping approach defined by a "patience" number of epochs ...*

avoiding overfitting issues that may occur e.g. if using a fixed number of epoch

Using a fixed number of epochs does not necessarily lead to overfitting. When using a fixed number of epoch, you can save the epoch with the lowest validation loss, instead of the
last epoch. You can also try the ReduceLROnPlateau learning rate scheduler which will decrease the learning rate when the loss has stopped improving (https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html#torch.optim.lr_scheduler.ReduceLROnPlateau).

Thank you, we have removed that phrase. We agree with the comment that overfitting with a fixed number of epochs may not occur. Though, we still thinks that our early-stop criteria and the usage of the Adam optimizer (that autonomously adapt the learning rate) definitely allow for a smoother and stable convergence and should be preferred, without any prior bias based on a fixed value (i.e. number of training epochs).

26. *Page 19*

"a 3-point smoothing to suppress rapid oscillation"

Why is this step necessary? In other words, what would happen if you remove this step? Will it decrease the peak probability value? I am interested because we usually do not smooth the probability curves before extracting picks. Also, this operation has not been implemented in Seisbench.

We implemented the smoothing because we use scipy.signal.find_peaks to identify picks. We then use the peak location and scipy.signal.peak_widths to get a width for the peak pulse in order to construct a gaussian pick (pick time plus uncertainty).

If the probability function has oscillations, then there can be biases in the peak position relative to the visual centre of the probability pulse, especially for smaller training sets and fewer epochs. Oscillations can also bias the peak width estimates. The oscillations can be seen in Figure 9, particularly for PhaseNet (up to epoch 10) which converges slower than DKPN with epoch.

If we do not smooth, then the peak value will in general increase, as the smoothing removes high-frequency spikes.

27. *Page 19*

"b. pick detection at peaks of amplitude greater than specified thresholds and separated by more than 0.5 sec"

Why do you require picks to be separated by more than 0.5 sec? Do you mean any picks (including P and S) should be separated by more than 0.5 sec, or picks of the same phase should be separated by more than 0.5 sec?

With 100 Hz sampling and local/regional seismograms, for the purpose of earthquake location, we considered that picks closer than 0.5 sec would not provide useful information and could be indicators of a double peak or oscillation in a single peak pulse.

The processing is done on each probability channel, i.e. P and S are processed separately. So only picks of the same phase are required to be separated by > 0.5 sec.

28. *Page 19*

"In this study we repeat the training workflow using 7 different random numbers to extract different training and validation subsets of traces for each experiment. "

What do you mean by "7 different random numbers"? You may need to rewrite this sentence to make it more clear.

Changed to "In this study we repeat the training workflow using 7 different, randomly selected training and validation subsets of traces for each experiment."

29. Do you evaluate the 7 models on different test sets? If so, is it appropriate to directly compare the test results evaluated on different test sets? I think it may be more reasonable to use the same test set consistently.

Interesting question. We considered that our goal is to broaden the evaluation as much as possible, and using different models and different test sets would seem to do this more than using a fixed test set. Our goal is not to compare the specific trained models, but to compare the overlying ML architectures (DKPN and PN) that produce the trained models. Our perspective is that as long as the test datasets feature similar statistics and encompass similar distributions then the difference in test datasets should be irrelevant, or an advantage in case of occasional, atypical traces. We are confident that this is the case for our set of 7 different test datasets.

30. *Page 19*

"... the number for P or S of correct Gaussian predicted arrivals (true positives; TP), incorrect predicted arrivals (false positives; FP), and incorrect prediction of no arrivals (false negatives; FN)."

If the model predicts 100 picks on a seismogram, one of which is close enough to the ground truth. Do you count this example as a true positive or a false positive? Maybe it is better to treat this prediction as 1 true positive and 99 false positives?

We indeed count such an example as 1 TP and 99 FP.

Similarly, if the model give you 100 picks on a seismogram, none of which is close to the labeled pick. Do you regard this example as a false positive (because there are false picks) or false negative (because the labeled pick is overlooked by the model)? If you consider it as a false negative, the false picks are ignored. On the other hand, if you consider this prediction as a false positive, you treat it the same as the prediction on a waveform with only one false pick, which is unfair.

We count it as a FN, but also keep the remaining picks as false-positives (FP) instead. Overall, for each channel separately (P/S), we do the following: for each reference pick in the window, we seek a possible match (true positive, TP) among the predicted picks if the time difference is below a certain threshold (0.1 sec and 0.2 sec for P and S respectively). If a match is found we account for a TP and if there aren't possible matches, we account this as false negative (FN). Even if there's a TP, we account the rest of predicted picks as FPs.
We clarified the text and added "There may be multiple FP picks for P or for S on each data window." (line 455)

By taking into account every predicted pick, we avoid doing statistics using a simplistic approach assuming a single-reference pick per window / single predicted picks per window for each channel. We expect this makes our statistics more robust.

31. *Page 19*

"we examine a range of thresholds $0.1 \leq Ap \leq 0.9$ to find optimal metrics such as F1 score ..."

Do you search for the optimal threshold based on the validation set or the test set? In my opinion, the validation set is used to hyperparameters, and the threshold can also be considered as a hyperparameter.

We indeed do not seek for a precise threshold-parameter tuning, rather we show how it can affect the F1/precision/recall scores. As specified in the text, we use the outcome from our test-dataset for our figures.

32. *Page 19*

"Additionally, filtering of false picks can be done in phase association and hypocenter loca- tion processing stages"

That's true, but too many false picks may have an adverse effect on phase association.

That is also true. There is a trade-off between too many picks and missing picks. We do not attempt to investigate or resolve this issue in this study.
Changed to "... filtering of a limited number of false picks can …"

33. Please use higher-resolution figures.
Meaning? We should provide PDFs figures…

We increased the resolution of the testing graphs Fig 3 etc and testing histograms Fig 4 etc. Other figures were generated at high resolution.

34. *Page 20-21, Figure 3* Avoid using underscores for clarity. Fore xample,"P_f1"can be replaced with"F1 for P picking". In the legend,"DKPN_P_f1_mean"can be replaced with"Mean F1 for DKPN",where P can be omitted because it has been indicated in the label of the y-axis.

In addition, capitalize the labels, “threshold” → “Threshold”.

35. *Page 20-21, Figure 3*
It is a bit difficult to tell the difference between the F1-score curves of DKPN and those of PN. Maybe it would look more straightforward if you put the F1 scores of PN and DKPN in the same figure.

You can show the value of the highest F1 in the legends, such as "DKPN_f1_mean (0.90)".
I remember we did that, but went back to the separate plots outline. Let me know if I need to adjust it

We tried that, but with 4 curves of 3 different types for each method the plot became more complicated and the main information we wanted to convey was obscured.

36. *Page 22, Figure 4*
Did you test the three models on different test sets, for the total number of available picks is different for each example? I suggest fixing the test set to compare different models.

See our reply to your comment 29 above. Our goal is not to compare the specific trained models, but to compare the overlying ML architectures (DKPN and PN) that produce the trained model.

37. *Page 22, Figure 4*
What does “thr” mean in the titles of the subfigures? Why is “thr” larger for P than for S? In the test of INSTANCE-MICRO (the second row), for the P residual (the left column), “PN thr: 0.3” is different from the other tests where “PN thr: 0.4”, why?

"thr" in titles means threshold.
As stated in the captions "Results shown for the pick amplitude threshold giving the highest F1 score for each dataset size." We have clarified this to define "thr" and to note that the highest F1 score is per dataset size and per method (DKPN or PhaseNet).

38. *Page 22, Figure 4*
Capitalize the labels. "count" → "Count", "residuals" → "Residuals", "mean" → "Mean", "std" → "STD".

Done.

39. *Page 22, Figure 4*
You can also calculate the mean absolute error and median absolute deviation, which are less sensitive to outliers (Figure 2 in Mu¨nchmeyer et al., 2022; Figure 3 in Bornstein et al., 2024).

Mu¨nchmeyer, 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.

Bornstein, T., Lange, D., Mu¨nchmeyer, J., Woollam, J., Rietbrock, A., Barcheck, G., et al. (2024). PickBlue: Seismic phase picking for ocean bottom seismometers with deep learning. Earth and Space Science, 11, e2023EA003332.

The main differences to the eye in the histograms are an overall residuals shift between PN and DKPN (e.g. INSTANCE NANO2 P and MICROS in Fig 4; PNW P and NANO S in Fig 8) and only rarely a difference in spread (error). Mean and median would capture this shift while MAE and MAD would not, they capture spread. We use trimming to avoid outliers.

40. *Page 23, lines 3-5*
"To remove outlier data, the mean and standard-deviation statistics use trimmed residuals, within twice ΔTp : ±0.2 sec for P and ±0.4 sec for S."

The interval ±0.2 or ±0.4 seems too small. In phase association, we usually use a larger threshold (e.g. 2 seconds) to account for inaccuracies of velocity models or picks. It means that picks with residuals larger than 0.4 seconds but smaller than 2 seconds can also be associated. Hence it is important to know the statistics of the residuals in a larger interval. You can just follow (Mu¨nchmeyer et al., 2022; Bornstein et al., 2024), setting the range of trimmed residuals as ±1s or ±1.5s

These intervals are used in statistical comparison of two ***observed*** values: ML picks and manual picks. It is important that the differences between these picks is smaller than the typical final residuals in earthquake location after application of station correction procedures or 3D velocity model inversion. For local/regional high-precision locations, these residuals are typically much less than 0.5 sec and often less than 0.1 sec.

We are not doing a statistical analysis of ***observed*** ML pick times relative to ***predicted*** times that depend on velocity and other models during phase association, location or other procedures. In the latter case, we agree that larger differences should be included, especially before station corrections or 3D models have been developed.

We also think our choice of intervals is also justified by the fact that, for our histograms, ±1s would be way off the limits of the plots, thus it seems the results do not demand or justify such large limits.

41. The sections 3.2 and 3.3 have very similar section titles.

Right! Thanks. Changed.

42. Section 3.3 is quite similar to section 3.2. Would it be better if you move most of section 3.3 to the supplement and just summarize the new information from the cross-domain test on PNW in the main text?

There is not much text in either section, while the corresponding figures, 5 and 7, 6 and 8, which take most space, are notably different. So it seems best to keep them all parallel and in the same place.

43. *Page 30, the first sentence of the second paragraph*
I suggest using numbers to describe the improvement, like "... show an improvement in F1 of [number]".

Thanks, done.

44. *Page 32, the first sentence of the third paragraph*
"... (Figure 3, 5, 7), DKPN seems to be more stable than PhaseNet in P performances across many threshold levels."

The stability of P phase picking with respect to threshold seems similar for the two models. For S picking, DKPN seems slightly more stable.

This stability refers to the dashed curve showing the upper and lower limits of F1 curves obtained at each threshold with different random realisations of datasets. These curves are clearly less spread for DKPN than PN for NANO2. We have

clarified in the text that "stable" means less as indicated by the spread of upper/lower limits of the mean (dashed curves).

Again, it is hard to compare the curves for DKPN and the curves for PN by eye. You can consider plotting them in the same window.

See our response above to your comment 35.

45. *Page 33, the first sentence of the last paragraph*
"... content into abrupt, step- or pulse-like waveforms (Figures 4)"

It seems that you are referring to Figure 1 instead of Figure 4?

Thanks, should be Figure 2.

46. *Page 33*
"This result may be related to the PNW dataset, relative to INSTANCE and ETH, having a large number of clear, impulsive S arrivals, which may match well impulsive P arrivals for which classical pickers such as Filter Picker are optimized, and thus more easily detected by DKPN."

To support your assumption, have you tried training on PNW and test on INSTANCE to see whether DKPN still shows a better performance than PN?

We add this point in the discussion as a conjecture (and not an assumption) and do not test it because, besides time limitations, it is not central to PhaseNet-DKPN comparison. But we think it is an interesting point for thought and potential future work.

47. Turn on line numbering to facilitate reviewing
(https://www.youtube.com/watch?v=0rS126JLbcI)

Sorry, we agree, this was an oversight at submission.