Shear Wave Splitting Analysis using Deep Learning (SWSNet)

Megha Chakraborty[1,2], Georg Rümpker[1,2], Wei Li[1], Johannes Faber[1,3], Frederik Link[2], Nishtha Srivastava[1,2]

[1]Frankfurt Institute for Advanced Studies, 60438 Frankfurt, Germany
[2]Institute of Geosciences, Goethe University Frankfurt, 60438 Frankfurt, Germany
[3]Institute for Theoretical Physics, Goethe University Frankfurt, 60438 Frankfurt, Germany

Key Points:

- We present a novel deep learning-based approach to identify splitting parameters from waveforms, which is faster than grid-search.
- The model is trained on synthetic data which is made to mimic real data using a series of deconvolution operations.
- When tested on real waveforms from the USArray dataset, a seismic anisotropy pattern comparable to previous studies is obtained.

Corresponding author: Georg Rümpker, `rumpker@geophysik.uni-frankfurt.de`

Abstract

Teleseismic shear-wave splitting analyses are typically performed by reversing the splitting process through the application of frequency- or time-domain operations minimizing transverse-component waveforms. These operations yield two splitting parameters, ϕ (fast-axis orientation) and δt (delay time). In this study, we investigate the applicability of a recurrent neural network, SWSNet, for determining the splitting parameters from pre-selected waveform windows. Due to the scarcity of sufficiently labelled real waveform data, we generate our own synthetic training dataset. The model is capable of determining ϕ and δt with a root mean squared error (RMSE) of 9.58° and 0.143 s for noisy synthetic test data. The application to real data involves a deconvolution step to homogenize the waveforms. When applied to data from the USArray dataset, the results exhibit similar patterns to those found in previous studies with mean absolute differences of 11.12∘ and 0.25 s in the calculation of ϕ and δt, respectively.

Plain Language Summary

In this study, we explore the use of a deep learning model called SWSNet to analyze seismic wave data. This method helps determine the properties of the Earth's mantle that affect how seismic waves travel. Typically, researchers use complex calculations to analyze seismic data, but our approach uses a deep learning model trained to recognize patterns in the data. Since there is not enough labelled data available for training, we create our own synthetic data for this purpose. Our model can accurately determine important characteristics of the subsurface layer, and when applied to real-world data, it produces results similar to previous studies. This work shows that SWSNet is a promising tool for analyzing seismic data and understanding Earth's interior.

1 Introduction

The analysis of seismic anisotropy serves as a unique tool for investigating the elusive dynamic processes occurring within the Earth's mantle. Inferring vertically and laterally varying anisotropic structures from surface-recorded seismic waveforms can provide vital constraints for geodynamic models of mantle deformation and flow. The study of teleseismic shear-wave splitting, a technique in use for over three decades, provides key insights about seismic anisotropy, aiding in the analysis of the dynamic processes within Earth's interior (Long & Silver, 2009; Reiss & Rümpker, 2017; Savage, 1999; Silver & Chan, 1991).

Two primary mechanisms contribute to the development of seismic anisotropy in the Earth's mantle: strain-induced lattice preferred orientation (LPO) of upper mantle minerals such as olivine (resulting from differential motion between the lithosphere and asthenosphere, and mantle flow) (Silver & Chan, 1991) and shape preferred orientation due to the presence of vertically aligned fluid-filled fractures, cracks, and microcracks (Holtzman & Kendall, 2010).

When a shear wave enters an anisotropic medium, it is split into two orthogonally polarized components that propagate at different speeds. This phenomenon can be described by two splitting parameters: the fast axis orientation (the polarization direction of the faster wave) ϕ, and the time delay between the two components δt. While ϕ represents the orientation of the anisotropic materials, δt measures the strength of anisotropy and the extent of the anisotropic material. Teleseismic phases are typically employed to investigate the anisotropic properties of the Earth. The most frequently used phases include SKS, SKKS, and PKS, and are collectively referred to as XKS phases. The conversion of these waves at the core-mantle boundary results in polarization in the direction of the back-azimuth (Jia et al., 2021; Liu & Gao, 2013; Reiss & Rümpker, 2017).

Several software codes have been developed to determine the splitting parameters ϕ and δt through grid search or correlation approaches. Examples of such codes can be found in the works of Silver and Chan (1991) such as Liu and Gao (2013); Savage et al. (2010); Teanby et al. (2004); Wüstefeld et al. (2008); Wuestefeld et al. (2010). (Semi-)automatic approaches were recently suggested by Reiss and Rümpker (2017) and Link et al. (2022).

In this paper, we present a novel Deep Learning-based approach for the analysis of shear-wave splitting. In a recent study, Zhang and Gao (2022) utilized an Convolutional Neural Network (CNN) for waveform classification to automatically select reliable SWS measurements. However, a comprehensive analysis to infer anisotropic splitting parameters has not yet been presented. Here, we introduce a Neural Network called SWS-Net (Shear-Wave Splitting Network) to determine the splitting parameters from pre-selected waveform windows. Due to the lack of sufficient labelled data, the model is trained on synthetic data, simulated under the assumption of a single anisotropic layer. A series of deconvolution and reconvolution steps are applied to real data to ensure maximum resemblance. We demonstrate that SWSNet can produce results comparable to those of previous studies such as Liu et al. (2014) when applied to real data from the USArray and obtain mean absolute differences of 11.12° and 0.25 s in the calculation of ϕ and δt, respectively.

2 Methods and Results

For our study we use a supervised learning approach, which is a machine learning paradigm that relies on labelled data for training a model. The Deep Learning model we use learns to map the waveforms to the corresponding labels (in our case ϕ and δt) by minimising the difference between the true and predicted labels defined by the loss function.

In principle, labelled waveform data from shear-wave splitting analyses is available from publications and data archives (see, e.g., Barruol et al. (2009)). However, for our purposes, the amount of available data is limited, and the labelling may not be as uniform, as it would be required for efficient training. In order to overcome this limitation, we will use synthetic data as an alternative. Ideally, the generated synthetic waveforms will mimic the properties and characteristics of real data.

2.1 Modeling shear-wave splitting

In our approach, we consider waveform effects due to a single anisotropic layer, which is characterized by a horizontal symmetry axis (referred to as the "fast axis" and oriented at an angle ϕ with respect to North). A vertically incident shear wave, splits into horizontally polarized fast and slow components, where the fast component aligns parallel to the symmetry axis, while the slow component is oriented perpendicular to it. Generally, these two quasi-shear waves propagate at different speeds, resulting in a separation by the delay time, δt, as they travel through the layer. A graphical representation of the coordinate systems used is given in Figure S1.

The equations to describe shear-wave splitting in layered structures have recently been summarized by Rümpker et al. (2023). In the frequency domain, the radial and transverse displacement components, after passing through the layer, can be expressed as

$$\begin{pmatrix} u_1^{(r)} \\ u_1^{(t)} \end{pmatrix} = \begin{pmatrix} \cos\theta + i\sin\theta\cos 2\alpha & i\sin\theta\sin 2\alpha \\ i\sin\theta\sin 2\alpha & \cos\theta - i\sin\theta\cos 2\alpha \end{pmatrix} \begin{pmatrix} u_0^{(r)} \\ u_0^{(t)} \end{pmatrix} \tag{1}$$

where $\theta = \omega\delta t/2$, $\alpha = \beta - \phi$ is the angular difference between back-azimuth and fast axis, and index 0 denotes waveforms before passing through the anisotropic layer. For XKS phases in a radially symmetric Earth, we can assume that $u_0^{(t)} = 0$ upon enter-

ing the (first) anisotropic layer on the receiver-side leg of the ray path, such that

$$u_1^{(r)} = (\cos\theta + i\sin\theta\cos 2\alpha)\, u_0^{(r)} \tag{2}$$

$$u_1^{(t)} = i\sin\theta\sin 2\alpha\, u_0^{(r)} \tag{3}$$

Note, that for relatively long periods, $T \gg \delta t$ (to first order in θ), this simplifies to

$$u_1^{(r)} \simeq (1 + i\omega\frac{\delta t}{2}\cos 2\alpha)\, u_0^{(r)} \tag{4}$$

$$u_1^{(t)} \simeq i\omega\frac{\delta t}{2}\sin 2\alpha\, u_0^{(r)} \tag{5}$$

where the factor $i\omega$ yields a derivative of the radial-component waveform and the amplitude is modulated by $\sin 2\alpha$. We will use this formulation in the development of our deconvolution approach, as described below.

2.2 Neural Network Analysis - Synthetic Data

We use synthetic data to train our model. The radial and transverse waveforms are generated with a sampling frequency of 50Hz for back-azimuths between $0-360°$ and fast axis ϕ ranging between $0-180°$. Consequently α can vary between $0-180°$, since ϕ and $\phi+180°$ represent the same fast axis orientation. Possible values for δt are between 0.2-2.0 seconds. Note that δt characterizes the anisotropy within the layer and is not equal to an "apparent" delay time which could be much larger (e.g. Silver and Savage, 1994).

Combinations of δt and ϕ are chosen from uniform random distributions for the ranges described above. We experiment with Convolutional layers (Kiranyaz et al., 2015), Bi-directional Long Short-Term Memory (Bi-LSTM) (Hochreiter & Schmidhuber, 1997) layers and a combination of both. The model hyperparameters are chosen by experimenting to maximise the model performance on validation data. Each 1D convolutional layer used has a Rectified Linear Unit (ReLU) activation function (Agarap, 2018). The model outputs three values corresponding to the probability of the measurement being non-null and the normalised predictions for δt and ϕ. Here, any measurement with $\alpha < 2$, $88 < \alpha < 92$ and $\alpha > 178$ is considered a null measurement. A ReLU activation function is used for layers predicting α and δt while a sigmoid function is used to output the probability corresponding the measurement being non-null. A schematic example of such an architecture is shown in Figure 1.

The model is trained using the Adam Optimiser (Kingma & Ba, 2014). Mean squared error and binary cross-entropy are used as loss functions for regression and classification respectively. Apart from using Maxpooling layers in the model architecture, early stopping (Prechelt, 2012) is used to further prevent overfitting, whereby training stops if validation loss does not decrease for 8 consecutive epochs. We find a Convolutional Neural Network to be working best on this dataset.

2.2.1 Results - Synthetic Data

We train the Neural Network on two types of dataset– one is noise-free and the other has 10-30% noise applied independently to the fast and slow components. Some examples for these datasets can be found in Figures S2 and S3 in the Supplementary Information. Figure 2 shows the results when these models are applied to a noisy synthetic test data. As can be seen from Figure 2, the Neural Network has RMSE 62.46° and 0.59 s in the predictions of α and δt respectively when trained on noise-free synthetic data. However when trained on noisy data, the RMSE in the prediction of α and δt decrease to 9.47° and 0.06 s thus demonstrating that the training of the neural network on noise-free synthetic data fails to appropriately mitigate the effects of noise. Since noise is bound

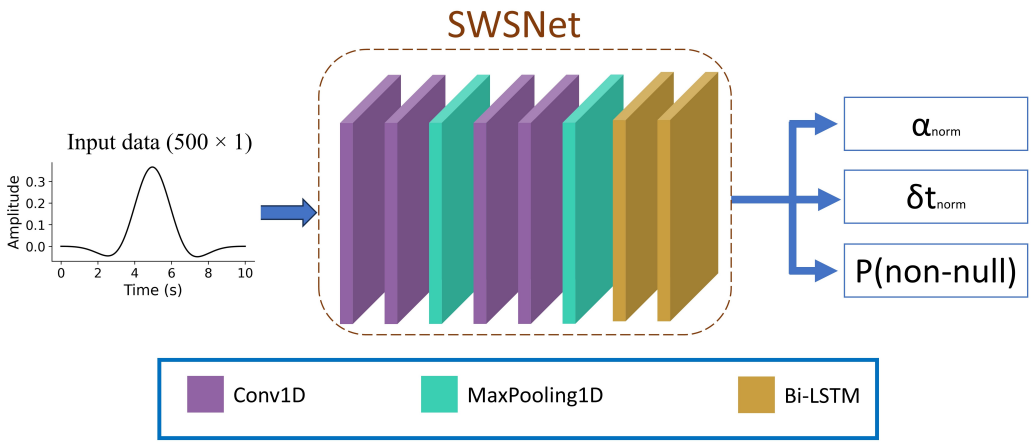


Figure 1. The architecture of SWSNet. The model takes as input the (deconvolved) transverse component and comprises of two blocks of 1D convolution and Maxpooling operations, followed by two bi-directional LSTM layers. The final outputs are the normalised values of α (α_{norm}) and δt (δt_{norm}) and the probability of the measurement being non-null.

to be present in the real data, we train our model on noisy data for the subsequent analysis. The results of this analysis on the noise-free test data are comparable and can be seen in Figure S4 in the Supporting Information.

2.3 Application to real data

2.3.1 Direct application of the Neural Network

When the Neural Network trained on the synthetic data is directly applied to the real data (radial and transverse components), it performs unsatisfactorily when predicting δt. A plot between the true and predicted δt in this case can be seen in the Supplementary Information (Figure S5). This happens as real waveforms look significantly different from the synthetic data. Thus a direct application of the trained Neural Network to the real waveforms renders unusable results. This necessitates an intermediate step to bridge the gap between the synthetic and real waveforms.

2.3.2 Deconvolution approach

Observed real waveforms are not only affected by anisotropic layering but may vary significantly due to different source mechanisms (and path effects). This poses a challenge to the training of the deep learning model, as it becomes impossible to include all waveform variations that may arise from different source mechanisms and complexities of the medium. Here, we choose a deconvolution approach to mitigate source effects and "homogenize" the waveforms. This method is similar to the one utilized in receiver-function processing, for instance Langston (1979); Owens et al. (1984); Ammon (1991).

We deconvolve both the radial and transverse component by the radial component. In the frequency-domain, in view of eq. (5), the procedure applied to real data can be described as follows:

$$u_*^{(r)} = u_1^{(r)}/u_1^{(r)} = 1 \tag{6}$$

$$u_*^{(t)} = i\omega\frac{\delta t}{2}\sin 2\alpha\, u_0^{(r)}/u_1^{(r)} \simeq i\omega\frac{\delta t}{2}\sin 2\alpha \tag{7}$$

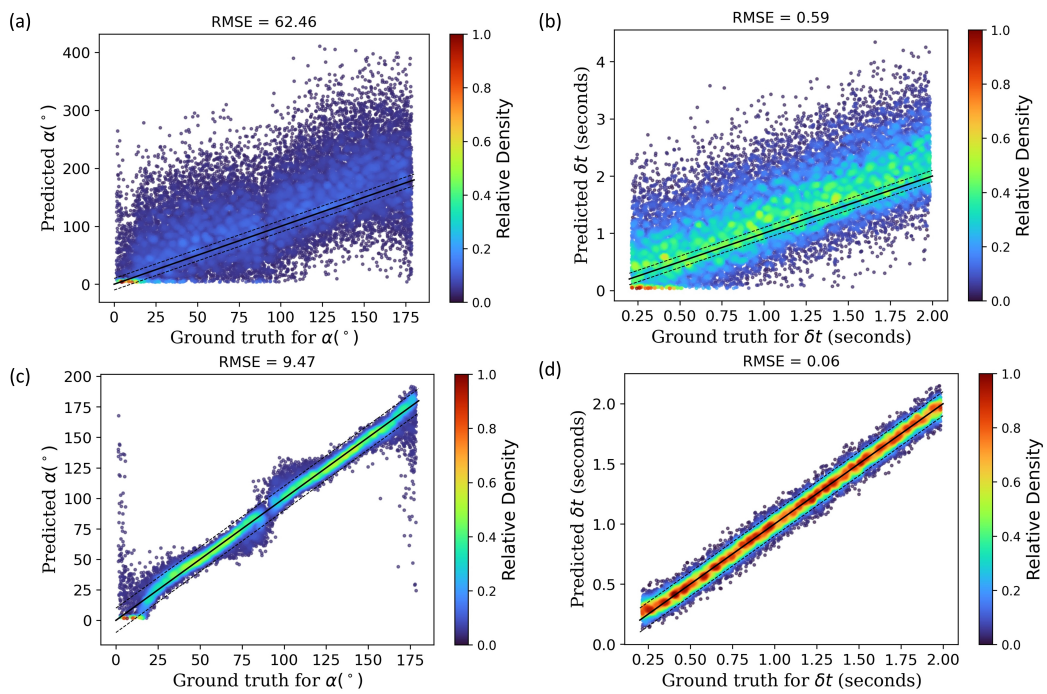


Figure 2. (a)-(b) Ground truth vs predicted values of α and δt for model trained on noise-free synthetic data. (c)-(d) Ground truth vs predicted values of α and δt for model trained on noisy synthetic data. In both cases the input data used for testing is afflicted with noise.

Note that we assumed $u_0^{(r)}/u_1^{(r)} \simeq 1$ in the derivation of eq. (7). This implies that the radial-component waveform is a sufficient representation of the incoming waveform (before it enters the anisotropic layer), which further limits the applicability to waveforms of relatively long periods. The value of 1 for the radial component in the frequency domain corresponds to a δ-function in the time domain. For the transverse component, the factor $i\omega$ causes a time-domain derivative (of the unsplit waveform) with amplitude modulated by $\sin 2\alpha$. In a second step, the deconvolved components can now be convolved with a reference waveform, such as the normalised derivative of an exponential function (Figure S6, radial component as shown in Figure 3), to yield a standard radial component, and uniform transverse component that depends on the two splitting parameters. Figure S7 shows the appearance of the transverse component for different α and δt pairs.

For real data, first the waveform within the selected time windows are resampled at 50Hz and then the mean is removed. For both the real and synthetic data the following steps are applied:

- A Hanning window is applied to smoothen the transition to zero amplitude at the boundaries of the time window.
- The data is zero padded to have a uniform total of 2000 time samples corresponding to a 40 s time window.
- A butterworth lowpass filter with corner frequency of 1 Hz is applied to suppress higher-frequency noise.
- The radial component is deconvolved from both the radial and transverse components as per equations 6 and 7.

Table 1. A comparison between splitting parameters for individual waveforms shown in Figure 3, calculated by Link et al. (2022) and SWSNet. A more detailed comparison between grid search results, results from Link et al. (2022) and SWSNet can be found in table S1 in the Supplementary Information.

Event ID	$\phi(^\circ)$ (Link et al., 2022)	$\phi(^\circ)$ (SWSNet)	δt (s) (Link et al., 2022)	δt (s) (SWSNet)
U14A2007-07-27T15:10:07SKS	48.00	50.64	1.44	1.62
G27K2019-03-10T08:33:53SKS	83.00	84.95	1.03	0.90
J48A2013-04-09T12:17:01SKS	75.00	72.89	0.82	0.90
E59A2015-05-30T11:45:14SKS	82.00	84.99	1.13	1.10

- The clean waveform shown in Figure S7 (radial component in Figure 3) is convolved with both the deconvolved waveforms (radial and transverse components).
- A Hanning window is applied to reduce the effect of possible sinusoidal "ringing" on the transverse component of the reconvolved data.
- The waveform is cropped to the central 10 seconds.
- Another Hanning window is applied followed by the normalisation of the data such that the absolute maximum amplitude in the transverse component is 1.

With this approach it is only the transverse component that carries meaningful information about the splitting parameters. Therefore we retrain our model on the transverse component of the de/reconvolved synthetic waveforms. Once again we experiment with different model architectures find out that the combination of Convolutional and Bi-LSTM layers, as shown in Figure 1, works best. This model will henceforth be called the SWSNet. A detailed description of the hyperparameters used can be seen in Figure S8. As the input data structure is relatively simple a deeper network does not improve the results and simpler network is sufficient. Please note that the labels corresponding to α and δt are always scaled to be in the range 0-1 as this is known to benefit learning. A training data size of 50,000 waveforms is experimentally found to be optimum (Figure S9). The performance of SWSNet on a synthetic test dataset is summarised in Figure S10.

2.4 Application to USArray

We apply our model to pre-selected waveforms from the USArray dataset and compare our results with Liu et al. (2014) and those calculated by the automatic Splitracer toolbox proposed by Link et al. (2022). To make sure that only meaningful results are used in the calculation of station averages we perform a quality check on the estimations made by the neural network on given waveforms. We perform splitting inversion using the splitting parameters predicted by the neural network and check the percentage reduction in the transverse component energy (sum of squared amplitudes) as proposed by Silver and Chan (1991). An experimentally chosen threshold of 60% reduction in transverse component of energy is used to select the waveforms to be used for calculating station averages. Some examples are shown in Figure 3. It is observed that the performance of SWSNet is comparable to that of SplitRacer. Figure 4 shows a visual representation of the station-averages of the splitting parameters calculated by SWSNet and Liu et al. (2014). Unlike Link et al. (2022), Liu et al. (2014) does not employ a joint splitting analysis, allowing for a more direct comparison with our approach, as it is also based on averaging results from individual split phases at a given station. The corresponding comparison between SWSNet and Link et al. (2022) can be seen in Figure S11.

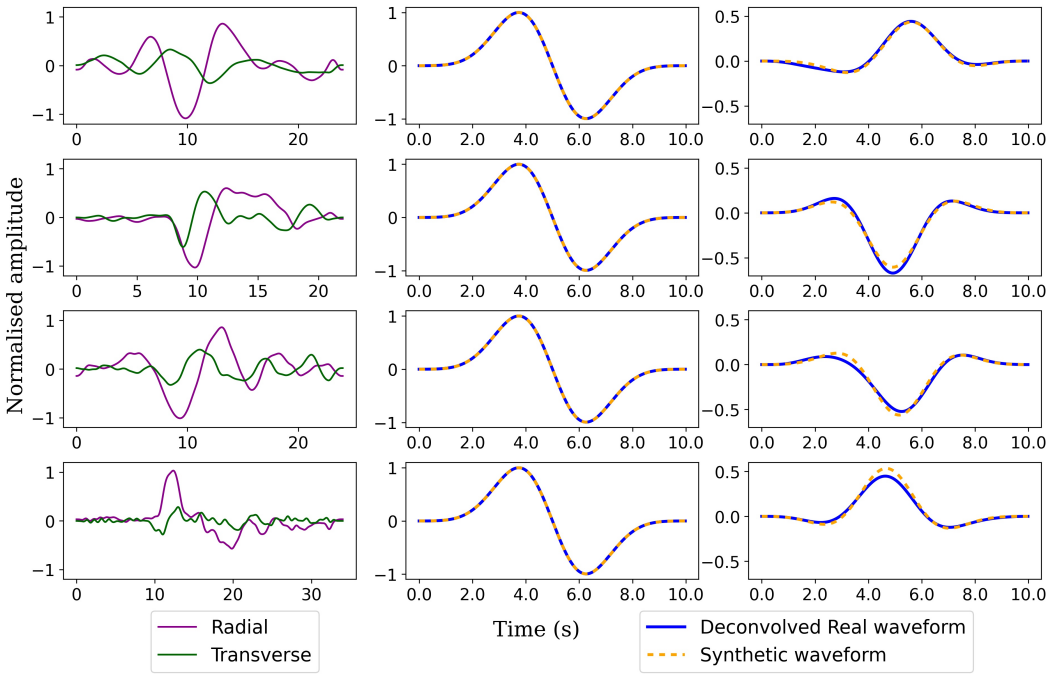


Figure 3. Examples of application of SWSNet to deconvolved real waveforms from USArray Dataset. The middle and rightmost panels show a comparison between the deconvolved real waveforms shown on the corresponding left panel and synthetic transverse components simulated using the splitting parameters predicted by SWSNet. The middle panel shows the radial component while the right panel shows the transverse components. The similarity between the transverse components is observable. The corresponding splitting parameters can be found in Table 1.

3 Discussion and Conclusion

We explore the different factors that affect the station-averaged results, and find that the predominant factor is the number of acceptable measurements for a given station, whereby the difference between the station averaged splitting parameters calculated by SWSNet and those from Liu et al. (2014) diminishes with an increased number of acceptable measurements corresponding to a station (Figure S12).

For the deconvolution approach, as applied to real data, we find errors to be much lower when the neural network is trained and tested on noise-free synthetic data as compared when it is trained and tested on noisy synthetic data. This makes sense, as the deconvolution homogenizes the waveforms, even if noise is present. However, there is a certain degree of discrepancy between the ground truth and the predicted splitting parameters for individual waveforms.

We also compare our method against a simple grid search algorithm that, like previous studies, finds the splitting parameters for which (upon waveform inversion) the energy in the transverse component is the lowest. We plot the energy distributions for different combinations of α and δt for five randomly chosen events from five different stations, and find the parameters to calculated by SWSNet to be quite close to those found by grid search and what is calculated by Link et al. (2022) (Figure S13). We further ob-

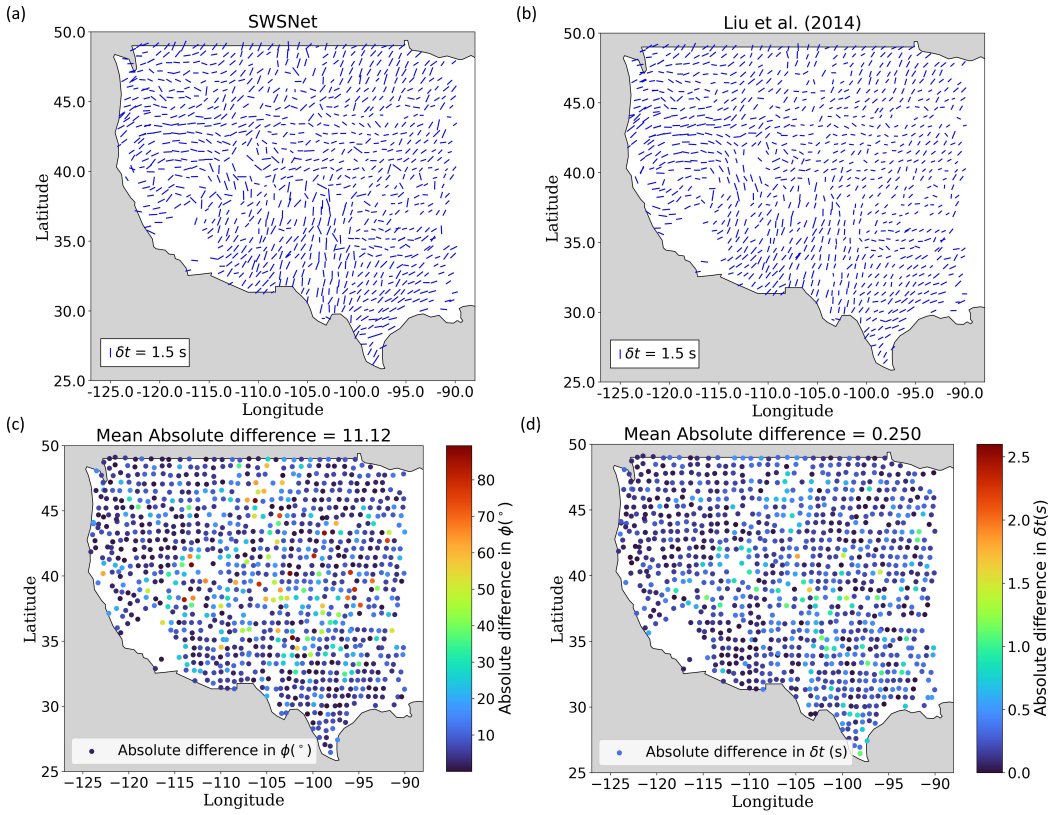


Figure 4. (a) Splitting parameters calculated by SWSNet (b) Splitting parameters calculated by Liu et al. (2014). The orientation of the straight lines is representative of the fast axis orientation while the length represents delay time. Similar general pattern is observed in both cases. (c) Station-wise comparison between ϕ calculated by SWSNet and Liu et al. (2014) (d) Station-wise comparison between δt calculated by SWSNet and Liu et al. (2014)

serve that grid search on an average takes 3-6 times the amount of time taken by SWSNet to calculate splitting parameters for a single waveform.

In this study we introduce a deep learning model SWSNet that has the potential to replace grid search methods used by previous studies to find splitting parameters for a waveform. Due to the dearth of labelled real data we train the model on synthetic data. We demonstrate that a direct application of model trained on the synthetic waveforms to real waveforms does not work well, the real waveform being affected by source mechanisms and path effects. This is resolved by using a deconvolution approach to minimise the difference between real and synthetic data. We retrain the model on transverse components of deconvolved synthetic waveforms contaminated by random noise, and show that the model learns to perform reasonably well in identifying the splitting parameters for such waveforms. We then apply our model to pre-selected waveforms from the USArray dataset and show that the station averages calculated using SWSNet follow the same general trends as previous studies. We observe that the robustness of the proposed method increases with increased number of measurements for a given station. The current version of the model is trained entirely on synthetic data, but in future versions real data can be added to the training set for improved representation.

4 Open Research

The raw seismic waveforms used in this study are open for download from the IRIS Data Management Center under the network code TA (IRIS Transportable Array, 2003). The event selection and corresponding labels used for training of SWSNet are available in the supplementary data alongside Link et al. (2022).

Acknowledgments

This research is supported by the "KI-Nachwuchswissenschaftlerinnen" - grant SAI 01IS20059 by the Bundesministerium für Bildung und Forschung - BMBF. GR gratefully acknowledges the support provided by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through grants RU 886/15 and RU 886/16. Calculations were performed at the Frankfurt Institute for Advanced Studies' new GPU cluster, funded by BMBF for the project Seismologie und Artifizielle Intelligenz (SAI). The authors are also thankful to Darius Fenner, Jonas Köhler and Kai Zhou for their kind suggestions.

References

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Ammon, C. J. (1991). The isolation of receiver effects from teleseismic P waveforms. *Bulletin of the Seismological Society of America*, *81*(6), 2504-2510. Retrieved from `https://doi.org/10.1785/BSSA0810062504` doi: 10.1785/BSSA0810062504

Barruol, G., Wuestefeld, A., & Bokelmann, G. (2009). SKS-Splitting-database. *Université de Montpellier, Laboratoire Géosciences*. doi: 10.18715/sks_splitting_database

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Holtzman, B. K., & Kendall, J.-M. (2010). Organized melt, seismic anisotropy, and plate boundary lubrication. *Geochemistry, Geophysics, Geosystems*, *11*(12). Retrieved from `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010GC003296` doi: https://doi.org/10.1029/2010GC003296

IRIS Transportable Array. (2003). *Usarray transportable array.* International Federation of Digital Seismograph Networks. Retrieved from `https://www.fdsn.org/networks/detail/TA/` doi: 10.7914/SN/TA

Jia, Y., Liu, K. H., Kong, F., Liu, L., & Gao, S. S. (2021, 07). A systematic investigation of piercing-point-dependent seismic azimuthal anisotropy. *Geophysical Journal International*, *227*(3), 1496-1511. Retrieved from `https://doi.org/10.1093/gji/ggab285` doi: 10.1093/gji/ggab285

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

Kiranyaz, S., Ince, T., Hamila, R., & Gabbouj, M. (2015). Convolutional Neural Networks for patient-specific ECG classification. , 2608-2611. doi: 10.1109/EMBC.2015.7318926

Langston, C. A. (1979). Structure under Mount Rainier, Washington, inferred from teleseismic body waves. *Journal of Geophysical Research: Solid Earth*, *84*(B9), 4749-4762. Retrieved from `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JB084iB09p04749` doi: https://doi.org/10.1029/JB084iB09p04749

Link, F., Reiss, M. C., & Rümpker, G. (2022). An automatized XKS-splitting procedure for large data sets: Extension package for SplitRacer and application to the USArray. *Computers & Geosciences*, *158*, 104961. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0098300421002478` doi: https://doi.org/10.1016/j.cageo.2021.104961

Liu, K. H., Elsheikh, A., Lemnifi, A., Purevsuren, U., Ray, M., Refayee, H., ... Gao, S. S. (2014). A uniform database of teleseismic shear wave splitting measurements for the western and central United States. *Geochemistry, Geophysics, Geosystems*, *15*(5), 2075-2085. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014GC005267 doi: https://doi.org/10.1002/2014GC005267

Liu, K. H., & Gao, S. S. (2013, 10). Making Reliable Shear-Wave Splitting Measurements. *Bulletin of the Seismological Society of America*, *103*(5), 2680-2693. Retrieved from https://doi.org/10.1785/0120120355 doi: 10.1785/0120120355

Long, M. D., & Silver, P. G. (2009). Shear Wave Splitting and Mantle Anisotropy: Measurements, Interpretations, and New Directions. *Surveys in Geophysics*, *30*, 407-461.

Owens, T. J., Zandt, G., & Taylor, S. R. (1984, 9). Seismic evidence for an ancient rift beneath the cumberland plateau, Tennessee: A detailed analysis of broadband teleseismic P waveforms. *J. Geophys. Res.; (United States)*. Retrieved from https://www.osti.gov/biblio/5820478 doi: 10.1029/JB089iB09p07783

Prechelt, L. (2012). Early stopping — but when? In *Neural networks: Tricks of the trade: Second edition* (pp. 53–67). Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-35289-8_5 doi: 10.1007/978-3-642-35289-8_5

Reiss, M. C., & Rümpker, G. (2017, 01). SplitRacer: MATLAB Code and GUI for Semiautomated Analysis and Interpretation of Teleseismic Shear-Wave Splitting. *Seismological Research Letters*, *88*(2A), 392-409. Retrieved from https://doi.org/10.1785/0220160191 doi: 10.1785/0220160191

Rümpker, G., Kaviani, A., Link, F., Reiss, M., & Komeazi, A. (2023). Testing observables for teleseismic shear-wave splitting inversions: ambiguities of intensities, parameters, and waveforms. *Ann. Geophys.*, *66*. Retrieved from https://www.annalsofgeophysics.eu/index.php/annals/article/view/8870 doi: https://doi.org/10.4401/ag-8870

Savage, M. K. (1999). Seismic anisotropy and mantle deformation: What have we learned from shear wave splitting? *Reviews of Geophysics*, *37*(1), 65-106. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/98RG02075 doi: https://doi.org/10.1029/98RG02075

Savage, M. K., Wessel, A., Teanby, N. A., & Hurst, A. W. (2010). Automatic measurement of shear wave splitting and applications to time varying anisotropy at Mount Ruapehu volcano, New Zealand. *Journal of Geophysical Research: Solid Earth*, *115*(B12). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010JB007722 doi: https://doi.org/10.1029/2010JB007722

Silver, P. G., & Chan, W. W. (1991). Shear wave splitting and subcontinental mantle deformation. *Journal of Geophysical Research: Solid Earth*, *96*(B10), 16429-16454. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/91JB00899 doi: https://doi.org/10.1029/91JB00899

Teanby, N. A., Kendall, J.-M., & van der Baan, M. (2004, 04). Automation of Shear-Wave Splitting Measurements using Cluster Analysis. *Bulletin of the Seismological Society of America*, *94*(2), 453-463. Retrieved from https://doi.org/10.1785/0120030123 doi: 10.1785/0120030123

Wuestefeld, A., Al-Harrasi, O., Verdon, J. P., Wookey, J., & Kendall, J. M. (2010). A strategy for automated analysis of passive microseismic data to image seismic anisotropy and fracture characteristics. *Geophysical Prospecting*, *58*(5), 755-773. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2478.2010.00891.x doi: https://doi.org/10.1111/j.1365-2478.2010.00891.x

Wüstefeld, A., Bokelmann, G., Zaroli, C., & Barruol, G. (2008). SplitLab: A shear-wave splitting environment in Matlab. *Computers & Geosciences*, *34*(5), 515-528. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0098300407001859` doi: https://doi.org/10.1016/j.cageo.2007.08.002

Zhang, Y., & Gao, S. S. (2022). Classification of Teleseismic Shear Wave Splitting Measurements: A Convolutional Neural Network Approach. *Geophysical Research Letters*, *49*(12), e2021GL097101. Retrieved from `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021GL097101` (e2021GL097101 2021GL097101) doi: https://doi.org/10.1029/2021GL097101

Figure 1.

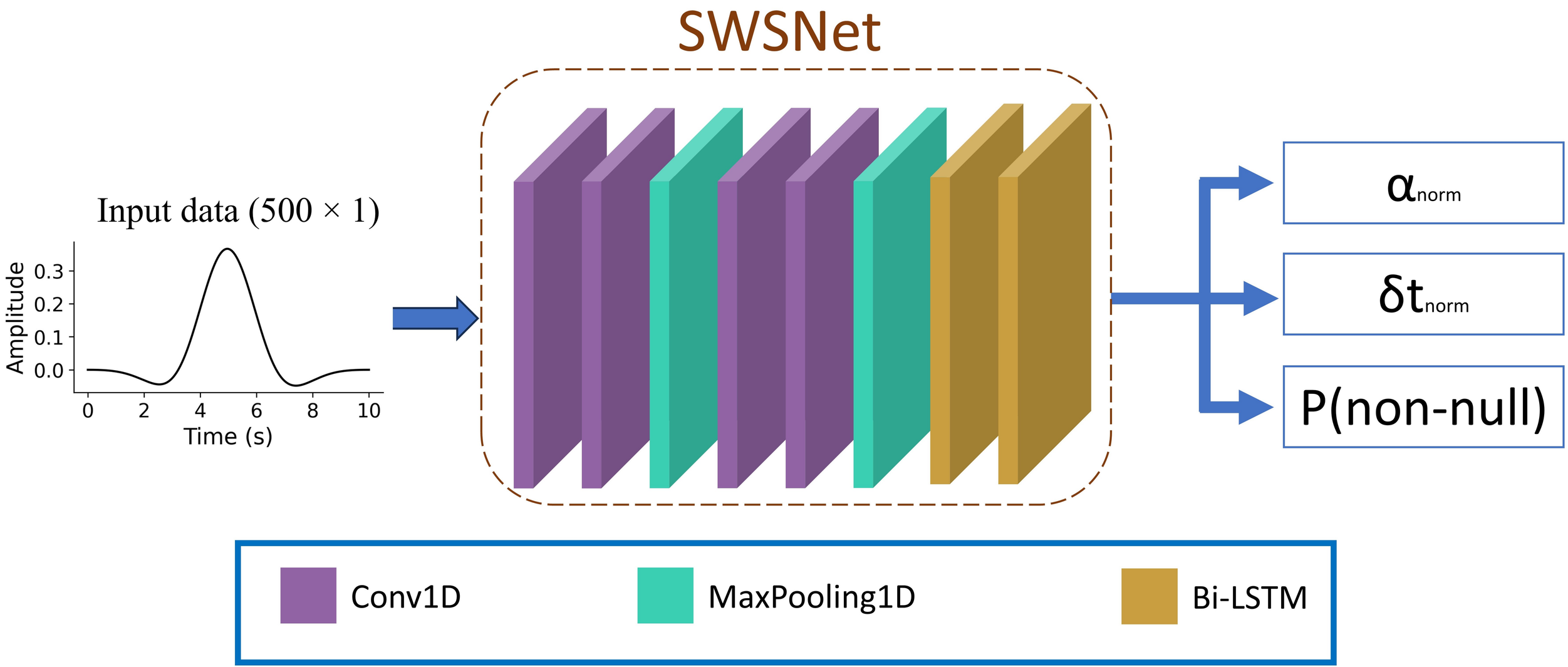

SWSNet
Input data (500 × 1)
Amplitude
0.3
0.2
0.1
0.0
0
2
4
6
8
10
Time (s)
αnorm
δtnorm
P(non-null)
Conv1D
MaxPooling1D
Bi-LSTM

Figure 2.

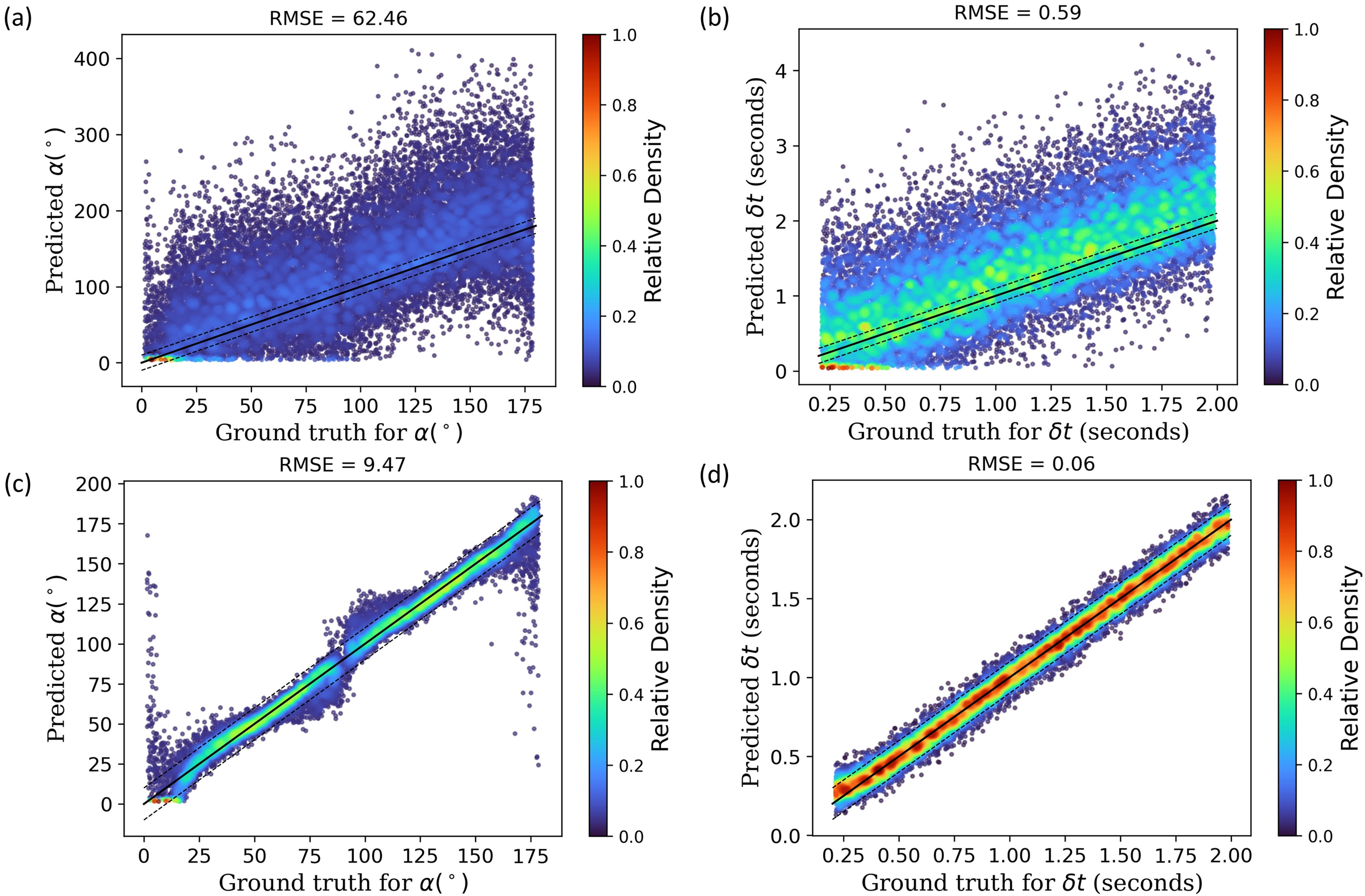

(a)
RMSE = 62.46
Predicted α(°)
Ground truth for α(°)
Relative Density
(b)
RMSE = 0.59
Predicted δt (seconds)
Ground truth for δt (seconds)
Relative Density
(c)
RMSE = 9.47
Predicted α(°)
Ground truth for α(°)
Relative Density
(d)
RMSE = 0.06
Predicted δt (seconds)
Ground truth for δt (seconds)
Relative Density

Figure 3.

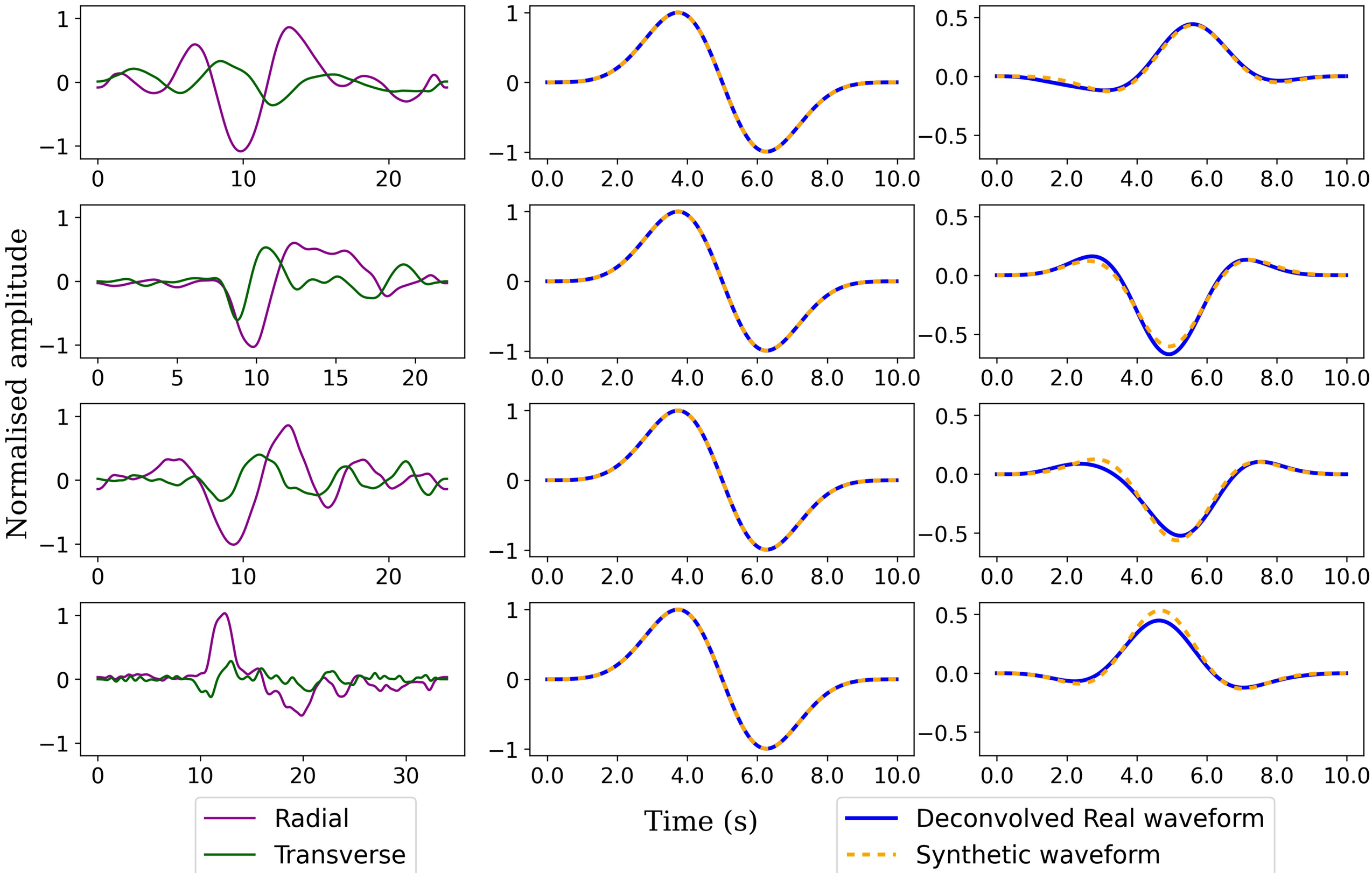

Normalised amplitude
Time (s)
Radial
Transverse
Deconvolved Real waveform
Synthetic waveform

Figure 4.

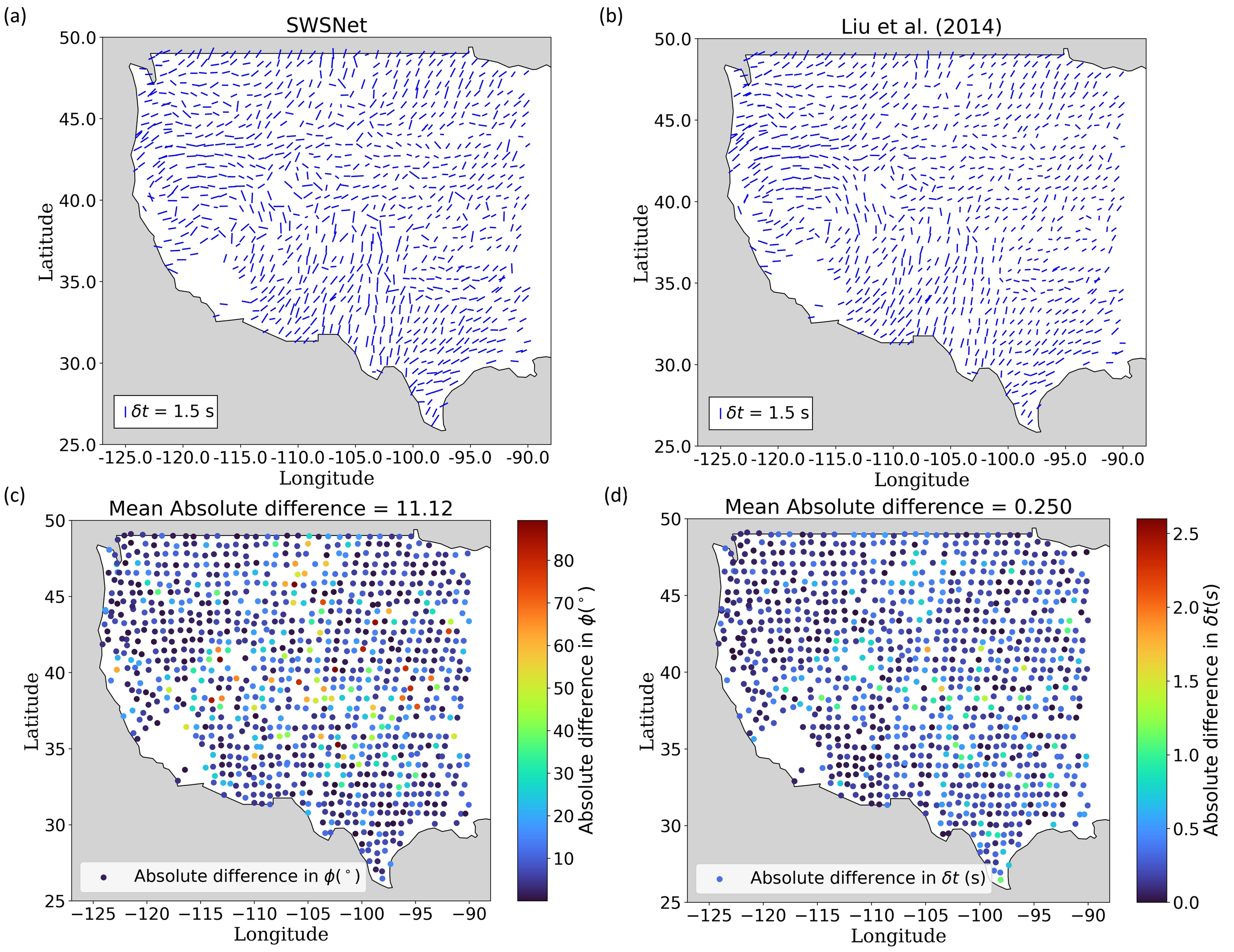

(a)
SWSNet
Latitude
Longitude
δt = 1.5 s
(b)
Liu et al. (2014)
Latitude
Longitude
δt = 1.5 s
(c)
Mean Absolute difference = 11.12
Latitude
Longitude
Absolute difference in ϕ(°)
Absolute difference in ϕ(°)
(d)
Mean Absolute difference = 0.250
Latitude
Longitude
Absolute difference in δt (s)
Absolute difference in δt(s)

Figure S1.

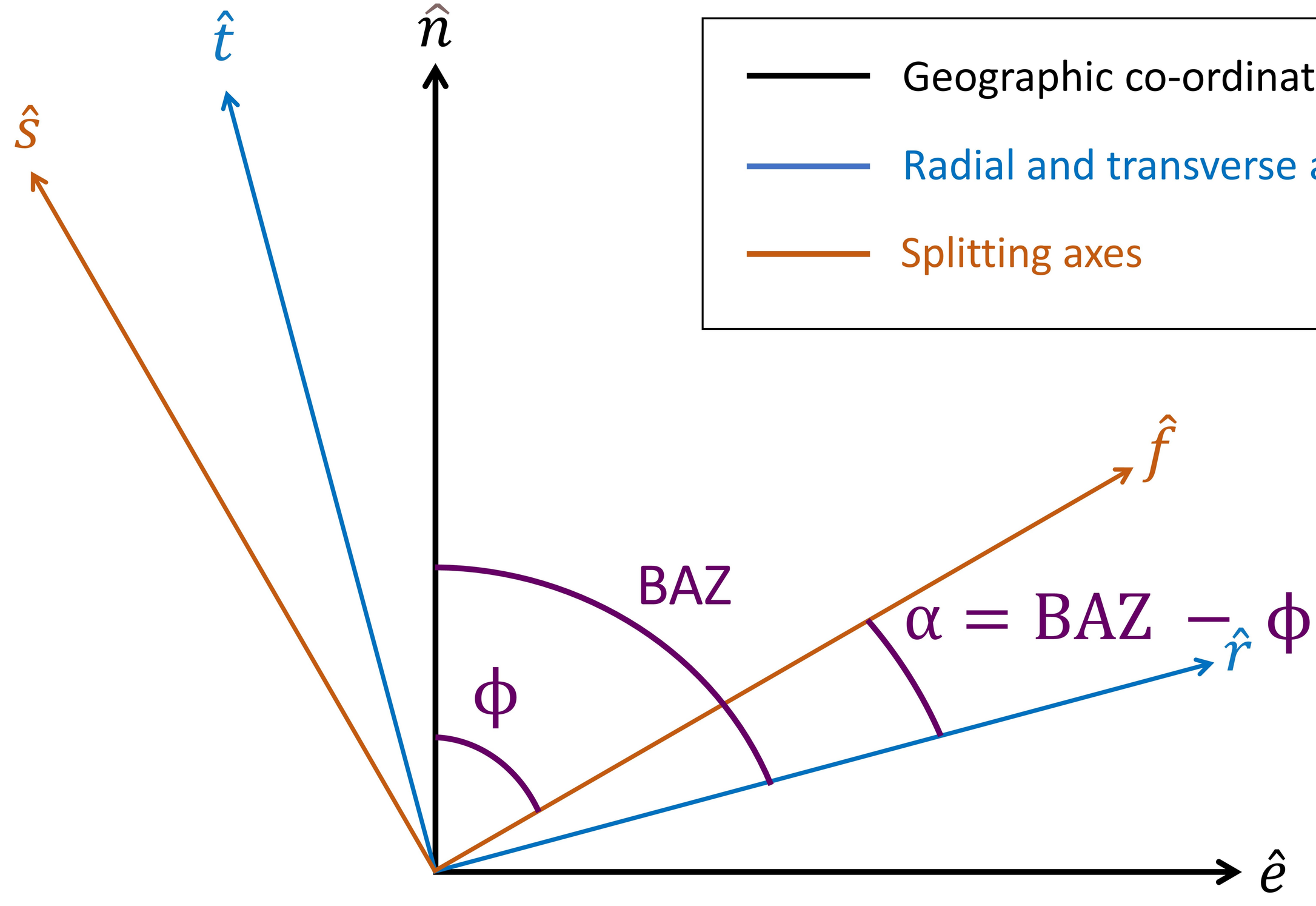

$\hat{t}$
$\hat{n}$
$\hat{s}$
Geographic co-ordinates
Radial and transverse axes
Splitting axes
$\hat{f}$
BAZ
$\alpha = \mathrm{BAZ} - \phi$
$\hat{r}$
ϕ
$\hat{e}$

Figure S2.

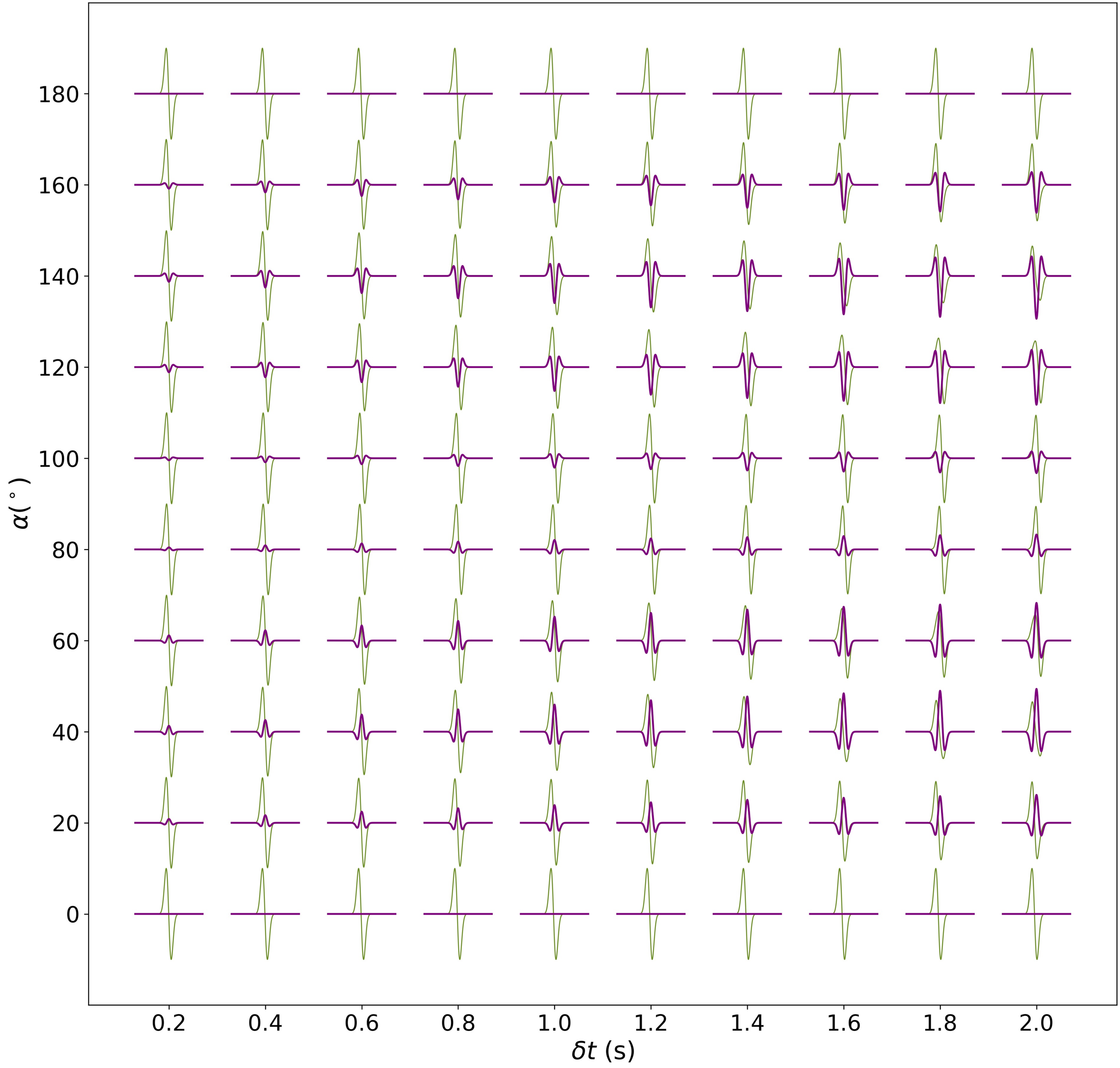

180
160
140
120
100
80
60
40
20
0
α(°)
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
δt (s)

Figure S3.

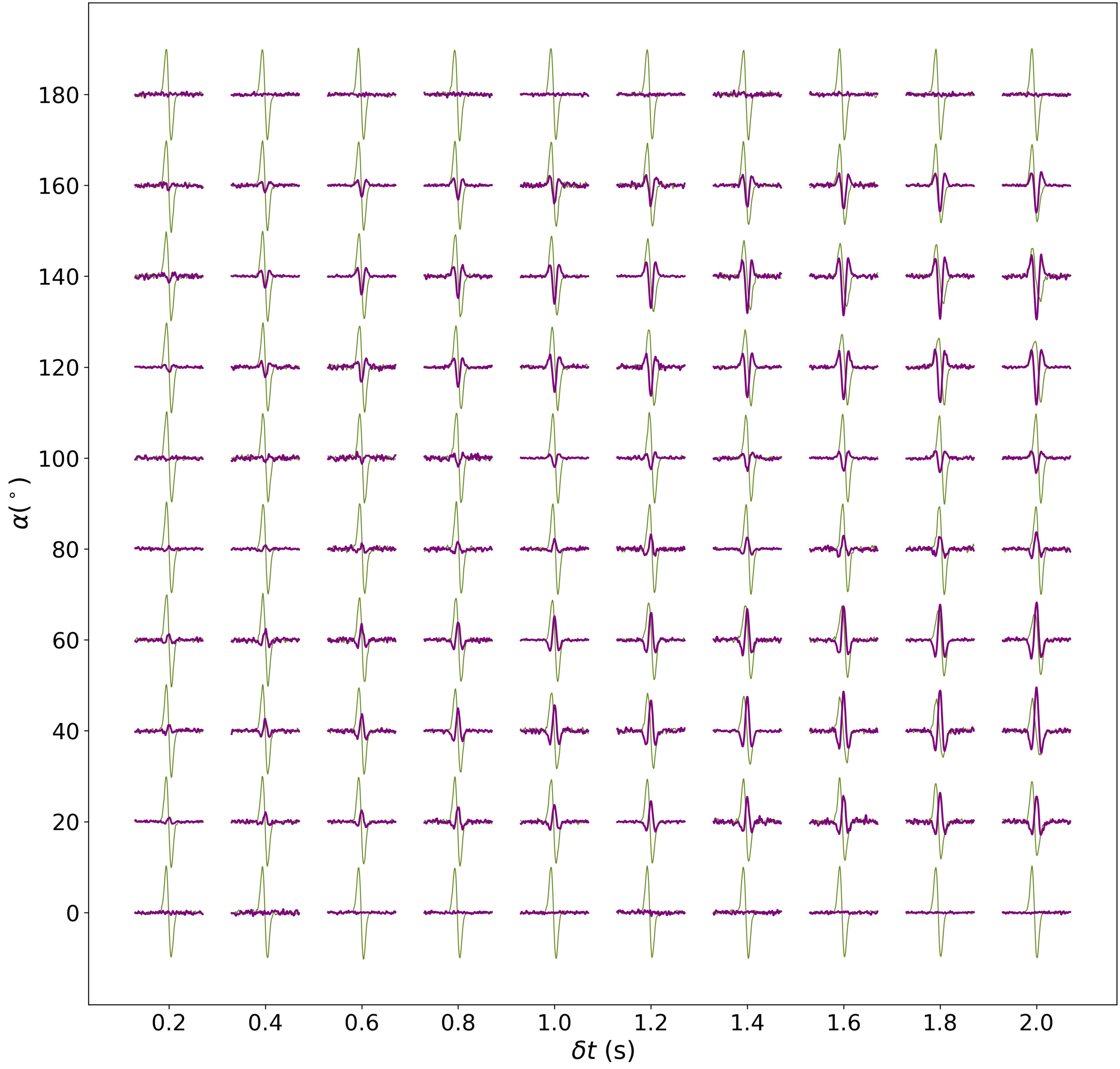

180
160
140
120
100
80
60
40
20
0
α(°)
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
δt (s)

Figure S4.

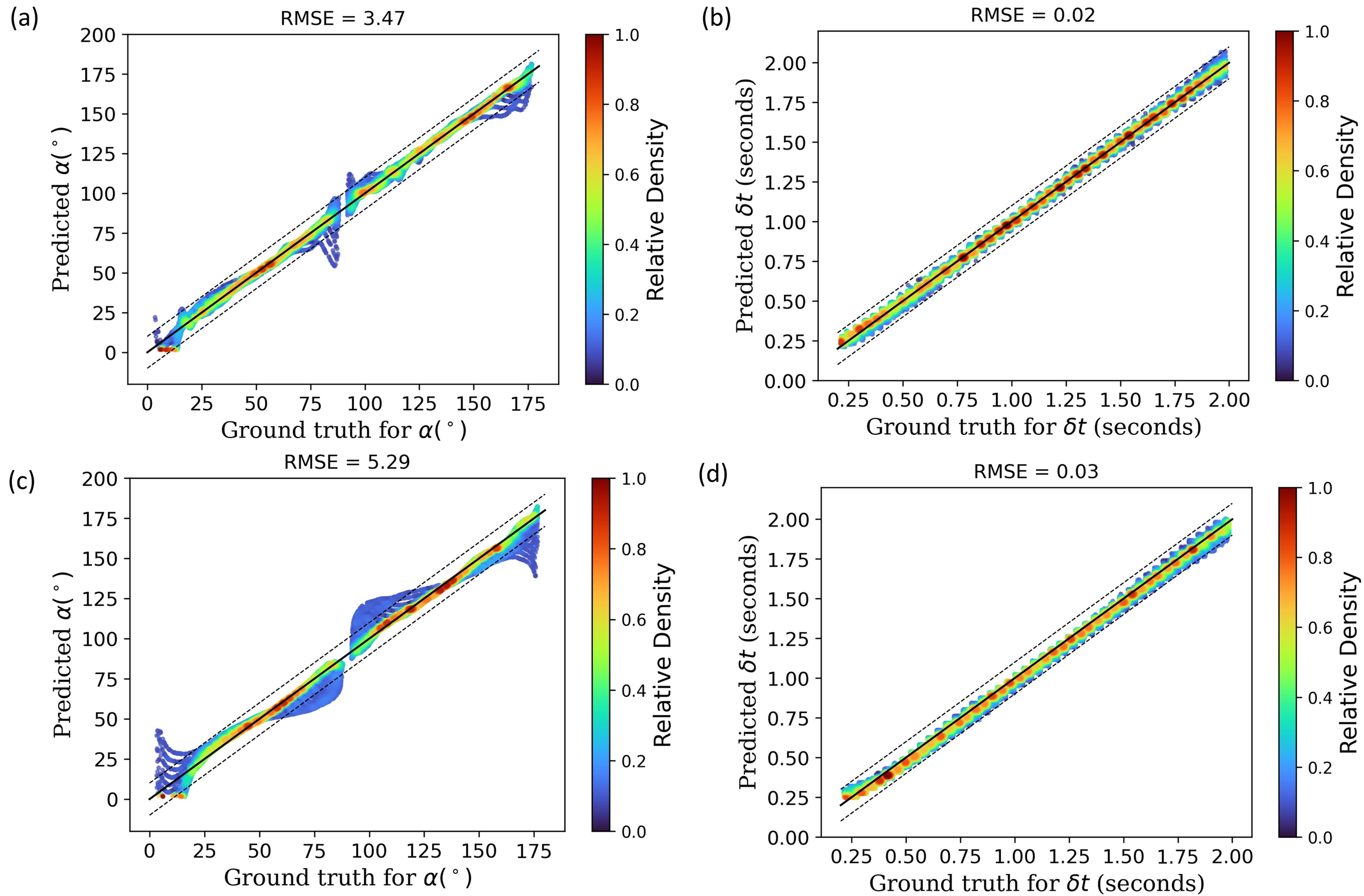

(a)
RMSE = 3.47
Predicted α(°)
Ground truth for α(°)
Relative Density
(b)
RMSE = 0.02
Predicted δt (seconds)
Ground truth for δt (seconds)
Relative Density
(c)
RMSE = 5.29
Predicted α(°)
Ground truth for α(°)
Relative Density
(d)
RMSE = 0.03
Predicted δt (seconds)
Ground truth for δt (seconds)
Relative Density

Figure S5.

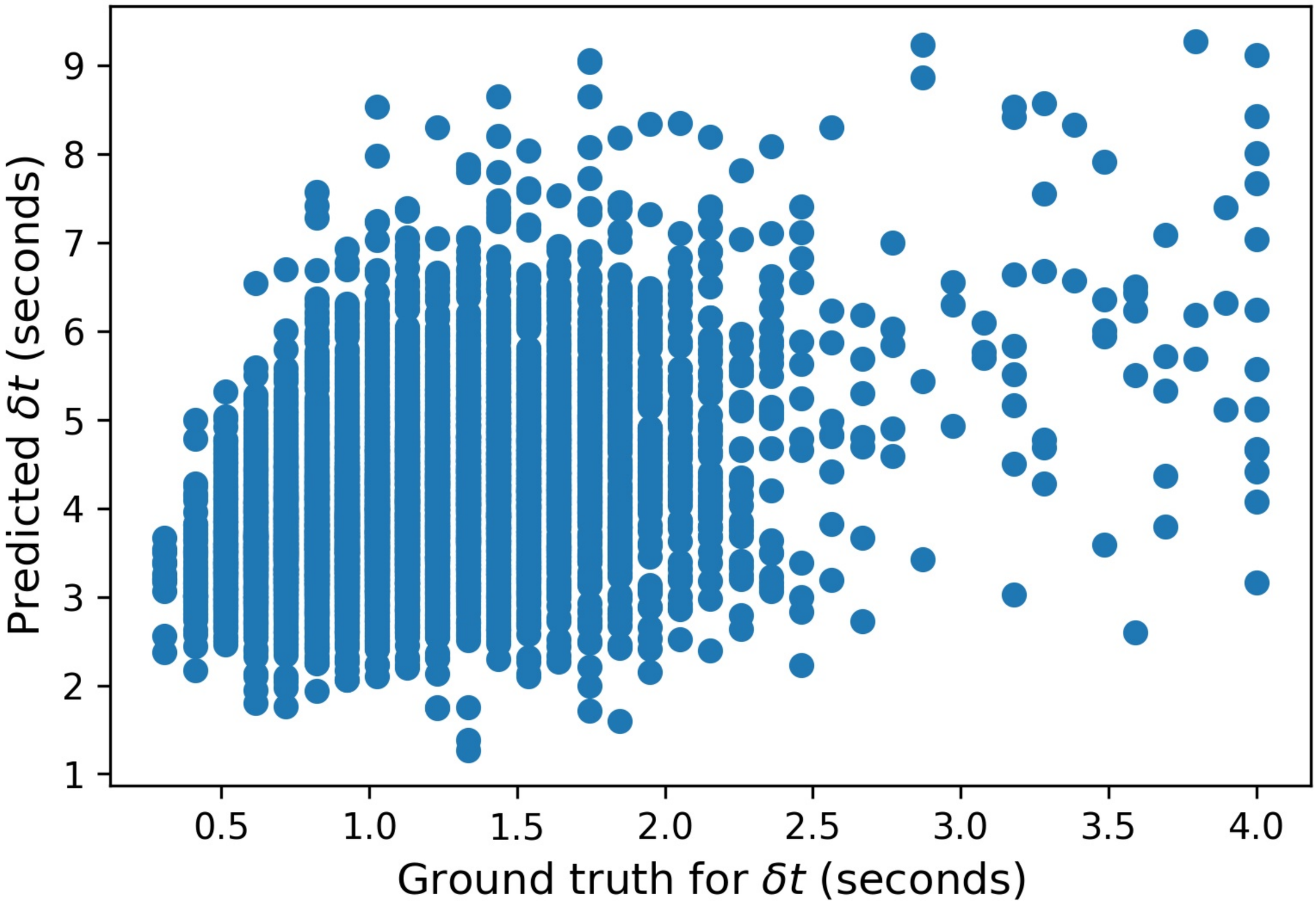

Predicted δt (seconds)
Ground truth for δt (seconds)
1
2
3
4
5
6
7
8
9
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Figure S6.

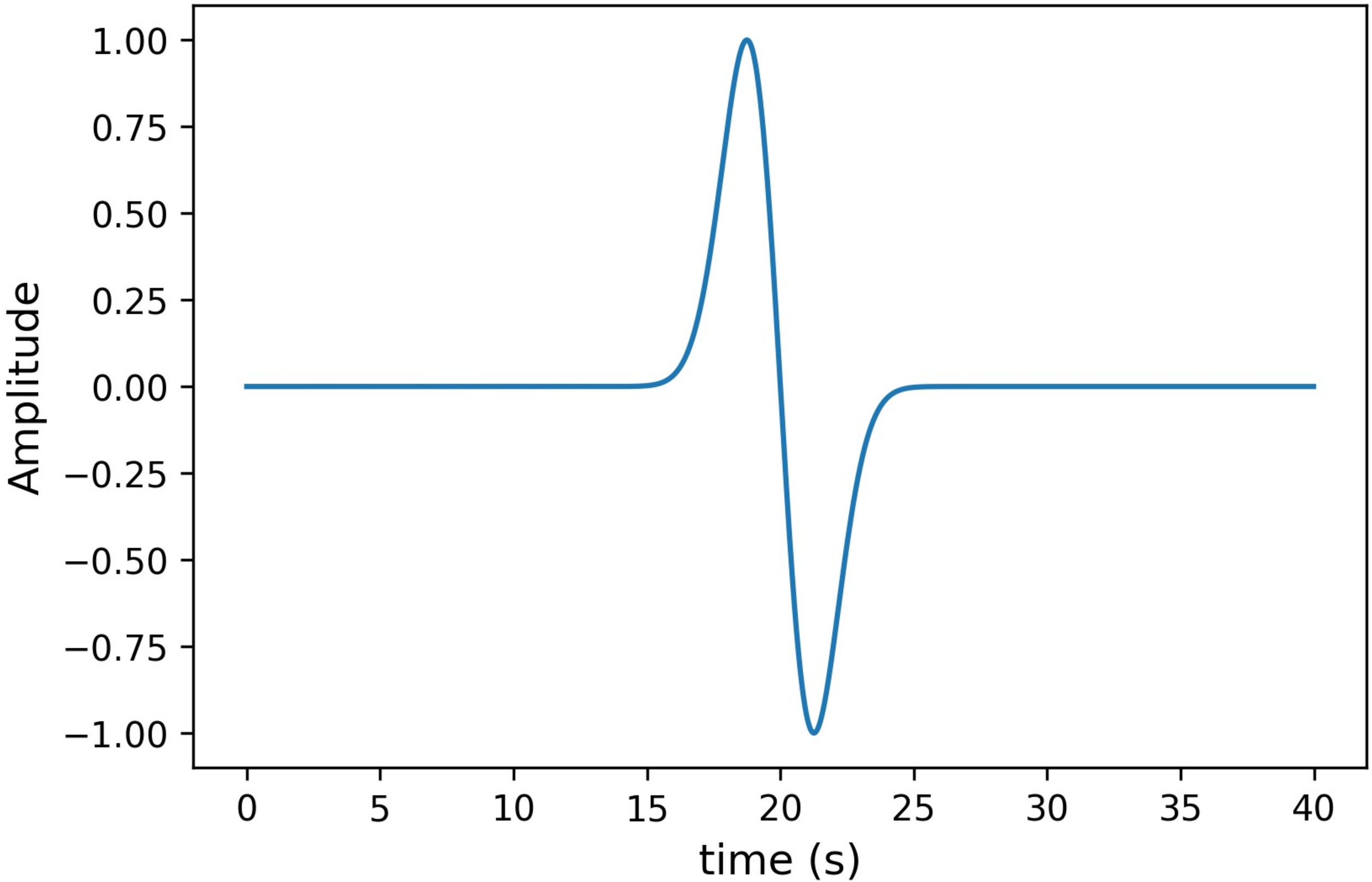

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Amplitude
0
5
10
15
20
25
30
35
40
time (s)

Figure S7.

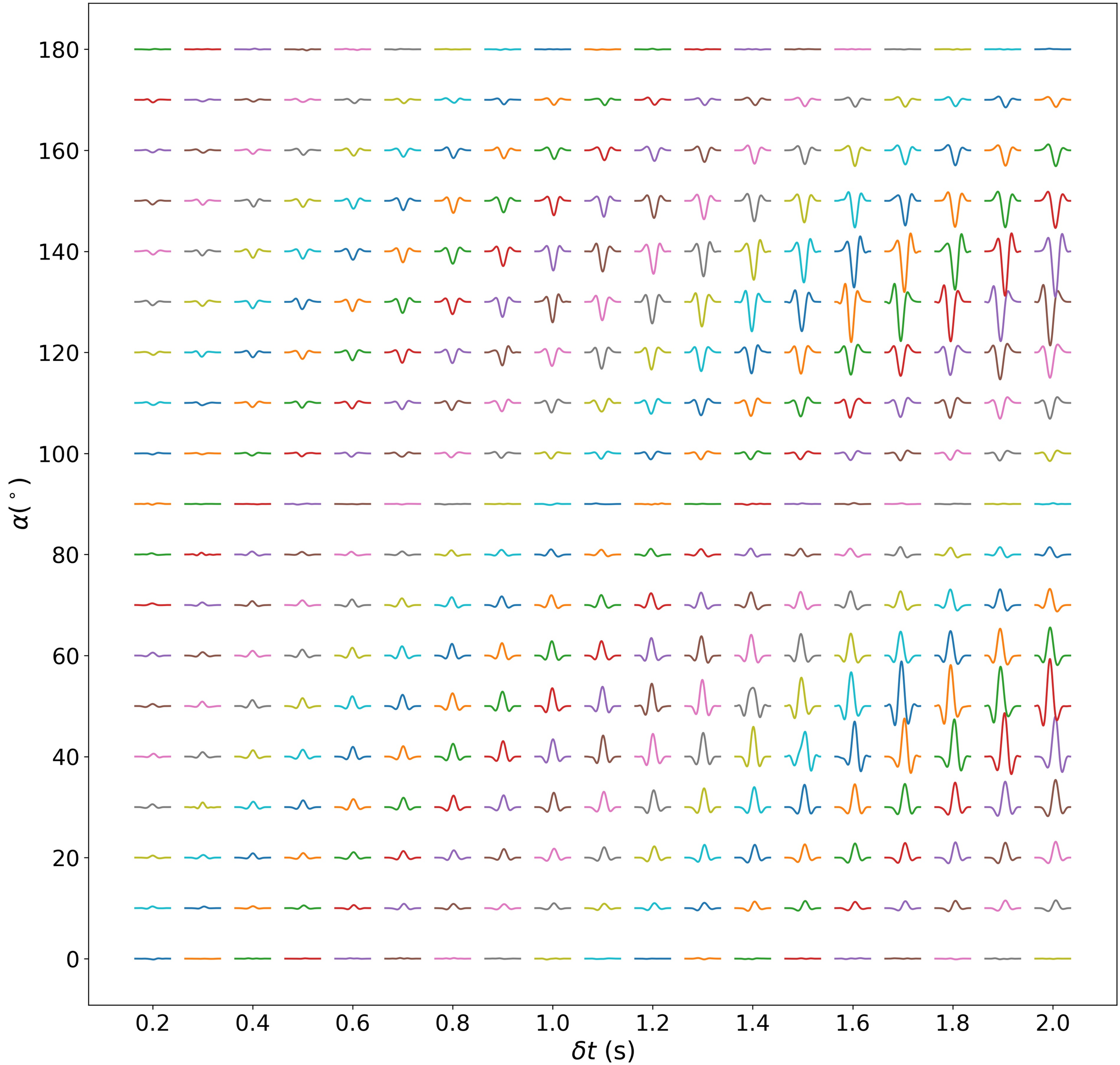
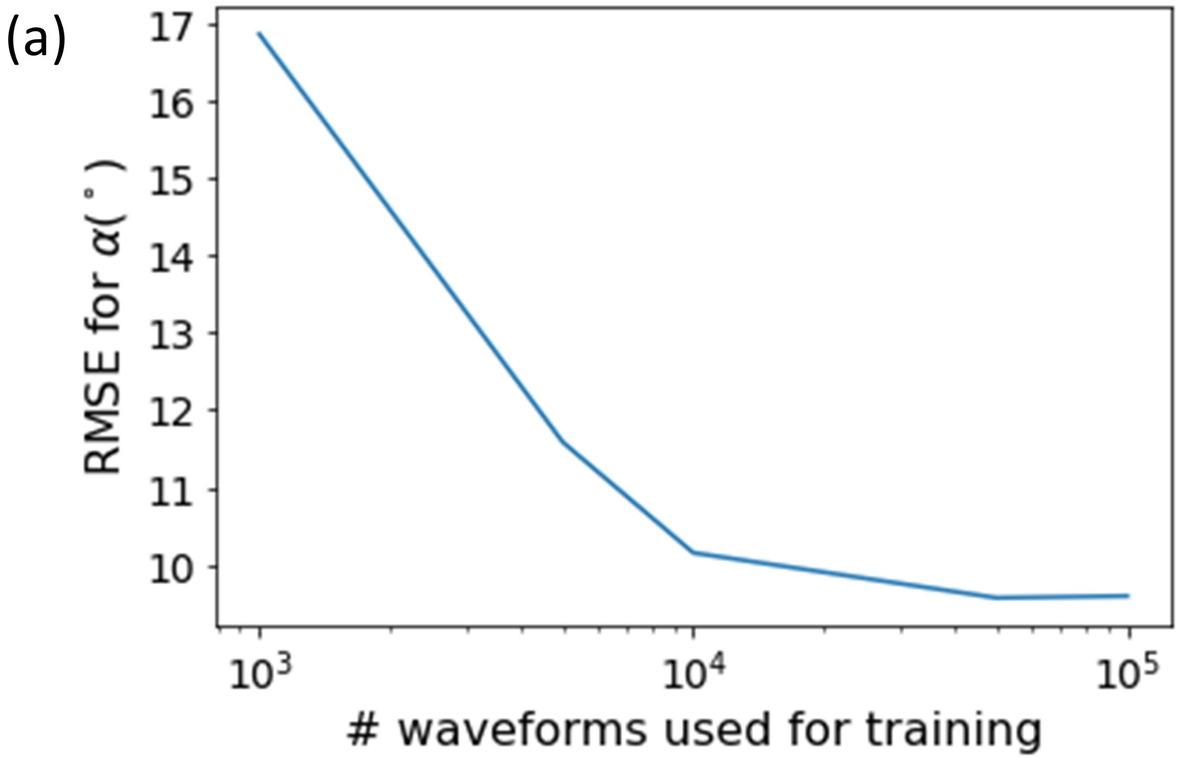
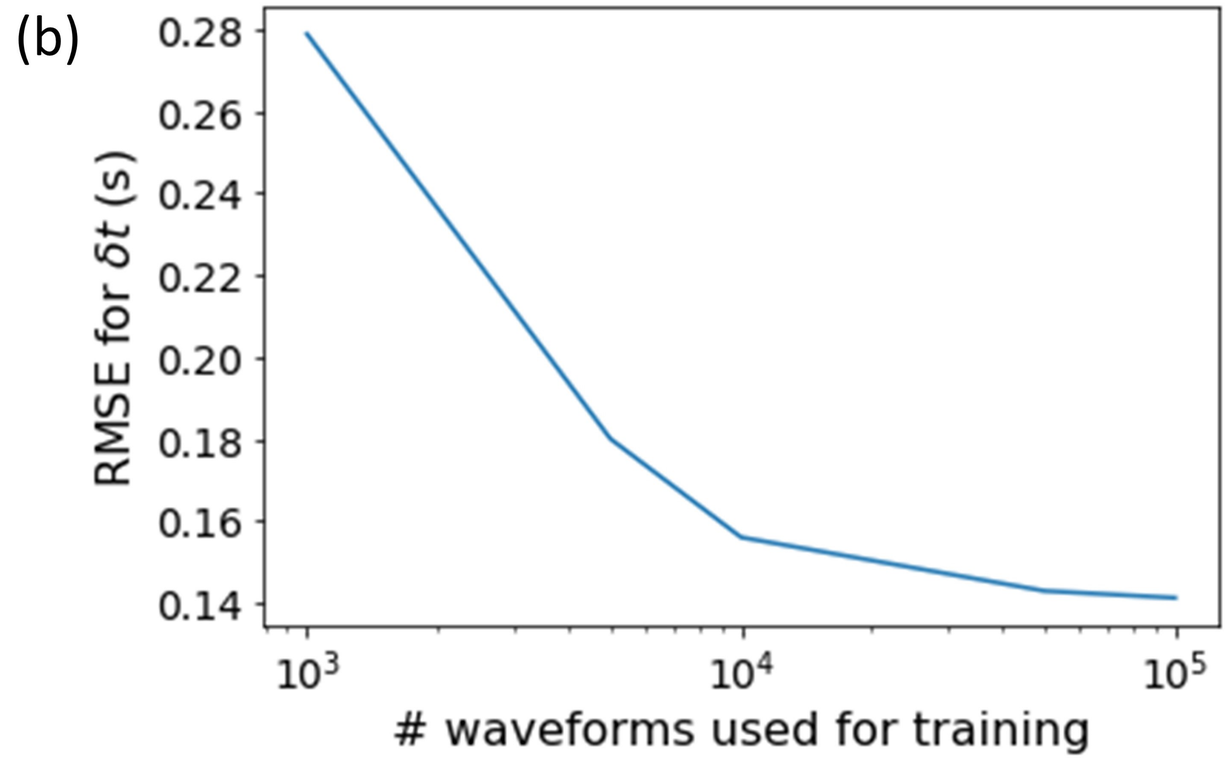

180
160
140
120
100
80
60
40
20
0
α(°)
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
δt (s)

Figure S8.

Input Data
1D Convolution
filters = 8; kernel size = 10
1D Convolution
filters = 6; kernel size = 4
1D Maxpooling
kernel size = 2
1D Convolution
filters = 8; kernel size = 5
1D Convolution
filters = 6; kernel size = 4
1D Maxpooling
kernel size = 2
Bi-directional LSTM
dimension = 128
Bi-directional LSTM
dimension = 64
Probability of non-null measurement
Prediction for δt
Prediction for φ

Figure S9.


(a)
RMSE for α(°)
17
16
15
14
13
12
11
10
10^3
10^4
10^5
waveforms used for training


(b)
RMSE for δt (s)
0.28
0.26
0.24
0.22
0.20
0.18
0.16
0.14
10^3
10^4
10^5
waveforms used for training

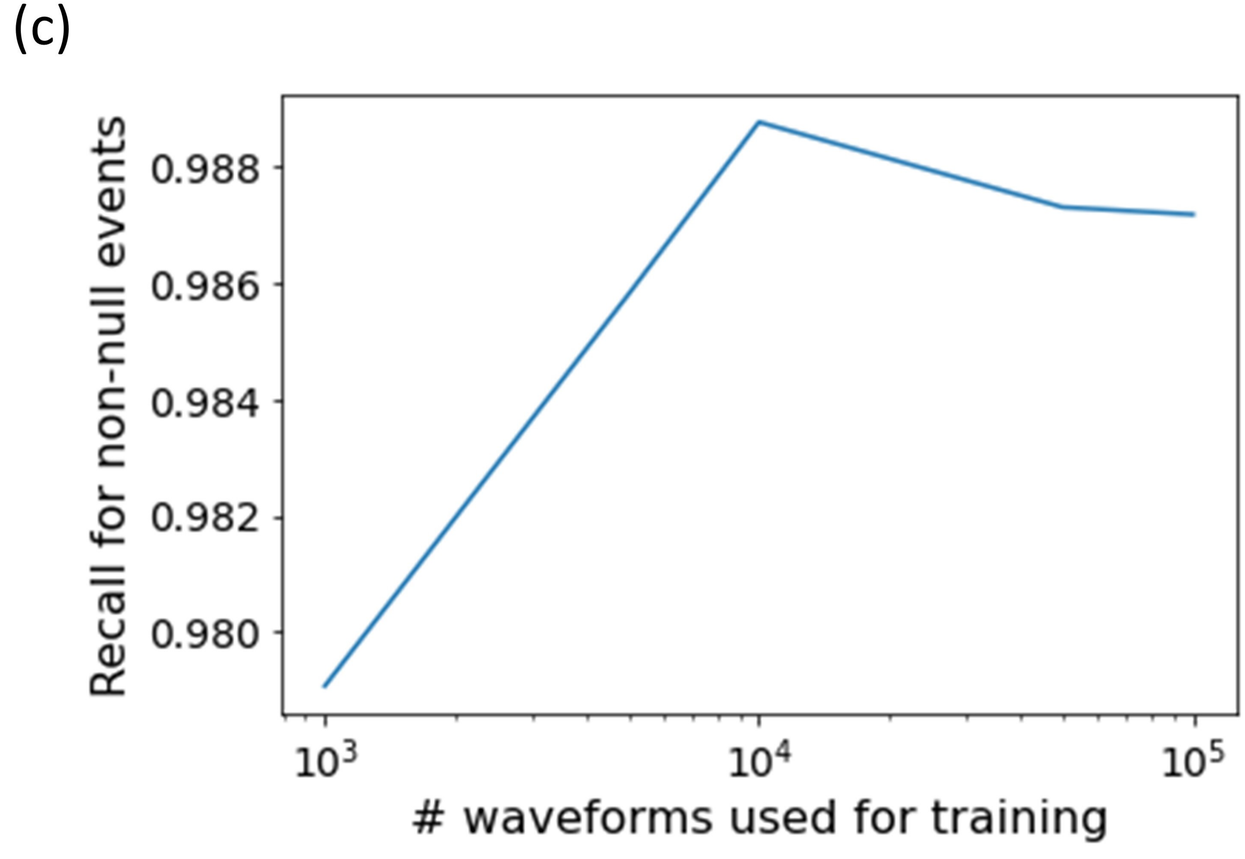

(c)
Recall for non-null events
0.988
0.986
0.984
0.982
0.980
10^3
10^4
10^5
waveforms used for training

Figure S10.

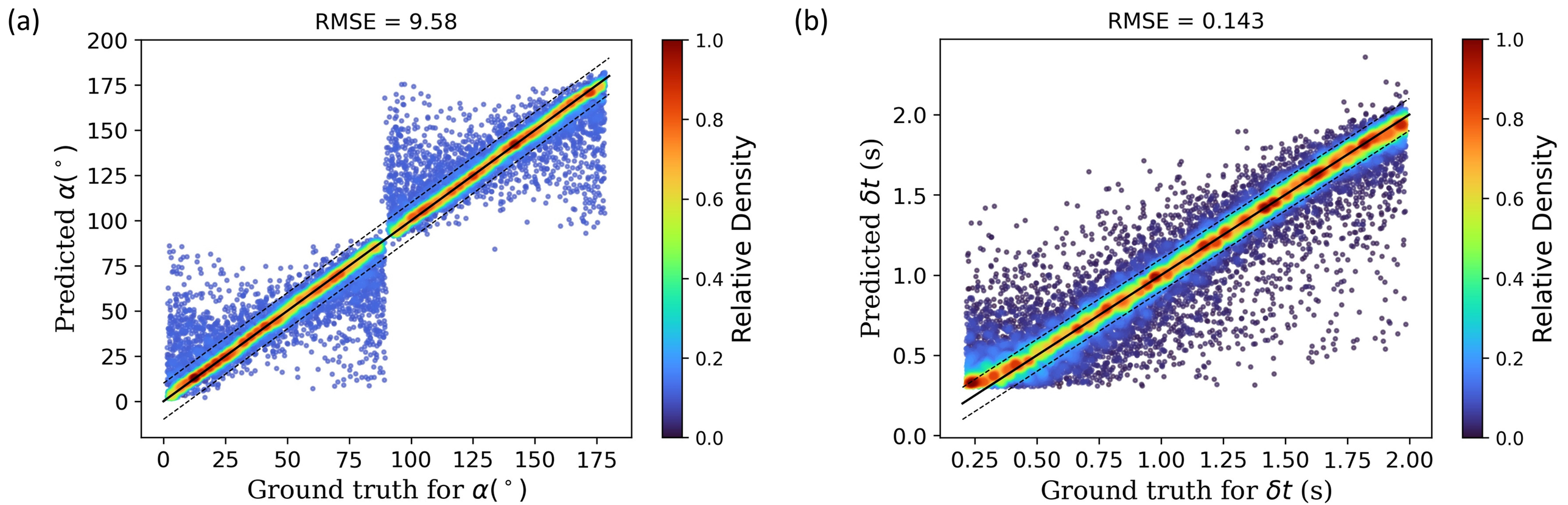

(a)
RMSE = 9.58
Predicted α(°)
Ground truth for α(°)
Relative Density
(b)
RMSE = 0.143
Predicted δt (s)
Ground truth for δt (s)
Relative Density

Figure S11.

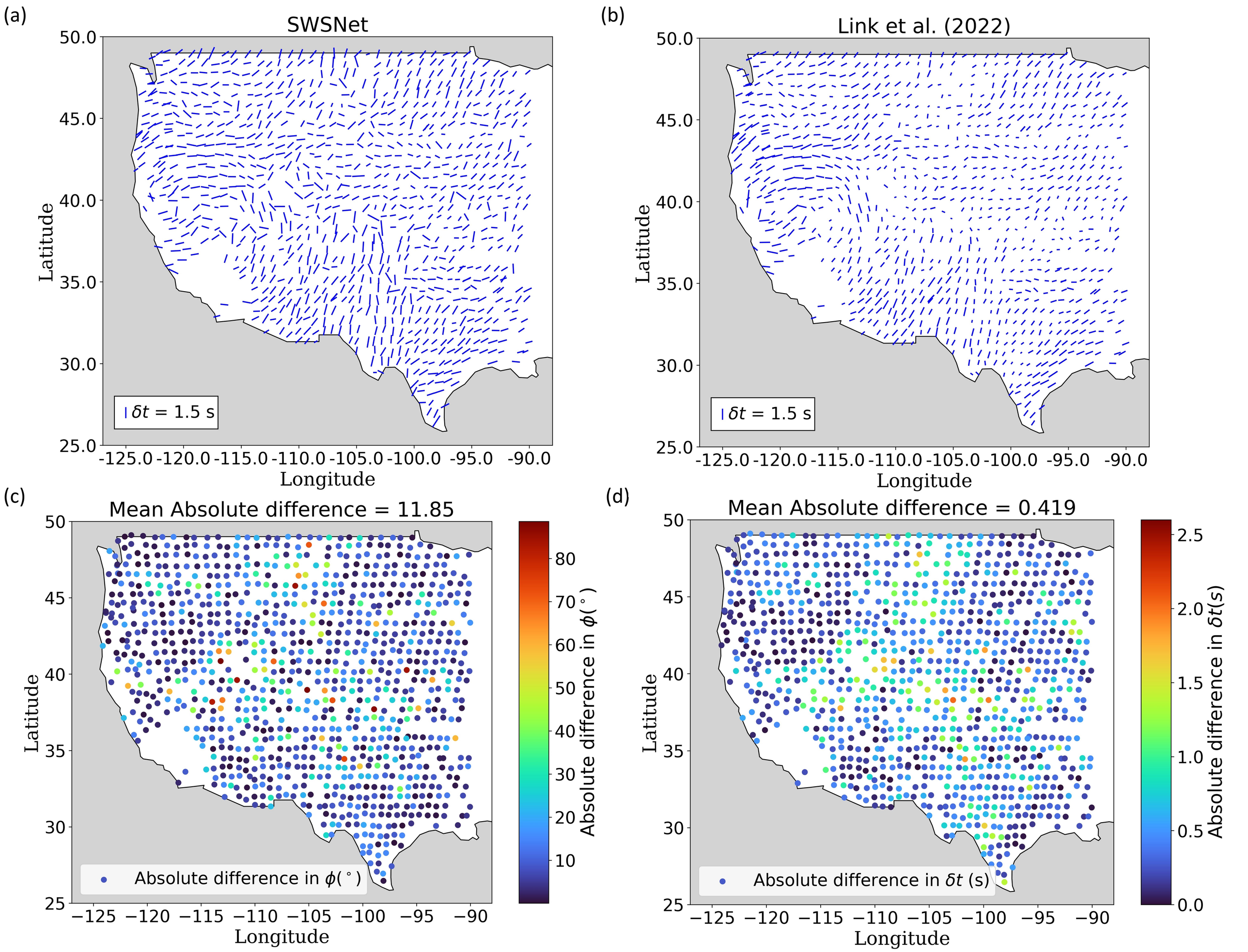

(a)
SWSNet
Latitude
Longitude
δt = 1.5 s
(b)
Link et al. (2022)
Latitude
Longitude
δt = 1.5 s
(c)
Mean Absolute difference = 11.85
Latitude
Longitude
Absolute difference in ϕ(°)
Absolute difference in ϕ(°)
(d)
Mean Absolute difference = 0.419
Latitude
Longitude
Absolute difference in δt (s)
Absolute difference in δt(s)

Figure S12.

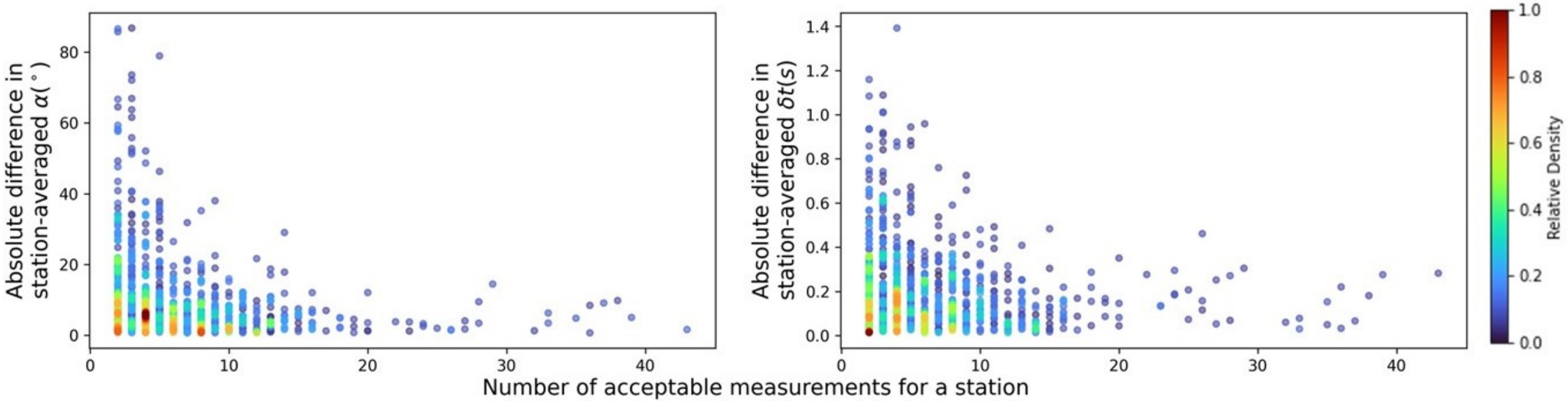
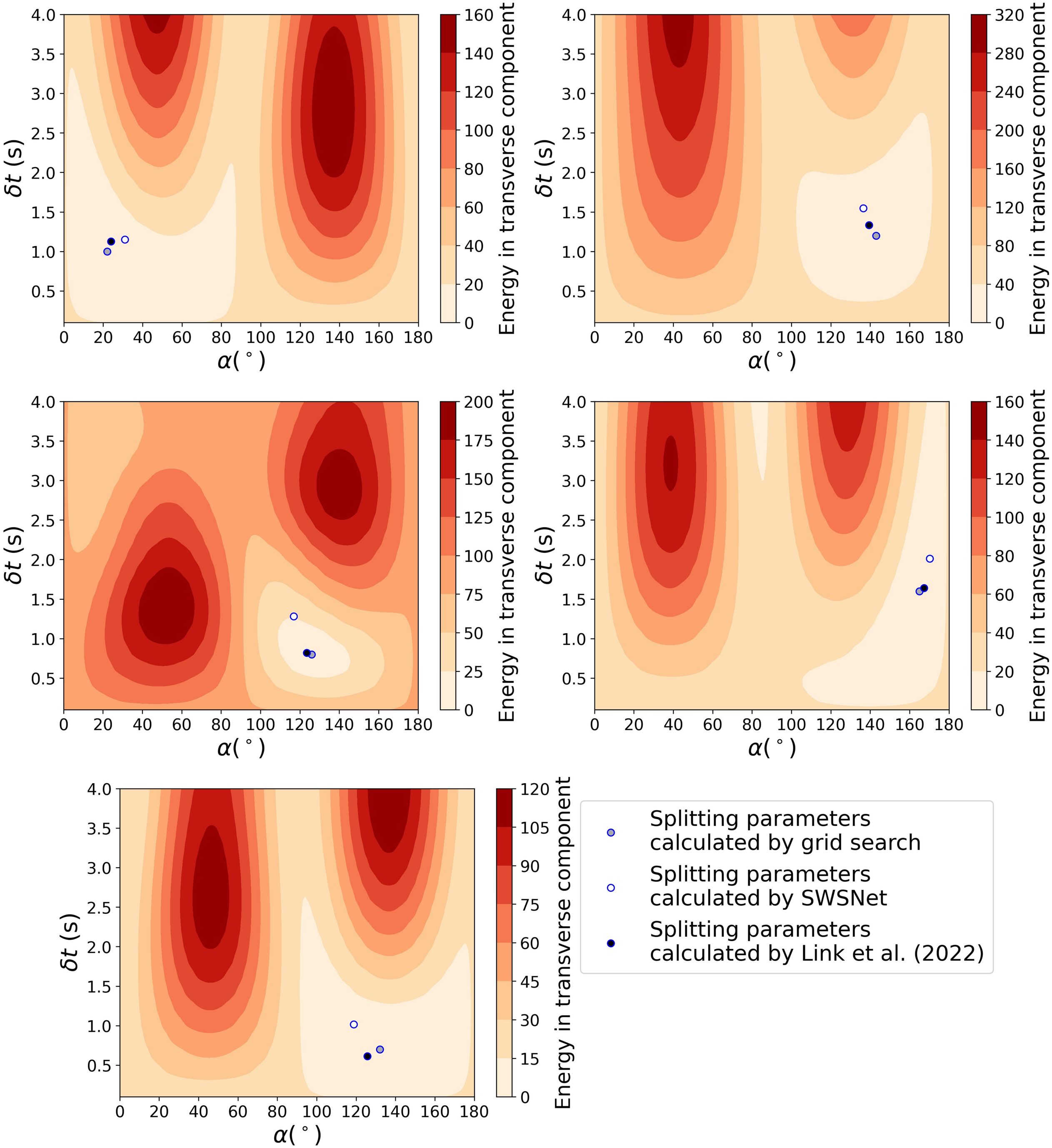

Absolute difference in station-averaged α(°)
Absolute difference in station-averaged δt(s)
Number of acceptable measurements for a station
Relative Density
0
20
40
60
80
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0

Figure S13.


δt (s)
α(°)
Energy in transverse component
Splitting parameters calculated by grid search
Splitting parameters calculated by SWSNet
Splitting parameters calculated by Link et al. (2022)