Seismic source characterization from GNSS data using deep learning

Giuseppe Costantino, Sophie Giffard-Roisin, David Marsan, Lou Marill, Mathilde Radiguet, Mauro Dalla Mura, Gaël Janex, Anne Socquet

JGR Solid Earth

RESEARCH ARTICLE
10.1029/2022JB024930

Seismic Source Characterization From GNSS Data Using Deep Learning

Giuseppe Costantino[1], Sophie Giffard-Roisin[1], David Marsan[2], Lou Marill[1,2], Mathilde Radiguet[1], Mauro Dalla Mura[3,4], Gaël Janex[1], and Anne Socquet[1]

[1]Université Grenoble Alpes, Saint-Martin-d'Hères, France, [2]Université Savoie Mont Blanc, CNRS, IRD, Chambéry, France, [3]Université Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, Grenoble, France, [4]Institut Universitaire de France (IUF), Grenoble, France

Key Points:

- We develop deep learning approaches on synthetics mimicking the spatio-temporal structure of static deformation and realistic Global Navigation Satellite System (GNSS) noise
- We design three deep learning models and we train and test them against three GNSS data representations
- Transformers and image time series of deformation can effectively characterize small deformation patterns associated with the seismic source

Supporting Information:

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

Correspondence to:

G. Costantino,
giuseppe.costantino@univ-grenoble-alpes.fr

Citation:

Costantino, G., Giffard-Roisin, S., Marsan, D., Marill, L., Radiguet, M., Mura, M. D., et al. (2023). Seismic source characterization from GNSS data using deep learning. *Journal of Geophysical Research: Solid Earth*, *128*, e2022JB024930. https://doi.org/10.1029/2022JB024930

Received 10 JUN 2022
Accepted 6 APR 2023

Author Contributions:

Conceptualization: Sophie Giffard-Roisin, Mauro Dalla Mura, Anne Socquet
Data curation: Giuseppe Costantino, Lou Marill, Gaël Janex
Formal analysis: Anne Socquet
Funding acquisition: Anne Socquet
Investigation: Anne Socquet
Software: Giuseppe Costantino
Supervision: Sophie Giffard-Roisin, David Marsan, Mathilde Radiguet, Mauro Dalla Mura, Anne Socquet

Abstract The detection of deformation in Global Navigation Satellite System (GNSS) time series associated with (a)seismic events down to a low magnitude is still a challenging issue. The presence of a considerable amount of noise in the data makes it difficult to reveal patterns of small ground deformation. Traditional analyses and methodologies are able to effectively retrieve the deformation associated with medium to large magnitude events. However, the automatic detection and characterization of such events is still a complex task, because traditionally employed methods often separate the time series analysis from the source characterization. Here we propose a first end-to-end framework to characterize seismic sources using geodetic data by means of deep learning, which can be an efficient alternative to the traditional workflow, possibly overcoming its performance. We exploit three different geodetic data representations in order to leverage the intrinsic spatio-temporal structure of the GNSS noise and the target signal associated with (slow) earthquake deformation. We employ time series, images, and image time series to account for the temporal, spatial, and spatio-temporal domain, respectively. Thereafter, we design and develop a specific deep learning model for each dataset. We analyze the performance of the tested models both on synthetic and real data from North Japan, showing that image time series of geodetic deformation can be an effective data representation to embed the spatio-temporal evolution, with the associated deep learning method outperforming the other two. Therefore, jointly accounting for the spatial and temporal evolution may be the key to effectively detect and characterize fast or slow earthquakes.

Plain Language Summary The continuous monitoring of ground displacement with Global Navigation Satellite System allowed, at the beginning of the 2000s, the discovery of slow earthquakes—a transient slow slippage of tectonic faults that releases stress without generating seismic waves. Nevertheless, the detection of small events is still a challenge, because they are hidden in the noise. Most of the methods which are traditionally employed are able to extract the deformation down to a certain signal-to-noise level. However, one can ask if deep learning can be a more efficient and powerful alternative. To this end, we address the problem by using deep learning, as it stands as a powerful way to automatize and possibly overcome traditional methods. We use and compare three data representations, that is time series, images, and image time series of deformation, which account for the temporal, spatial, and spatio-temporal variability, respectively. We train our methods on synthetic data, since real datasets are still not enough to be effectively employed with deep learning, and we test on synthetic and real data as well, claiming that image time series and its associated deep learning model may be more effective toward the study of the slow deformation.

1. Introduction

Global Navigation Satellite System (GNSS) is one of the reference sources of information in geodesy. Geodetic data can help analyze the ground displacement with millimeter precision as well as monitor its evolution through time (Blewitt et al., 2018). Such data is commonly used to monitor the ground displacement as a response to environmental (e.g., tides, snowpacks, or hydrology), tectonic, or seismic forcing, and to characterize the mechanical response of the Earth to these forcings. Notably, GNSS data has been widely used to study the deformation associated with the different phases of the earthquake cycle. This leads to a better understanding of the loading of faults between earthquakes, of the seismic ruptures studied with either static or kinematic approaches, and of the processes driving the post-seismic relaxation (Bock & Melgar, 2016; Bürgmann, 2018, and references therein). In this study, we address the problem of seismic source characterization using deep learning from raw GNSS

Validation: David Marsan, Mathilde Radiguet, Anne Socquet
Visualization: Giuseppe Costantino
Writing – original draft: Giuseppe Costantino
Writing – review & editing: Giuseppe Costantino, Sophie Giffard-Roisin, David Marsan, Mathilde Radiguet, Mauro Dalla Mura, Anne Socquet

position time series. Seismic source characterization consists of the inversion of the location, magnitude, and depth (and focal mechanism when possible) of seismic events. This is usually performed by employing seismic recordings because the data has a high signal-to-noise ratio (SNR) and contains the information needed to estimate the earthquake parameters (Dziewonski et al., 1981). Thus, catalogs having low magnitudes of completeness can be obtained. Thus, catalogs having low magnitudes of completeness can be obtained. Because they directly measure the ground displacement associated with seismic events (and not the acceleration or the velocity) and because they do not saturate in near-field (unlike velocimeters), GNSS data provides interesting measurements in near-field that complement seismic data to constrain the earthquake source. Therefore, several studies carried out the source parameter characterization with GNSS data. They usually focus on one particular event or tectonic area, involving visual inspection of the data and dedicated modeling methods with a fine-tuning of the parameters (Blewitt et al., 2009; Feng et al., 2015; Guo et al., 2015; Lin et al., 2019; Page et al., 2009; Riquelme et al., 2016; Weston et al., 2012). Indeed, GNSS data presents several challenges associated with a lower SNR than seismological records. In other words, GNSS data has an intrinsic detection threshold, meaning that the signature of a seismic event (or any phenomena producing ground displacement) can be perceived in GNSS time series down to a certain magnitude, which is much higher than the completeness magnitude of catalogs obtained from seismic recordings. This makes it difficult to automatically estimate the seismic source parameters based on the ground displacement recorded on the surface. Here, we want to explore the effectiveness of deep learning methods to automatically characterize the seismic source. To this end, as the static deformation associated with regular earthquakes can be approximated with a similar simple dislocation model (Okada, 1985), we use GNSS data to characterize the static deformation signature of earthquakes. Catalogs listing the source of all M_w earthquakes are made available by the routine analysis of seismic recordings by seismological agencies, allowing for a benchmark with real GNSS data against an independent ground truth.

Machine learning and deep learning methodologies have recently been successfully applied to geosciences. In seismology, they have been used to address topics such as earthquake detection and phase selection resulting in seismic catalogs of unprecedented density (Kong et al., 2019; Mousavi et al., 2020; Ross et al., 2019; Seydoux et al., 2020; Zhu & Beroza, 2019; Zhu et al., 2019), earthquake early warning (X. Zhang et al., 2021; Münchmeyer et al., 2021; Saad et al., 2020), prediction of ground deformation (Kong et al., 2019; Mousavi et al., 2020), and earthquake magnitude estimation (Mousavi & Beroza, 2020; Münchmeyer et al., 2020; Saad et al., 2020). However, machine learning techniques applied to the analysis of geodetic time series are less numerous. Relevant applications in the frame of the analysis of the slow slip events have been presented by Rouet-Leduc et al. (2019, 2020), Hulbert et al. (2019, 2020), and He et al. (2020), with notable applications to InSAR data by Rouet-Leduc et al. (2021) and Anantrasirichai et al. (2019). As we can remark from the literature, seismic recordings are still the main source of information for the analysis of surface ground movements, linked to either slow or regular earthquakes. Thus, this is another motivation to explore the potential of machine learning to analyze GNSS times series. We want to explore and test recent developments in machine learning applied to time series or image analysis, to be able to mine the geodetic data and characterize the events with a physics-based approach.

In this paper, we address the problem of the fast seismic source characterization, that is, estimating the location and magnitude of a "regular" seismic event, based on deep learning applied to GNSS position time series. To the best of our knowledge, this is the first attempt at using machine learning-based techniques in such a direction. We solve our problem as a regression in the framework of supervised learning, meaning that the input data used during the training are labeled. The data ground truth comes from seismic catalogs, serving as a benchmark for our analyses. We explore three different ways to represent GNSS data (time series, images, image time series) taking into account both the spatial coherency and the temporal variability of GNSS data. We associate a customized deep learning model to each data representation either by re-adapting already existing methods or by designing it afresh. Training and testing of the different methods are first made on synthetics. The performance of our methods is then evaluated against real GNSS data using an independent benchmark coming from actual earthquake catalogs. The strengths and the pitfalls of the presented methods are discussed by envisioning some possible strategies to improve the results.

2. Methods

2.1. Background Work and Positioning

2.1.1. Machine Learning and Deep Learning Methods for the Seismic Source Characterization

In the frame of the source characterization, deep learning has proven to be particularly effective, as demonstrated by van den Ende and Ampuero (2020) and Münchmeyer et al. (2021), among the most recent works. As pointed out, a multi-station approach may more effectively locate the seismic source, in spite of other approaches using single-station waveforms, as Mousavi and Beroza (2020). Yet, combining observations from multiple stations is indeed a nontrivial task. van den Ende and Ampuero (2020) explicitly inject the location of each seismic station in form of latitude and longitude coordinates, while Münchmeyer et al. (2021) employ a sinusoidal embedding (i.e., the position is encoded through sinusoidal functions (Vaswani et al., 2017)) for the station locations, outperforming already existing methods and showing promising results in terms of earthquake early warning and source characterization. Nevertheless, as a general remark, no straightforward guideline is available to effectively take both the temporal and the network geometry into account at the same time. Exploiting the spatial distribution is indeed a key problem which we are willing to address in this work.

2.1.2. Followed Approach

An overview of the proposed methodology is shown in Figure 1. The employed pipeline consists of a training and an inference phase. During the training process, a model is provided with data to learn from. In the case of supervised learning, a couple ⟨ input, desired output ⟩ is presented to the model, which *learns* by minimizing a certain error metric between the estimated output and the desired output, which serves as a reference. We use the epicenter position (fault centroid) and the magnitude of the event as a target output for the characterization, with GNSS data as input. In the inference phase, the trained model is used to make predictions on new data. We will test our methods both against synthetic and real data. We provide new input data to the trained model and we compare the outcomes with the reference outputs, that is, the epicenter position and the event magnitude associated with this new input data. Training our models with supervised learning applied to earthquakes allows us to benefit from a benchmark coming from real earthquake catalogs. Here we do not estimate the hypocentral depth. When adding depth as a fourth parameter, the training process becomes less constrained, resulting in a degradation of the performance on the location and magnitude resolution. In the case of a thrust earthquake located on the subduction interface, the depth can be inferred from the earthquake position, knowing the geometry of the slab. Yet, in the case of a variable focal mechanism, the depth estimation is more complicated. Indeed, the typical wavelength of the surface deformation generated by an earthquake not only depends on the hypocenter depth but also on the focal mechanism (e.g., thrust generating a wider deformation field than strike-slip), making the depth more difficult to assess. In general, the more the parameters, the more the trade-off. This is why we decided to assess the location and magnitude only.

We make use of synthetic data to train and validate our deep learning models and we test on synthetic and real data afterward. Japan is probably one of the best-instrumented regions in the world, with GNSS data among the cleanest and the densest ones. Yet, we did not train our models with real data for the following two main reasons.

1. GNSS data suffers from the presence of data gaps and missing stations. They can be associated with station inactivity (e.g., electricity blackouts) or to inconsistent daily measurements, for example, due to large earthquakes. Moreover, the number of GNSS stations may evolve over time, due to the installation of new receivers or the temporary unavailability of certain ones. It can moreover make it hard to collect regular and well-formatted subsets of data to train on. This drastically reduces the number of exploitable training samples, which is indeed a key issue when training deep learning models (LeCun et al., 2015).
2. Real data is not uniformly distributed in terms of source parameters, most notably position and magnitude. Since we are dealing with subduction events, most of the actual epicenters will be located on the subduction interface. This can constitute a limitation since a deep learning model trained on such a configuration might not generalize well for events that would be located inshore or sufficiently far from the training area. In addition, the magnitude distribution follows the Gutenberg–Richter scaling law (Gutenberg, 1956). As a consequence, the deep learning methods would be biased because of the small magnitude events, which will be more numerous, thus possibly resulting in worse performance on the larger ones. To this end, we generate synthetic ruptures whose source parameters are assumed to be random variables drawn from a uniform distribution.

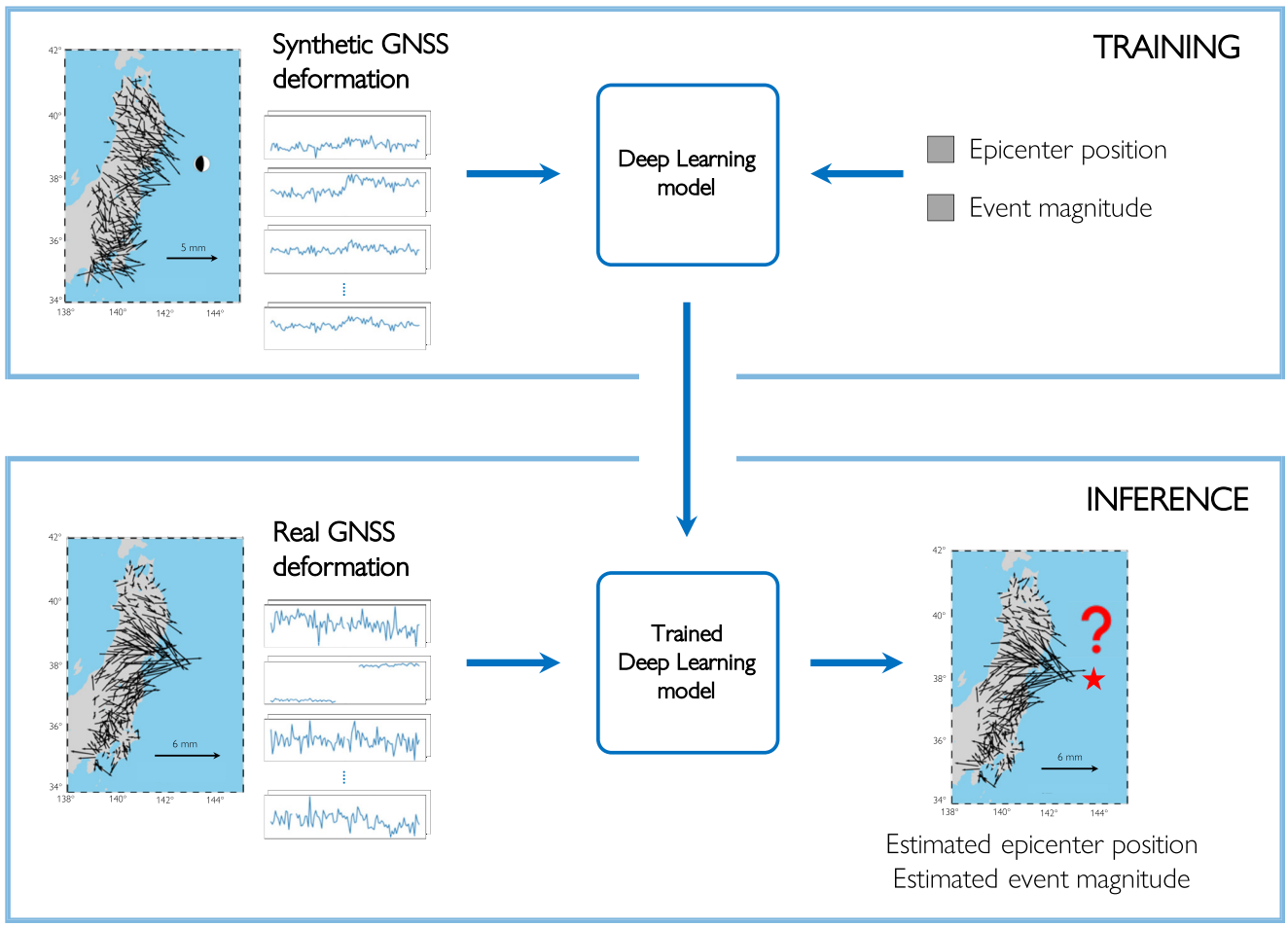

Figure 1. Schema of the proposed workflow, summarizing the training and the inference phases. A given deep learning model is trained by providing an input and a desired output. Here we use Global Navigation Satellite System (GNSS) data as input and a triple consisting of fault centroid latitude, fault centroid longitude, and magnitude as desired output for each event. During the training process, the model will learn a nonlinear function to map GNSS inputs to an approximation of its position and magnitude. Once trained, this model can be used to perform tests on new, independent data. Here we train on synthetic data and we test both on synthetic and real data.

By employing synthetic data, it is possible to generate as many samples as needed, overcoming the lack of data and exploiting the features of deep models. Nonetheless, the resemblance between the synthetic data and the real one plays a critical role, since it will have an impact on how well the deep learning model will perform on real data: we need to generate ultra-realistic time series. To this end, we add realistic noise computed from actual GNSS data, as it will be detailed in Section 2.2.

2.2. Generation and Representation of Synthetic Data

We generate synthetic data samples as the sum of a modeled displacement signal and a realistic noise sample. We rely on three data representations both for synthetic and for real samples and we associate each of them with a different deep learning model. More formally, the synthetic dataset is represented as a set of N couples $\{\mathbf{x}_n, \Theta_n\}_{n=1}^{N}$, with Θ a set of source parameters (epicenter position, magnitude, focal mechanism, etc.) and $\mathbf{x}$ being the data following an additive model:

$$\mathbf{x} = \mathbf{s} + \boldsymbol{\varepsilon} \tag{1}$$

with $\mathbf{s}$ the synthetic signal (cf. Section 2.2.1) and $\boldsymbol{\varepsilon}$ the noise term (cf. Section 2.2.2).

2.2.1. Synthetic Displacement

We obtain the synthetic displacement signals **s** by relying on Okada's dislocation model (Okada, 1985). The model input parameters are generated as follows. Earthquake hypocentral positions (longitude, latitude, depth) are assumed to be uniformly distributed random variables, with longitude $x \sim \mathcal{U}(139°, 146°)$, latitude $y \sim \mathcal{U}(35°, 41°)$, and depth $d \sim \mathcal{U}(2\text{ km}, 100\text{ km})$. Event magnitudes are generated as $m \sim \mathcal{U}(5.8, 8.5)$ and static moments M_0 are computed accordingly, as (Hanks & Kanamori, 1979):

$$M_0 = 10^{1.5m+9.1}\ \text{N} \cdot \text{m} \tag{2}$$

Fault azimuth direction ϕ_S (strike), dip angle δ, and slip angle λ (rake) are constrained to a thrust focal mechanism, by allowing for a certain variability of fault slip combinations: $\phi_S \sim \mathcal{U}(160°, 240°)$, $\delta \sim \mathcal{U}(20°, 30°)$, and $\lambda \sim \mathcal{U}(75°, 100°)$. Static stress drop $\Delta\sigma$ is assumed to be a lognormal random variable with an average value of 3 MPa and a standard deviation of $\pm$30 MPa. A circular crack is assumed with radius R computed as (Aki & Richards, 2002):

$$R = \left(\frac{7}{16}\frac{M_0}{\Delta\sigma}\right)^{1/3} \tag{3}$$

which can be used to approximate a rectangular dislocation, having length L and width W, by imposing the equality of the surfaces:

$$\pi R^2 = L \cdot W \tag{4}$$

The fault aspect ratio is assumed such that the fault length L and width W satisfy: $W = L/2$, with L computed as $L = \sqrt{2\pi}R$. It should be noticed that the dislocation surface does not change as a function of the aspect ratio between L and W. The average slip $\bar{u}$ is also derived for a circular crack and it is computed as (Aki & Richards, 2002):

$$\bar{u} = \frac{16}{7\pi}\frac{\Delta\sigma}{\mu} \tag{5}$$

with μ the shear modulus, assumed equal to 30 GPa.

Okada's dislocation model is applied to each one of this set of earthquake sources to compute the predicted synthetic displacement at each of the 300 GNSS stations in Honshu from the GNSS Earth Observation Network System in Japan (GEONET). Hence, the theoretical deformation field at all station locations in Honshu is obtained for each dislocation setting. Here, we decide to generate uniformly distributed epicenters having a thrust focal mechanism, instead of having them located on the subduction interface. Moreover, although we randomly generate events in a 3D space, we do not let them have a random focal mechanism. This choice is motivated by two reasons: (a) allowing for a complete variety of focal mechanisms would dramatically increase the size of the training database and (b) we can effectively assess the performance of the methods in locating the events, since no a priori has been made on the location, allowing us to more objectively test the performance of the models. This also allows the model to characterize events offset from the subduction interface (which is important when testing on real data, since the number of characterizable events is limited and includes both thrust subduction events and thrust crustal events, as we see in Figure 10).

It is worth remarking that the input of the deep learning models is driven by synthetic dislocations and the outputs are expressed in terms of point sources (cf. Section 2.1.2 and Figure 1). Here we estimate the magnitude of seismic events and their location, assumed as the position of the fault centroid. This does not represent an issue, since the approximation of the rupture as a dislocation in the synthetic database takes into account an extended fault, that allows a satisfactory first-order fit also for shallow sources beneath the GNSS network. Therefore, our approach should be able to characterize earthquakes even at shallow depths.

Finally, we also choose to fix the aspect ratio of the synthetic dislocations. Deep learning models are able to generalize the samples presented during the training phase through nonlinear combinations of the features computed within the network. Therefore, the combination of those features acts as if the aspect ratio was variable. Moreover, we are facing a problem that is naturally under-constrained: simplifying the choice of the parameters is a good trade-off between complexity and generalization ability on the training set.

2.2.2. Realistic Noise Computation

Noise in GNSS time series constitutes one of the most critical issues, as it is spatially and temporally correlated (Dong et al., 2002; Ji & Herring, 2013). Here we define noise as everything which is not the signal of interest, being the co-seismic signal offsets. At first approximation, its spectrum can be represented as a white noise at the lowest frequencies, and a colored noise having a $1/f^{\kappa}$ decay starting from a certain corner frequency, with the spectral index κ being usually fitted from the highest frequencies of the periodogram (Mao et al., 1999; Williams et al., 2004; J. Zhang et al., 1997). The spatial distribution of such a noise is not random. On one hand, some common patterns must be found among near stations; therefore it can be helpful to discriminate noise from other types of signals. On the other hand, making this type of analysis is difficult, because of the unpredictability of those spatial patterns as well as the intrinsic difficulty in handling such topological consistency in a consistent manner.

Realistic perturbations, that is, noise, are needed to mimic real displacement data. Here we rely on realistic noise samples computed from real GNSS time series by following an existing approach for surrogate data generation (Prichard & Theiler, 1994; Schreiber & Schmitz, 1996). By removing known signals (e.g., earthquakes, post-seismic relaxation, SSEs, jumps associated with antenna changes, etc.) from GNSS time series from a quadratic trajectory model (Marill et al., 2021), we obtain GNSS residual time series in the period 1997–2011 that contain the noise that we want to reproduce. Then, a Principal Component Analysis (PCA) is performed on 100-day windows, by taking into consideration all the stations at the same time. Afterward, a Fourier Transform (FT) is applied and the phase spectrum is randomized by picking a new phase $\varphi \sim \mathcal{U}(0, 2\pi)$. The same shuffling sequence is adopted for the whole network in order to preserve the spatial coherency between stations. After this process, an Inverse FT and an Inverse PCA (reconstruction by multiplication of each principal component by the corresponding transposed eigenvector) are performed. As a result, the transformed noise samples $\boldsymbol{\varepsilon}$ will have, on average, the same spatial covariance. Moreover, we can build new noise samples by randomizing the phase, since the Power Spectral Density of the transformed samples and the actual ones will be asymptotically equivalent. A schema of the artificial noise generation is given in Figure 2.

2.2.3. GNSS Data Representations

We build three data types: time series, images, and image time series. The raw data comes in the form of position time series. Then, we derive differential images to take the spatial information into account, and position image time series to take advantage of both the time and space patterns. A schematic view is provided in Figure 3. It might not be crucial to use time series (or image time series) to estimate co-seismic displacements. Nevertheless, the temporal information can help the method to better learn the noise structure in the time series, and therefore contribute to a better estimate of the co-seismic offset without having to preprocess the geodetic time series, which is one of the goals of automated methods.

Here we do not consider the vertical component of GNSS data because (a) it is noisier compared to the horizontal components and (b) we do not estimate the focal mechanism, for which the vertical displacement would be required.

Time series. We build synthetic position time series by considering a noise window of 100 days (cf. Section 2.2.2). We add a Heaviside step to simulate the co-seismic displacement (Bevis & Brown, 2014), with the onset time (cf. t_c in Figure 3) being at the center of the window. The step amplitude for each station depends on the modeled displacement (cf. Section 2.2.1). More formally, the time series structure is represented by a tensor $\mathbf{X} \in \mathbb{R}^{L \times T \times D}$, with L the number of stations and T the number of time steps, the location (latitude, longitude) of the station being given by $\mathbf{S} \in \mathbb{R}^{L \times D}$. D represents the number of components. In this study, $D = 2$ (N-S and E-W).

Differential images. Images of interpolated deformation field are computed as follows. By assuming the co-seismic onset at time t_c, we consider the difference between the displacement at time $t_c + 1$ day and $t_c - 1$ day, namely the differential co-seismic displacement field for each station in the GNSS network. We interpolate the deformation field in space as follows. We first employ a median anti-aliasing filter with a grid spacing of 25 arc min (≈45 km), then we interpolate the points in space by using adjustable tension continuous curvature splines (with tension factor $T = 0.25$) (Smith & Wessel, 1990). The resulting image dimensions are $76 \times 36 \times 2$ pixels. Afterward, we mask the sea by forcing to zero all the offshore pixels, in order not to extrapolate offshore, which may degrade the performance of the deep learning methods. Mathematically, the differential images are obtained by rasterizing for a given time step t_c an image as a tensor $\mathbf{D} \in \mathbb{R}^{I \times J \times 2}$ being $I \times J$ the resolution of the image $\mathbf{D}$

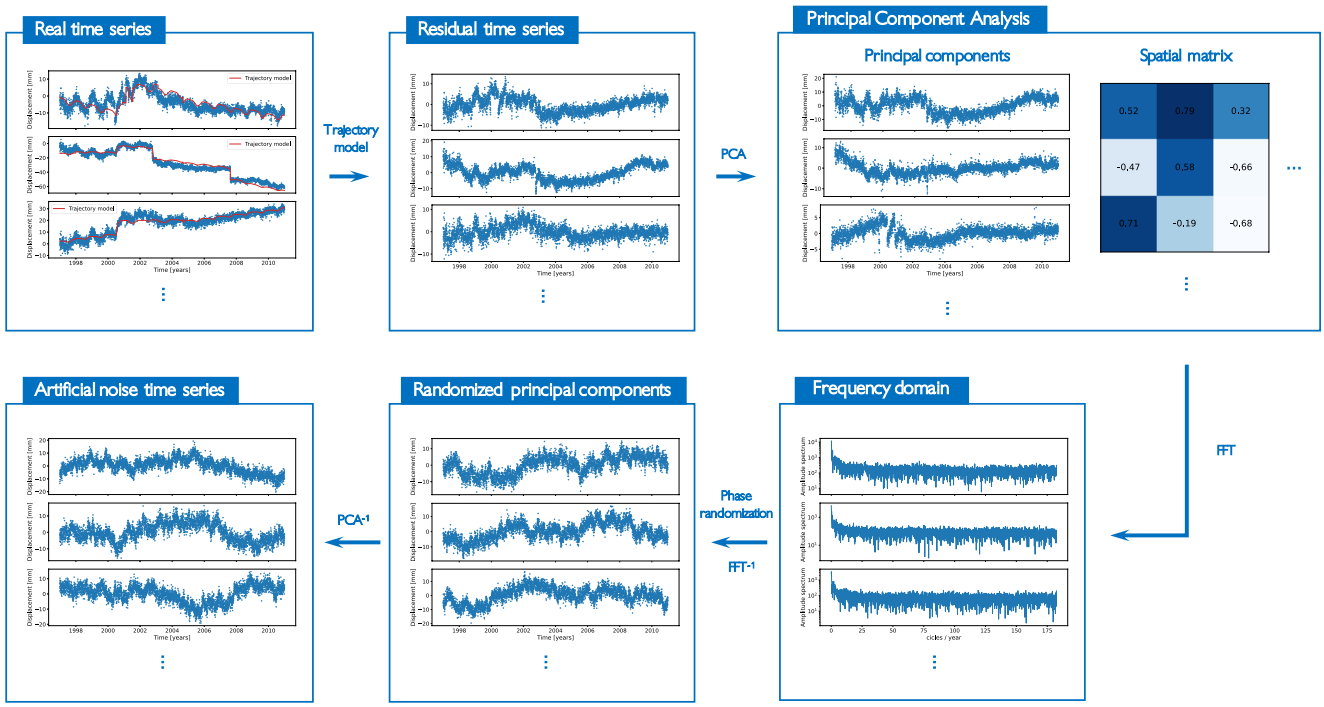

Figure 2. Schema of the artificial noise time series generation. Only 3 (notably stations 3032, 3033, and 3034) of the 300 GEONET Global Navigation Satellite System (GNSS) stations are shown for better visualization (N-S component). (1) Raw GNSS time series are scanned through the quadratic trajectory model from Marill et al. (2021). (2) Residual GNSS time series are obtained by subtracting the trajectory model obtained at the previous step. (3) A principal component analysis is performed in the spatial domain: 300 principal component time series are obtained by rotating the original data through a *spatial matrix*, whose columns are the eigenvectors of maximum–variance spatial directions. (4) The Fourier transform is applied to the principal components. (5) The phase of the spectra is randomized (cf. Section 2.2.2) and the randomized principal components are obtained via the inverse Fourier transform. (6) Finally, the artificial noise time series are obtained by projecting the randomized principal components back to the original space, via the transpose of the *spatial matrix*.

and $\mathbf{D}(\mathbf{S}(k)) = \mathbf{X}(k, t_c + 1) - \mathbf{X}(k, t_c - 1)$ with $\mathbf{S}(k)$ the position (latitude, longitude) of the kth station and t_c the time of the co-seismic offset. The value of I and J, as well as the content of the pixels $\mathbf{D}(\mathbf{S}(k))$, for $k \notin \mathbf{S}$, have been described before. The chosen image resolution (76×36) corresponds to a grid spacing of 5 arc min (about 9.3 km), which we found to be a good compromise between the deep learning network size and complexity (each pixel corresponds to a "neuron" of the deep network) and to the ability to capture small displacement variations in the spatial domain (a too small grid spacing would also introduce aliasing and higher noise variability into the image). Also, we chose to use adjustable tension continuous curvature splines (Smith & Wessel, 1990) because the obtained displacement fields are better suited to interpolate data from physical models (e.g., Okada's dislocation model) with respect to conventional bilinear or bicubic interpolation.

Image time series. Image time series are built from position time series by interpolating the position information at each frame with the same approach employed for the differential images. We consider 15 days of data, with the first seven frames corresponding to the week before the co-seismic displacement, the central frame corresponding to the co-seismic offset, and the remaining 7 days corresponding to the week after the co-seismic. Each frame of the image time series has dimensions $76 \times 36 \times 2$ pixels. Formally, an image time series is represented by tensor $\mathbf{T} \in \mathbb{R}^{M \times I \times J \times 2}$, with M the length of the image time series and $\mathbf{T}(t_i, \mathbf{S}(k)) = \mathbf{X}(k, t_c + i)$, $i \in \left(-\lfloor \frac{M}{2} \rfloor, \ldots, 0, \ldots, \lfloor \frac{M}{2} \rfloor\right)$.

In all three representations, we consider that the co-seismic offset time t_c is known.

2.3. Employed Deep Learning Methods

We developed a deep learning method specifically designed for the characteristics of each chosen data representation. We designed three methods by adapting different state-of-the-art methods that were not originally designed

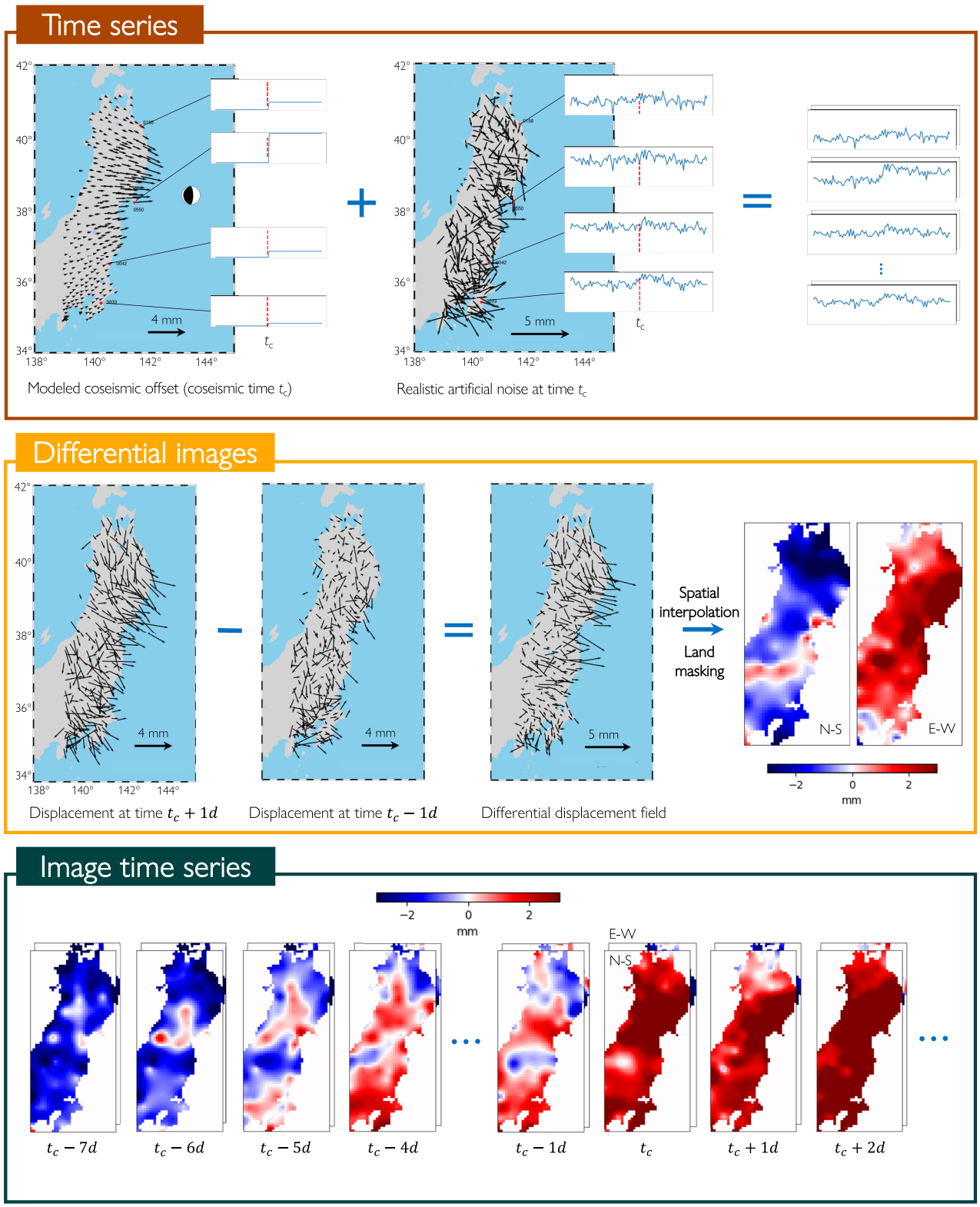

Figure 3.

for geodetic data, in order to best address our specific problem. A graphical outline of the methods is provided in Figure 4.

2.3.1. Time-Series Based CNN (TS)

Time series can be effectively processed by Convolutional Neural Networks (CNNs), extracting succinct information coming from temporal domain, as reviewed by Bergen et al. (2019) and Kong et al. (2019). Here we rely on the architecture proposed by van den Ende and Ampuero (2020), originally proposed for seismic data. A visual summary of the model is outlined in the first box of Figure 4. Their model has been selected as a potential candidate as it presents several interesting features that can be leveraged also when dealing with geodetic data. The first portion of their network consists of three convolutional blocks with an increasing number of feature maps. In each block, three convolutional layers are used for the feature extraction, followed by a max–pooling layer, employed for subsampling the data. Afterward, the coordinates of every station associated with an input waveform are injected into the model, as taking into account the location of seismic stations can improve the performance, which is the key characteristic of the model. The max–reduce strategy helps in aggregating the features related to the stations, in order to select the feature from the station corresponding to the most relevant contribution for the prediction. We adapt their model as follows. In order to further mitigate the vanishing gradient problem, the rectified linear unit (Agarap, 2018) activation function has been chosen for the hidden layers. The injected horizontal coordinates (latitude, longitude) of GNSS stations are previously scaled in [0,1]. The original model is also equipped with weights associated with the waveforms accounting for inactivity or missing data from a station. We set them to 1 as the GNSS network in Japan is quite dense and all the stations in synthetic data were assumed to be functioning. Yet, it can represent a further useful development, as it will make the model more flexible when testing on actual data as well as testing against other regions.

2.3.2. Image-Based CNN (IMG)

We use a 2D CNN to analyze and extract features from interpolated deformation images. They are an effective solution to leverage the spatial coherency and covariance of data structured as images (LeCun et al., 2015) and have become one of the reference architectures for image-based tasks (Goodfellow et al., 2016), also with relevant applications in the geosciences (Anantrasirichai et al., 2019; Rouet-Leduc et al., 2020).

A scheme of the architecture is provided in the second box of Figure 4. Here we rely on the architecture of MobileNetV2 (Sandler et al., 2018) as the feature extractor. This particular architecture has been chosen as it is lighter (in terms of the number of parameters) with respect to other state-of-art models, such as the VGG family (Simonyan & Zisserman, 2014). Yet, it presents some interesting features, such as the linear bottleneck layers and the depth-wise convolutions. The architecture presents a first convolutional layer followed by seven bottleneck layers. These layers perform an efficient convolution by relying on point-wise and depth-wise convolutions, presenting residual connections when there is not any stride in the convolutions. We use a global average pooling strategy after the feature extractor.

2.3.3. Image Time Series-Based Transformer (ITS)

Image time series-based approaches are required to account for both the spatial and the temporal variability in the input data. Deep sequence models such as LSTM (Long-Short Term Memory) or GRU (Gated Recurrent Unit) have been successfully used in geosciences to exploit the sequential behavior of the data (Bergen et al., 2019; Wang et al., 2017), as well as Transformers, which have overcome the former becoming the reference methods in the state-of-the-art (Mousavi et al., 2020; Münchmeyer et al., 2021; Vaswani et al., 2017). We tested both the LSTM and the Transformer approaches and we chose the latter, whose complexity is justified by its better ability to constrain the spatiotemporal evolution.

Figure 3. Outline of the three employed data representations. Each arrangement is designed for a specific deep learning model (cf. Figure 4 with corresponding colors). The data-arrangement procedure is shared between synthetic and real data, except for time series, which are directly available from Global Navigation Satellite System (GNSS) recordings. **(time series)** is associated with the TS model. Synthetic position time series are built by adding a modeled signal (cf. Section 2.2.1) to a realistic noise time series (cf. Section 2.2.2) by imposing the time of the co-seismic offset to be at the center of the window (cf. Section 2.2.3). **(differential images)** is associated with the IMG model. Differential images of ground deformation are built by differentiating the GNSS displacement on the day following and the day preceding the co-seismic time. Then, the differential deformation field is interpolated in space for each direction. **(image time series)** is associated with the ITS model. Image time series are the 3D-equivalent of position time series. A total of 15 days of deformation is collected, by selecting the week before and the week after the co-seismic offset (included). For each day, a spatial interpolation is performed by employing the same method as for differential images to produce a couple of images (N-S, E-W) representing a frame in the whole time series.

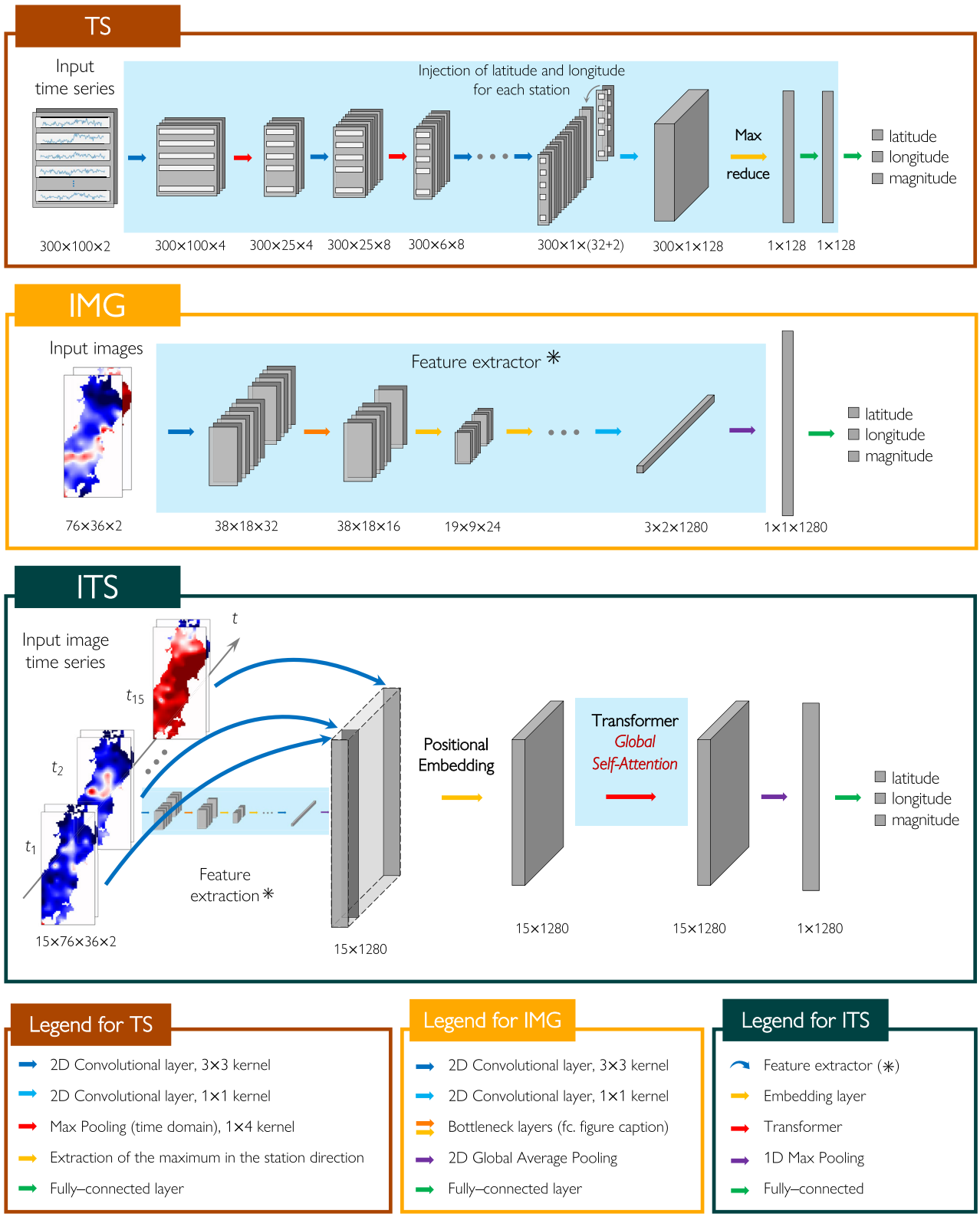

Figure 4.

The ITS architecture is presented in the third box of Figure 4. Here we design a relatively simple model to validate to consider both spatial and temporal features jointly, which can serve as a baseline to add more complexity in the future. We first use a feature extractor to compress the input data dimensionality to obtain a reduced representation. We use the same architecture of the IMG feature extractor and we distribute it in time, that is, we use the same feature extractor for each frame of the image time series. As a result, we obtain a feature vector for each frame of the image time series. Afterward, we stack all the feature vectors in one matrix to be exploited by the Transformer layer, as shown in the third box of Figure 4. Since the self-attention is, in general, order agnostic, we apply a Positional Embedding layer to ensure that the relative position of the frame information is correctly enforced (Chollet, 2021). We chose not to have a fixed mapping; therefore the embedding weights are learned during the training phase. After the embedding layer, we use a Transformer equipped with additive self-attention, as in Mousavi et al. (2020). For simplicity, we use only one global self-attention. According to our preliminary tests, the performance is not considerably increasing when adding a second level of attention, possibly because our model is still too simple to benefit from a hierarchical attention structure. After the self-attention, we apply another dropout (dropout rate 0.5) layer (cf. Section 2.3.1) followed by a one-dimensional Global Max Pooling. As a final remark, we train the model by enforcing the feature extractor to evolve from weights already learned by IMG. Therefore, we apply a sort of fine-tuning which may be beneficial for the self-attention to reach some acceptable parameter configurations in the early stage of the training already.

2.4. Implementation and Training Details

We enforced the mean squared error (squared $L2$ norm) as loss function, that is, the objective function which is minimized during the training, defined as follows:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\frac{1}{d}\sum_{i=1}^{N}\sum_{j=1}^{d}(y_{i,j} - \hat{y}_{i,j})^2 \tag{6}$$

where $\mathbf{y} \in \mathbb{R}^{N\times d}$ and $\hat{\mathbf{y}} \in \mathbb{R}^{N\times d}$ represent the ground truth and the predicted output, respectively, with N being the number of observations and d the number of dimensions. Notably, $d = 3$, being latitude, longitude, and magnitude the output variables. Hence, the loss function jointly minimizes the error on both position and magnitude. Since the ranges of the output variables are not comparable, they are first scaled in (0, 1). Thanks to this transformation, the high-range variables do not prevail on the others, possibly masking small variations on low-magnitude variables. As a result, the loss minimization turns out to be more regular and effective.

All three models have been provided with a last fully-connected layer with three outputs and a linear activation function (linear combination). Since the output variables are uniformly distributed, such an activation function would not squash the predictions in the boundaries of the output range, possibly making the model more flexible when predicting patterns laying outside of the ranges used in the training process. Thereafter, we enforce a dropout regularization (Srivastava et al., 2014) in this final layer (dropout rate 0.5) at training time, which helps prevent the models from overfitting the training data, in addition to the dropout regularization which may already be enforced throughout the previous layers.

We performed the training of the three models by adopting a mini-batch stochastic gradient learning (Bottou et al., 2018) with a batch size of 128 samples and the ADAM method (Kingma & Ba, 2014) for the optimization. The learning rate was chosen according to a grid-search optimization and the best value was found at 0.001. We initialize all the network weights with an orthogonal initializer (Saxe et al., 2013) for TS and with a uniform Xavier initializer (Glorot & Bengio, 2010) for IMG and ITS.

Figure 4. The three reference deep learning methods designed in this work. Shaded cyan rectangles represent existing state-of-the-art models. Such models have been slightly modified or adapted, where specified (cf. Section 2.3). Further details, such as dropout layers, stride, and activation functions, have not been depicted to facilitate the reading. Arrows represent the layers operating between the input (left) and the produced output (right). **(TS)** The network progressively computes features from convolutions and downsamplings in the time dimension. The latitude and longitude information is then injected. The resulting 2D-array is finally expanded and the contribution coming from the most informative Global Navigation Satellite System (GNSS) station is taken (*max–reduce* operation in yellow). Model readapted from van den Ende and Ampuero (2020). **(IMG)** is inspired to the MobileNetV2 architecture (Sandler et al., 2018). The input two-channel image is processed with convolutions and downsamplings by employing bottleneck layers (cf. Section 2.3.2) with and without residual connections (orange and yellow arrows, respectively). **(ITS)**. The first part of the network exploits the feature extractor of IMG to compute spatial features for each frame, which are packed in a 2D-array. Then, a positional embedding enforces time sequencing and prepares the intermediate-level data for the sequential analysis performed by the Transformer (self-attention as in Mousavi et al. (2020)).

Table 1
Position and Magnitude Error of the Tested Methods (Median ± Median Absolute Deviation) on the Synthetic Test Set

Model	Position error (km)	Magnitude error
TS	116.53 ± 67.58	0.25 ± 0.12
IMG	64.35 ± 44.84	0.11 ± 0.07
ITS	**52.65 ± 32.34**	**0.08 ± 0.05**

We employ twenty thousand synthetic samples that we divide it into training, validation, and test sets with proportions of 60%, 20%, and 20% respectively. We used the training and validation sets for the training phase. When the loss on the validation set is not decreasing anymore in a certain number of training steps, the training is terminated and the model's weights are loaded with the ones associated with the best loss value. Moreover, the validation set has been employed to tune the hyperparameters of the models (such as the learning rate, the best architecture, etc.) in order to prevent any overfitting. The test set is used for the final inference and for the performance analysis.

The code was implemented in Python using the Tensorflow (Abadi et al., 2016) library as well as the higher-level package Keras (Chollet et al., 2015). The training was run on NVIDIA Tesla V100 Graphics Processing Units (GPUs).

3. Results on Synthetic Data and Discussion

We first evaluate the performance of the three models on a synthetic test set, independent of the training and validation ones. In order to concretely compare the three methods, the synthetic and real datasets under consideration are the same for all the models and differ only in their input representation.

Table 1 shows the position and magnitude error in terms of median and median absolute deviation for the three models with respect to the synthetic test set.

The position error is assumed as the Euclidean distance and is computed for each sample as:

$$E_p^i = \sqrt{\left(X_i - \hat{X}_i\right)^2 + \left(Y_i - \hat{Y}_i\right)^2} \tag{7}$$

where X_i and Y_i represent the actual fault centroid longitude and latitude and $\hat{X}_i$ and $\hat{Y}_i$ the predicted fault centroid longitude and latitude, respectively. We adopt a Mean Absolute Error for the magnitude, which is computed for each sample as:

$$E_m^i = |m_i - \hat{m}_i| \tag{8}$$

where m_i and $\hat{m}_i$ are the actual and predicted magnitude, respectively. Then, the total position and magnitude errors are computed by averaging E_p^i and E_m^i.

The quantitative results evidence that the ITS method outperforms the other two, in terms of median error, both in location (52.65 km) and in magnitude (0.08), with a lower median absolute deviation in position (32.34 km) and magnitude (0.05).

3.1. Analysis of the Performance

Figure 5 shows the prediction of the three models on the synthetic test set, color-coded by the actual magnitude of the test events. The performance of all the models depends on the magnitude, which is closely related to the SNR. As we can observe in the third row, low magnitudes tend to be overestimated by all models, likely because of an intrinsic resolution threshold preventing the models from achieving good performance when the SNR is not sufficiently high. For the lower magnitude events (blue points), also the localization ability is poor, as the predictions of the three models do not follow, in general, the ideal prediction line. This behavior may thus be linked to an intrinsic limitation of data information.

The solid black lines in the plots show the rolling median on the scatter plot computed on 150 samples, providing the general trend of the predictions. The median prediction of IMG and ITS models better follows the perfect prediction line, with respect to the TS model. Also, the median prediction of ITS is more precise than the IMG model also for magnitudes in the range (6.2, 7). The trend of the magnitude prediction for TS deviates from the ideal prediction line both for small and for large magnitudes, presenting a median saturation around M_w 6.3 and an offset for $M_w > 6.4$, respectively (cf. black solid lines). The sharp saturation for low magnitudes could be due

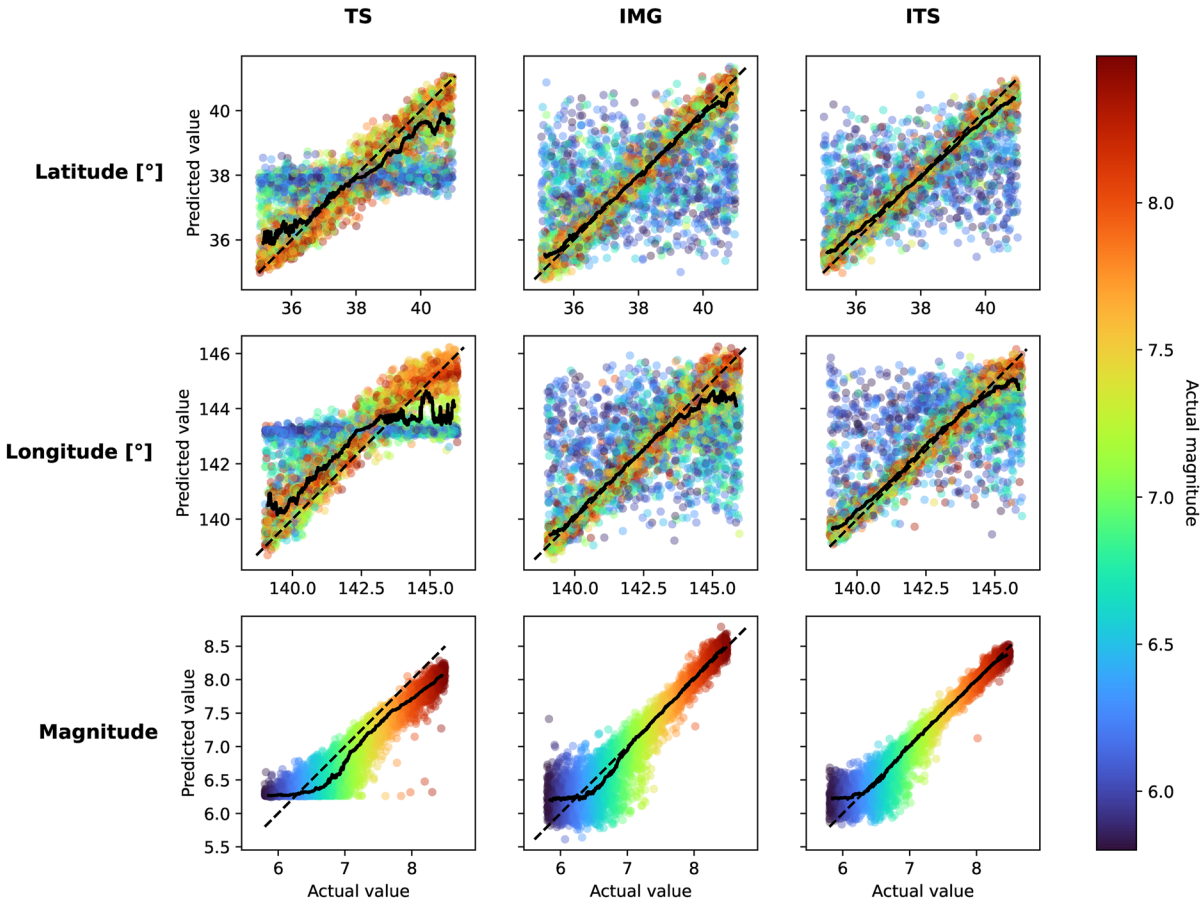

Figure 5. Comparison of the performance of the tested models at inference time. TS, IMG, and ITS models are shown on columns respectively. For each row, latitude, longitude, and magnitude predictions are reported, respectively. Each point of the scatter plots represents a test sample, whose magnitude is indicated by the color bar, and it is illustrated as a function of both its actual and predicted value. Black dashed lines represent the ideal prediction, while solid black lines represent the rolling median.

to the employed network architecture as well as to specific features associated with the type of data. The magnitude prediction for the image-based methods, that is, IMG and ITS, better adhere to the ideal prediction line, with a progressively smaller error variance at larger magnitudes, in line with the SNR improvement. As for the magnitude resolution, ITS is the method associated with smaller error variances and with a better median trend. Also, in the case of the ITS model, the median prediction trend can be used to individuate a tentative magnitude threshold value, that is the value under which the magnitude prediction is significantly degraded, corresponding to the magnitude at which the perfect prediction line significantly deviates from the rolling median (cf. dashed and solid lines in Figure 5). We can derive a resolution limit for the ITS model: $M_w^{ITS} = 6.2$.

From the latitude and longitude prediction, that is, the localization performance, we can observe that the models do not treat similarly the low and high magnitudes. Notably, for magnitudes smaller than the SNR limit, TS assigns them an average position (i.e., near 38 for the latitude and 143.5 for the longitude). This behavior is clearly indicated by the horizontally-clustered blue points. This pattern is indeed coherent with the choice of the quadratic loss function used to train the model. In fact, at first order, the best guess is represented by the mean value of the output range subject to the posterior distribution (Haykin, 2008; Moon & Stirling, 2000). We can derive that, when the SNR is below a certain resolution threshold, the model associates low-magnitude events to average coordinates, which likely minimize the average error. For higher magnitudes, the TS latitude predictions are more clustered around the ideal prediction line. Yet, a tendency toward the mean values is still present, because TS predicts the longitude of high-magnitude events either in the proximity of the GNSS network (longitudes less than ∼142) or in far-field (longitudes higher than ∼144). Conversely, image-based methods (IMG and ITS) characterize low-magnitude events as having a random position in the region of interest (cf. scattered blue points),

while being able to precisely constrain higher-magnitude events, with predictions tightly clustered around the ideal line. The median prediction lines for IMG and ITS are more stable with respect to TS and significantly bend only at the highest longitudes (i.e., for earthquakes located offshore close to the trench, far from the measurement network located inland Japan), that is the models have the tendency to underpredict the longitude. This is a feature that is a known bias when we study offshore earthquakes with geodesy, and this is due to the geometry of the measurement network. If the displacement was evenly sampled spatially, the bias would disappear.

3.2. Spatial Variability of the Location Error

Figure 6 shows the location error as a function of the ground truth spatial coordinates. The plot has been computed by interpolating the location error for each test data sample onto a grid, corresponding to the area of interest. This smoothed heatmap indicates the amount and the distribution of location errors all over the tested region, for different magnitude ranges. This type of representation can help assess the physical consistency of the tested models, as well as reveal systematic biases in the error pattern for test events as a function of the magnitude and their relative position with respect to the GNSS network.

The heatmaps of the first two lines, corresponding to the magnitude ranges (5.8, 6.3) and (6.3, 6.8), show how the three methods handle the characterization of low-magnitude events (cf. 3). The ITS model is able to better resolve small magnitude events in the near field (i.e., in the proximity of the GNSS network). When the magnitude increases, the error amplitude of IMG and ITS decreases, affecting only the points which are far from the network (eastwards). For high magnitudes, TS tends to localize many events in far field, with a higher average error with respect to IMG and ITS.

The error pattern for the image-based methods is more physically consistent. The most reasonable explanation is that image-based models can better capture spatial information by extracting spatial features which are essential for the characterization. As a general comment, we do not see any clear bias and the error patterns exhibit correct behavior, since, as the magnitude increases, the highest errors are pushed toward the far field. For low magnitudes, the maximum error associated with the TS is about 200 km less than the other models: the bias in the TS predictions correctly minimizes the average error, yet without providing any discriminant ability to the model. By increasing the magnitude, errors become smaller and smaller, with the events contributing to the largest errors being distributed on the eastern (offshore) side, in favor of ITS, which is associated with the most reasonable error pattern.

The roughness of the spatial error distribution observed in Figure 6 may be due to the noise realization or to the source parameters of the tested events. In general, each deep learning model is associated with a characterization limit, depending on the magnitude, location, and depth, at first order. Because of this threshold, small-magnitude deep events are not well characterized, regardless of their location with respect to the GNSS network (cf. Figures 5 and 9). Since the event magnitude and location in the evaluation set are uniformly distributed, poorly characterized samples are equally spread in the tested area, making the spatial interpolation non-smooth.

3.3. Influence of the Distance From the GNSS Network on the Predictions

Figure 7 shows the dependency of errors of events based on the relative position with respect to the GNSS network. Each scatter plot represents the error as a function of the distance to the nearest GNSS station. Such a distance is computed from the coordinate of a hypocenter as the 3D Euclidean norm, in order not to take into account the Earth curvature. This kind of representation is effective in revealing patterns of the position and magnitude errors as a function of both distance, on the x axis, and magnitude, in color code. We identify three regions, according to the relative distance to the nearest station: being d the distance to the nearest station, we will refer to near, intermediate, and far field when $d \leq 0.5°$ ($\approx$55 km), $0.5°$ ($\approx$55 km) $\leq d \leq 3°$ ($\approx$334 km), and $d \geq 3°$ ($\approx$334 km), respectively (see dashed lines in Figure 7). The solid lines correspond to the median for several magnitude ranges (cf. Figure 6).

The TS model is associated with higher position errors for events located both in near and intermediate field, while image-based methods can correctly locate a larger number of high and even low-magnitude events. TS is also associated with higher magnitude errors for high magnitude events, clustered around a value of 0.4, showing that not even the events occurring in the proximity of the GNSS stations are well retrieved. Conversely,

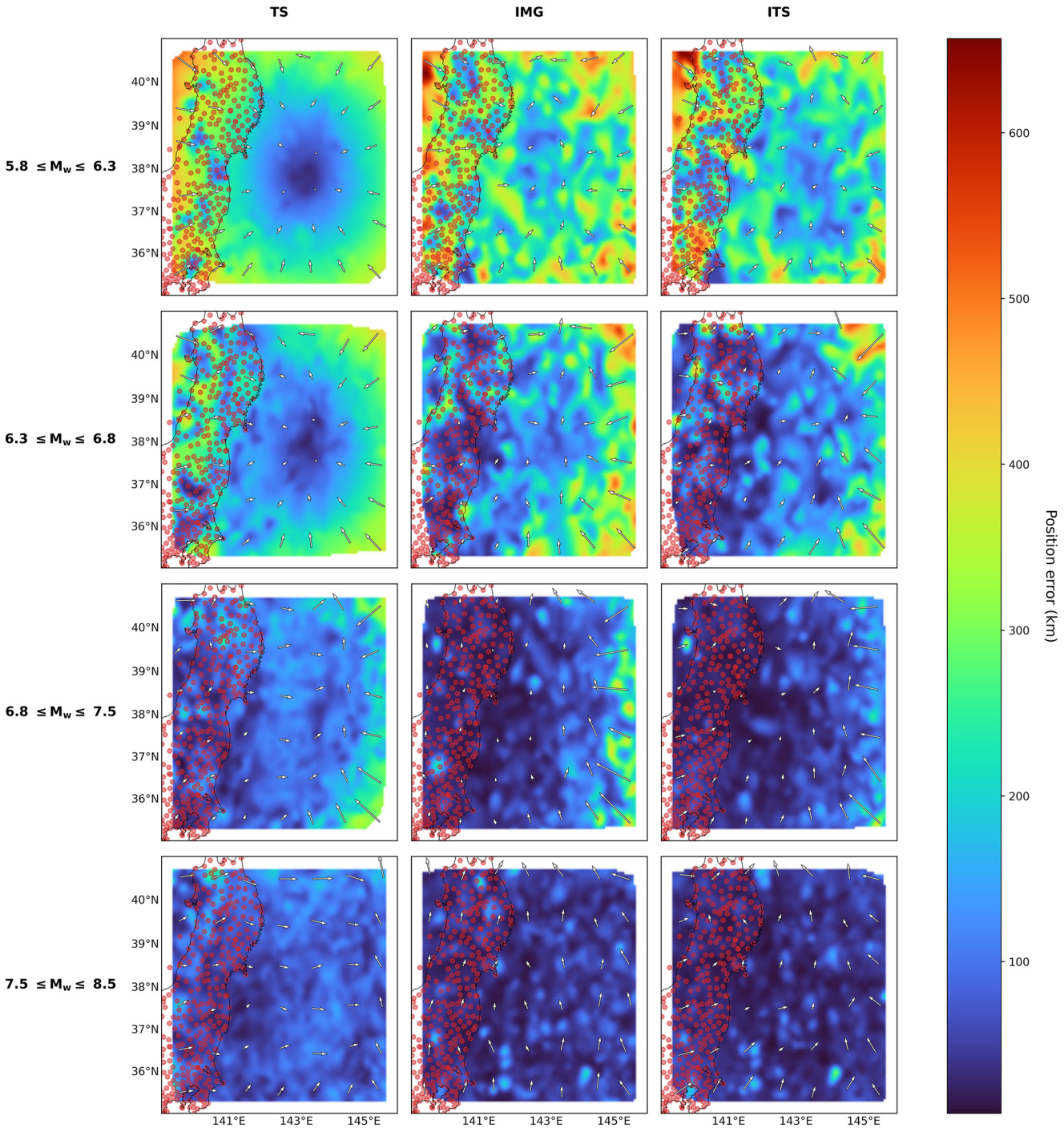

Figure 6. Comparison of the location error of the tested models, reported in the columns. Each subplot shows the location error associated with the test samples, interpolated on a grid whose corresponding spatial coordinates are indicated along the axes. Magenta data points represent the position of Global Navigation Satellite System (GNSS) stations in Japan. The heatmap depicts the distribution of the error in position committed by the tested models, for different magnitude ranges, in rows. Arrows show the average direction of position error for patches of 1 × 1 arc degree. The arrows have the same scale throughout all the subplots, making a comparison possible among different models.

image-based methods are more accurate in the magnitude estimation, with a less biased error pattern. The median curves of errors increase with the distance, both for the magnitude and the position estimation. This may not happen in correspondence with the smallest and largest magnitude bins because the models have rather a random behavior at small magnitudes or because the error at high magnitudes does not depend on the distance to the

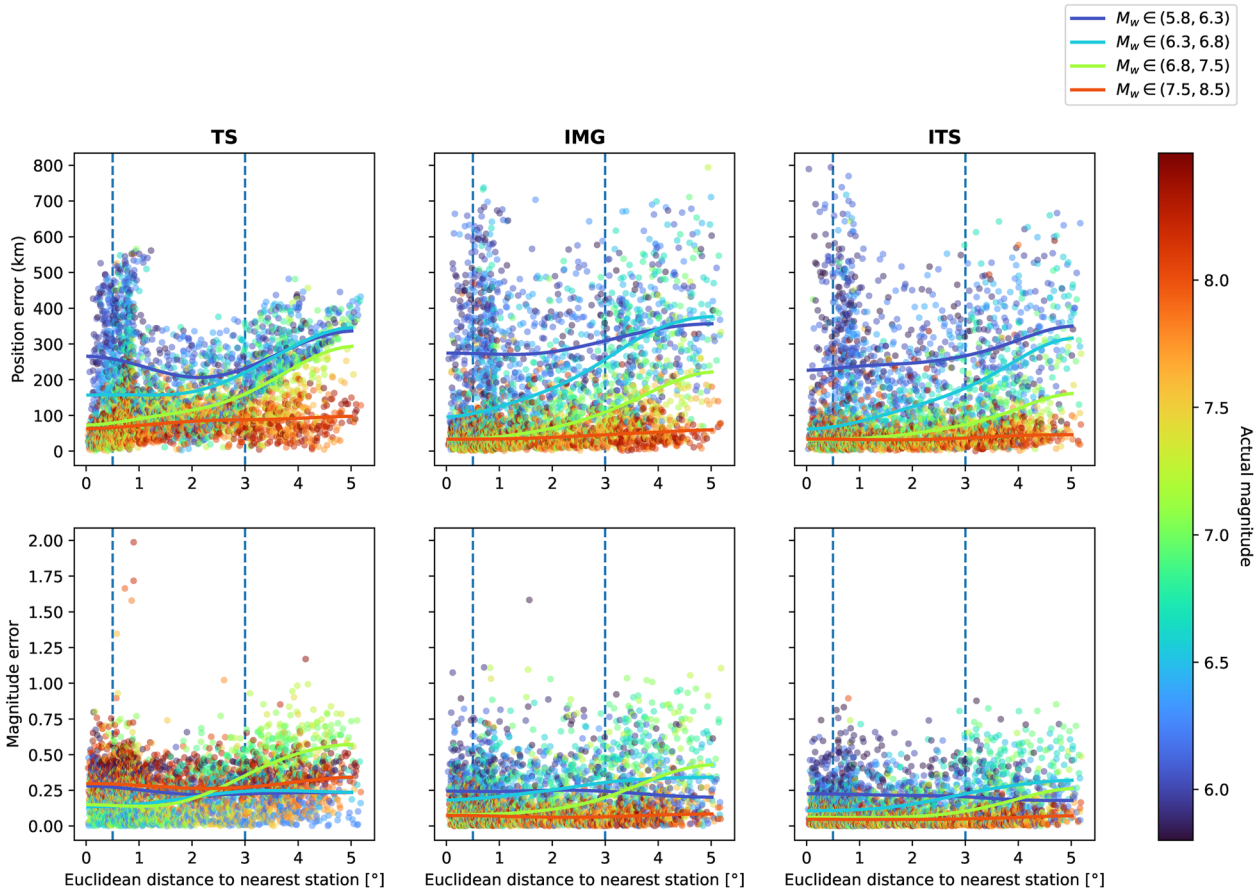

Figure 7. Comparison of errors as a function of the distance to the nearest Global Navigation Satellite System (GNSS) station. The deep learning models are shown in columns, while the rows indicate position and magnitude errors, respectively. Each scatter plot depicts errors as a function of the Euclidean distance to the nearest GNSS station, expressed in arc degrees. Each data point, representing the position error and the absolute magnitude error between the test samples and the model predictions, is color-coded by the actual magnitude of the event. Solid lines represent the median of subsets of the data points, filtered by magnitude ranges as indicated in the legend in the top right. Vertical dashed lines discriminate among near, intermediate, and far field, respectively.

GNSS network anymore. Since the depth has been taken into account when computing the distance to the nearest GNSS station, the offset in the large magnitude prediction associated with TS (cf. Figure 5) affects very shallow and near events, leading to the conclusion that image-based data representation can bring more exploitable information about the deformation field. Therefore, more low-magnitude events are captured.

3.4. Magnitude Threshold Estimation From ITS Localization Error

Since we will test the deep learning models on real data (cf. Section 4), we define here a criterion to assess whether a characterization coming from a learning model is reliable. Figure 8 shows the position error for the ITS method, computed for each test sample, as a function of the magnitude, with each subplot corresponding to a different range of hypocenter–station distances and hypocentral depths. The general idea is to get an estimation of the magnitude limit (the magnitude down to which the models provide acceptable estimations) for different settings, that is, for different values of depth and distance to the GNSS network. Based on the magnitude limit, the event location, and depth, we assign it a hard threshold (*characterizable*, *non-characterizable*) that can be used, when testing on real data, to filter out all those events which would necessarily be incorrectly characterized. We draw those measures from the method which best performed on synthetic data, namely ITS, and we will use them for all three methods.

As discussed in previous sections, as the depth increases, the magnitude detection limit also increases. For near field events having a depth $d \leq 30$ km, we can set a magnitude threshold at M_w 6, by selecting a limit where the error is reasonably low with respect to the general trend. As for intermediate and far field, it is harder to assign

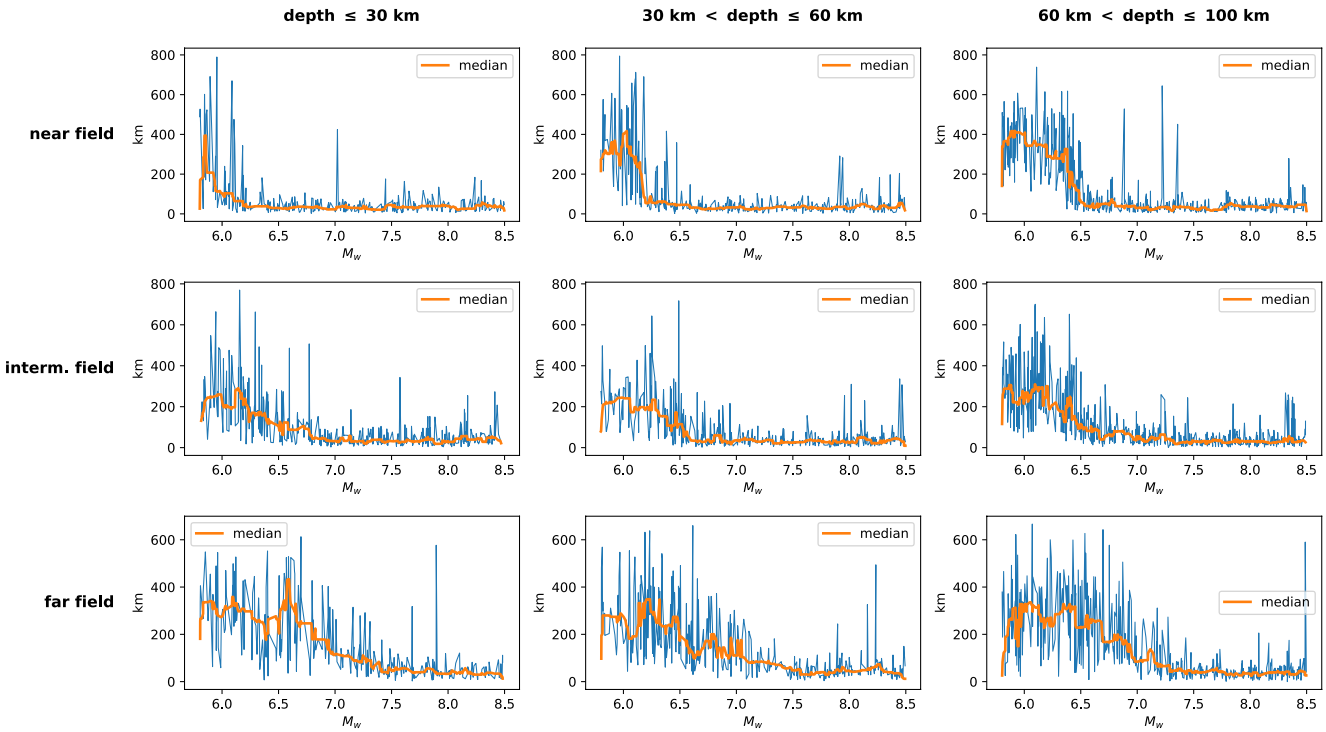

Figure 8. Position error, computed for each test sample, as a function of the magnitude (x axis), the depth range (columns), and the distance range (cf. Figure 7) with respect to the Global Navigation Satellite System (GNSS) network (rows) for ITS. The orange solid line represents the result of a median smoothing by employing a kernel size of 15 points.

a magnitude threshold value, as the interplay between magnitude, distance, and depth is generally nonlinear. However, a general tendency can be still observed. The estimated thresholds are be above M_w 6.8 and 7.5 for intermediate and far fields, respectively. The values for every chosen combination of depth and distances are resumed in Table 2 and will be used in Section 4 when testing the deep learning models on real data. We should also consider that, to parity of depth range, the relative distance between the event and the GNSS network strongly affects the probability of correct retrieval, making the magnitude threshold larger and larger. This poses some limitations in the characterization of deep and far offshore events, with only large magnitude earthquakes being characterizable in those conditions.

3.5. Interplay Between Depth and Magnitude in the Magnitude Resolution

Figure 9 shows the cumulative histograms of the difference between the actual and predicted magnitude as a function of the distance ranges to the GNSS network (rows) for the three deep learning models (columns), color-coded by (actual) magnitude ranges (see Figure 6). The residuals of the TS predictions are not centered around zero for any magnitude subsets. For low magnitudes (blue bin), the actual magnitude is overpredicted for a non-negligible fraction of the test samples, which corresponds to the sharp lower limit for the TS magnitude estimation and to the bias in the location estimation (see Figure 5). As the magnitude increases, the residuals become positive and an underestimation of the magnitude is observed. When the events are located farther with respect to the GNSS

Table 2
Magnitude Thresholds of ITS Estimated Against the Synthetic Test Set

	Depth ≤ 30 km	30 km < depth ≤ 60 km	60 km < depth ≤ 100 km
Near field	6	6.2	6.5
Intermediate field	6.8	6.8	7
Far field	7.5	7.5	7.8

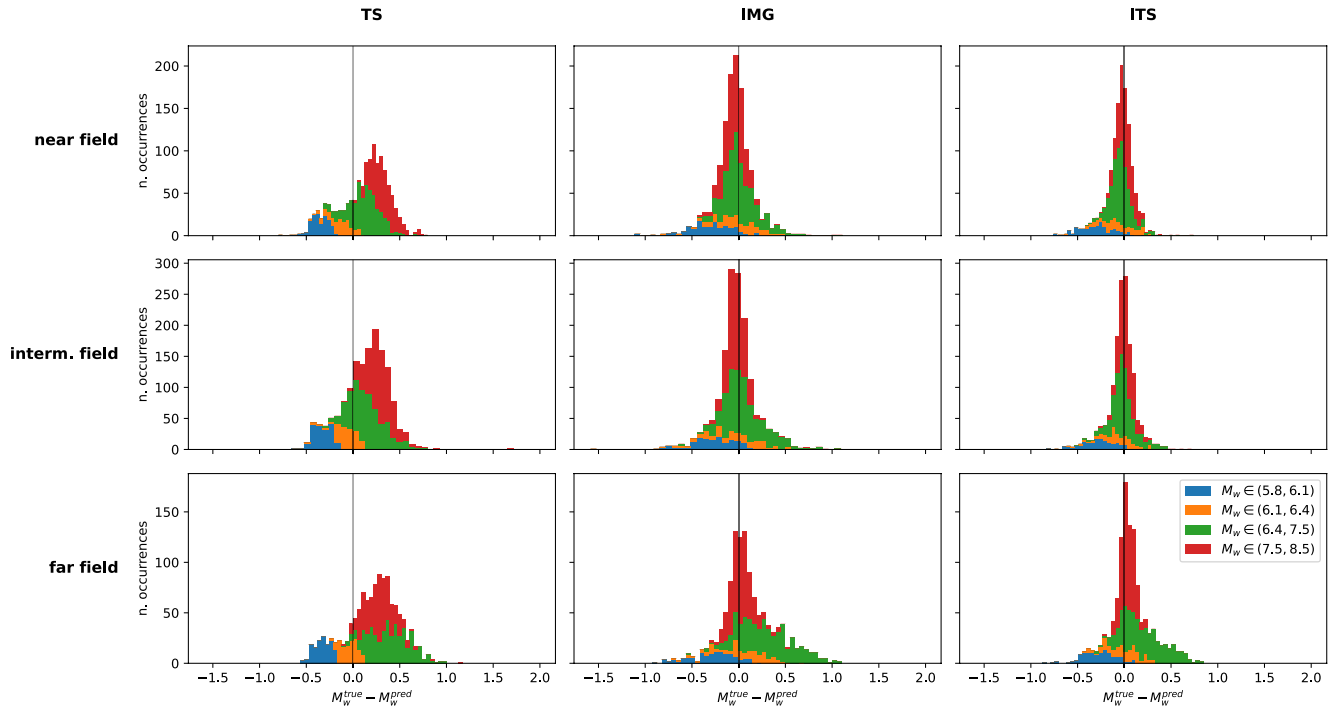

Figure 9. Cumulative histograms of the magnitude difference $M_w^{true} - M_w^{pred}$ for the three models (columns) as a function of the distance range (cf. Figure 7) with respect to the Global Navigation Satellite System (GNSS) network (rows) and for different magnitude ranges (color). Vertical lines individuate the zero.

network, the trend does not change, which means that the TS model cannot take advantage of the GNSS network to ameliorate the prediction even for events located in near field.

The residuals of the IMG and ITS models are generally centered around zero. Low-magnitude events (blue bins) are overestimated and are more dispersed with respect to TS. For magnitudes greater than 6.4, IMG and ITS behave almost the same in near and intermediate field. In far field, ITS performs better in the magnitude range (6.4, 8.5), being able to well retrieve the events that are underestimated by IMG (see the median prediction line for IMG compared to ITS in Figure 5). Those events could likely be the deepest ones. In fact, IMG underestimates their magnitude, which affects also the localization performance, especially the longitude estimation (cf. Figures 5 and 6). ITS shows a better performance on those events, suggesting that it may have better constrained the trade-off between depth and magnitude in far field. The TS model does not well resolve the ambiguities coming from the interplay between magnitude, position, and depth. Thus, arranging the GNSS data into differential images and image time series might help better estimate deeper and deeper events to parity of location and magnitude, without any prior on the depth itself.

3.6. Testing of the Models on Data Affected by a Post-Seismic Signal

The data that we use to train the deep learning models does not contain any post-seismic signal following the co-seismic rupture. Nevertheless, we test the models on real data, which is affected by the post-seismic relaxation signals. Therefore, we generate a synthetic test set made of noise, co-seismic signals, and post-seismic signals. We model the post-seismic signals $p_s(t)$ at each station s as:

$$p_s(t) = H \log_{10}\left(1 + \frac{t - t_c}{\tau}\right) \tag{9}$$

where t_c is the time of the earthquake (co-seismic). The amplitude H is assumed as a uniform random variable:

$$H \sim \mathcal{U}(0.5c, 1.5c) \tag{10}$$

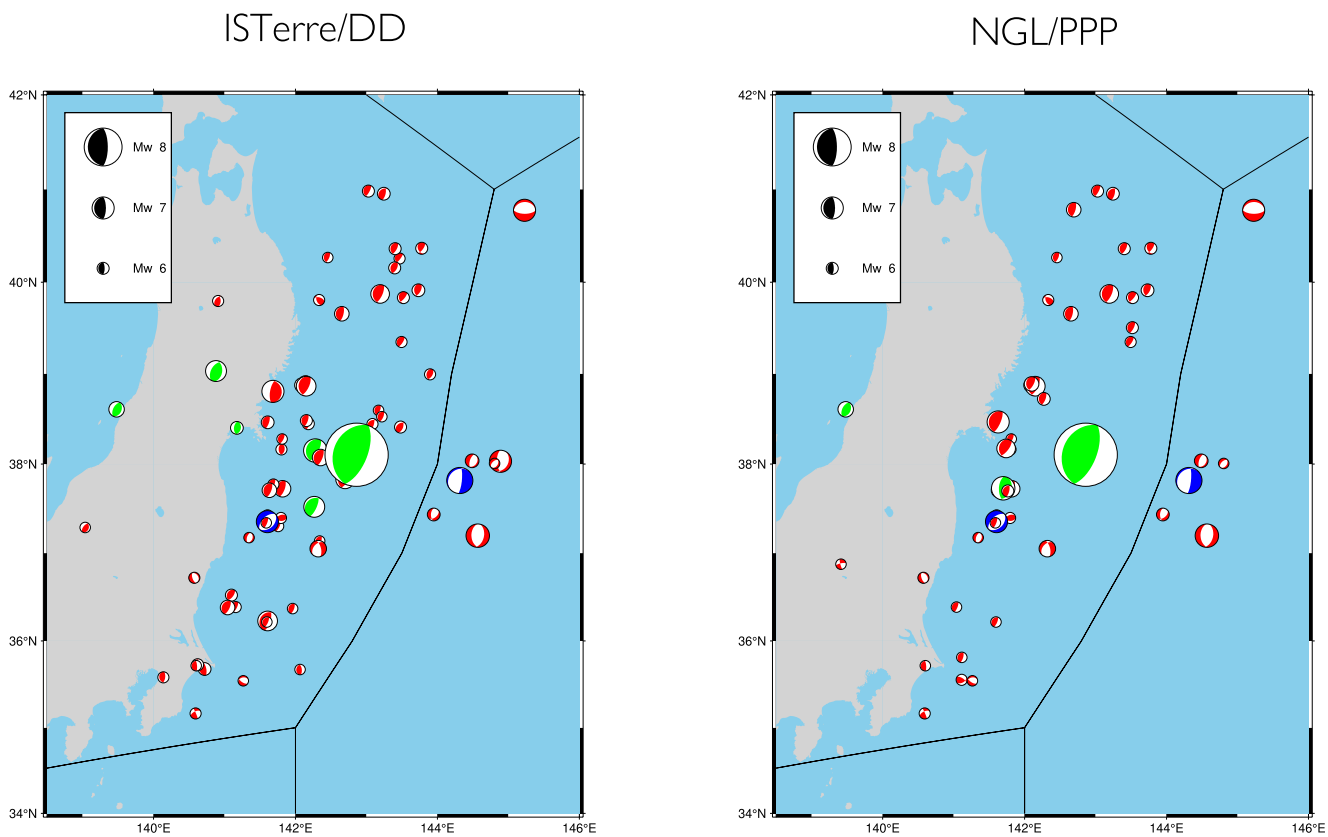

Figure 10. Seismic catalog associated with the ISTerre/DD and NGL/PPP datasets, respectively. ISTerre/DD dataset contains 69 events ranging from 1998 to 2019, while NGL/PPP set contains 51 events ranging from 2009 to 2021. Focal mechanisms are shown for each event and their size is proportional to the magnitude, according to the legend at the top left. Based on the "characterizability" thresholds from Table 2, focal mechanisms are colored as follows: green indicates the characterizable events having a thrust focal mechanism, blue for the characterizable events having any focal mechanism but thrust, while red indicates all the events that cannot be characterized.

where c is the amplitude of the synthetic co-seismic displacement at the station s. The relaxation time is assumed as a uniform random variable as well:

$$\tau \sim \mathcal{U}(1, 50)\,\text{days}. \tag{11}$$

The results are shown in Figures S1–S5 in Supporting Information S1. The performance of the models degrades when testing (without retraining) on data having a quite strong post-seismic signal following the co-seismic offset. The magnitude and latitude predictions do not significantly differ from the ones in Figure 5, while the longitude estimation is degraded, for all the models. This result is not surprising, since the data is different from the training one and since the longitude was already a difficult parameter to retrieve, notably for events occurring offshore. In fact, the time series and the image time series may vary significantly, while differential images contain a larger co-seismic displacement value for small values of τ.

4. Application to Real GNSS Data

After testing the models and extracting some statistics on synthetic data, we use the trained models to make further inference on real data. For each cataloged earthquake, we run our processing pipeline (details in Section 4.1) to obtain the estimated fault centroid location and magnitude, which we compare with the cataloged ones, in order to analyze the performance of the tested deep learning models in presence of raw noisy GNSS data.

Table 3
Position and Magnitude Error of the Tested Methods on the Characterizable Events Belonging to the Real Data Sets (Median ± Median Absolute Deviation) Having Thrust Focal Mechanism (cf. Green Events in Figure 10)

	ISTerre/DD		NGL/PPP	
Model	Position error (km)	Magnitude error	Position error (km)	Magnitude error
TS	137.61 ± 75.78	0.47 ± 0.09	175.51 ± 32.62	**0.11 ± 0.08**
IMG	**52.47 ± 21.82**	**0.13 ± 0.09**	140.98 ± 61.49	0.63 ± 0.17
ITS	76.73 ± 21.62	0.25 ± 0.14	**101.92 ± 10.04**	0.45 ± 0.16

Note. These results should be taken carefully, since they have been obtained on a very limited number of samples, notably five for ISTerre/DD and three for NGL/PPP.

4.1. Data Processing

The seismic catalog selection for real events in Japan has been conducted as follows. The F-Net catalog from NIED (cf. https://www.fnet.bosai.go.jp) has been exploited and events ranging from 1998 to 2021 have been selected according to the studied range of characteristics (epicentral position, hypocentral depth, magnitude, see Section 2.2.1) for a total of 174 events. Magnitudes have been allowed to exceed the 8.5 limit in order to further test the models on high-magnitude events, even though it is out of the training range. Since our approach can deal with only one event per day, if more than one event is recorded in the same date, only the maximum magnitude event is kept and therefore estimated. Also, consecutive events have been discarded in order to provide the correct static offset values to the IMG model. All events in 2011 have been removed except the Tohoku event (11 March 2011). Indeed, the earthquake and subsequent tsunami damaged several GPS stations, and the time series of the remaining ones are dominated by a strong post-seismic relaxation effect making GNSS time series difficult to interpolate and interpret on an automated manner. The magnitude of the Tohoku-Oki earthquake was estimated by NIED as M_w 8.7. However, Lay (2018) show that its actual value is rather 9.1. For this reason, we replaced the cataloged magnitude with the correct value. After this pre-processing, 84 events are present in the catalog.

Two GNSS datasets have been collected: the data processed in double difference at ISTerre (Institut des Sciences de la Terre) that range from 1998 to 2019 (gnss products, 2019; Marill et al., 2021) and the data processed in PPP at NGL (Nevada Geodetic Laboratory) (Blewitt et al., 2018) that range from 2009 to 2021. We performed outlier detection and removal by processing the data with the *hampel filter* (Pearson et al., 2016) with a window length $n = 3$. Thereafter, we extracted, for each date in the seismic catalog, a window of 100 days, centered onto the co-seismic offset (cf. Section 2.2). We considered a 100-days stack of time series as valid if at least 60% of the stations are present (i.e., ∼180) and if at least the 70% of the median number of data points in the 100-days window (i.e., 70) is not undefined (i.e., less than 30% of data gaps). The remaining data gaps are filled as follows. After centering the time window on the co-seismic offset date, we compute the linear trend in the first and the second half. Thanks to this procedure, an approximation is provided for the small data gaps and also a first order reconstruction of the co-seismic offset when that information may be missing. Finally, the data is detrended, that is, the linear trend is subtracted for every 100-days stack.

After the previous processing, the ISTerre/DD and the NGL/PPP datasets contain 69 and 51 labeled time series. We used the magnitude thresholds obtained for ITS (cf. Table 2) to differentiate the theoretically-characterizable events from the rest, as shown in Figure 10, that is if magnitude, depth, and position of the events are such that they satisfy the experimentally-derived relationships detailed in Table 2. We found eight (three of which are thrust subduction events and three are thrust crustal events) and five (two of which are thrust subduction events and one is a thrust crustal event) characterizable events for ISTerre/DD and NGL/PPP datasets, respectively. The data is further rearranged into differential images and image time series and the performance of the three deep learning methods is evaluated. Since the deep learning models have been trained to assign a point source (expressed in terms of synthetic fault centroid) to finite fault dislocations, the latitude and longitude of the cataloged epicenters cannot be used as a benchmark. Thus, we use the coordinates of the centroid for each of the real characterizable events from the Global Centroid-Moment-Tensor (CMT) catalog (https://www.globalcmt.org/) (Dziewonski et al., 1981; Ekström et al., 2012).

4.2. Results and Discussion

The quantitative results are shown in Table 3, while Figure 11 shows the performance of the tested methods on the two real datasets. The displacement fields associated with all the characterizable events in the ISTerre/DD dataset are represented in Figure 12.

From the numerical results listed in Table 3, associated with the "characterizable" events having a thrust focal mechanism, it is hard to assess the best method because they have been obtained on an insufficient number of samples. The IMG model performs better on the ISTerre/DD dataset in terms of median prediction error, with time-series-based models (i.e., TS and ITS) being more accurate on the NGL/PPP dataset on magnitude and location accuracy, respectively. All the models have a larger error associated with the NGL/PPP dataset, probably

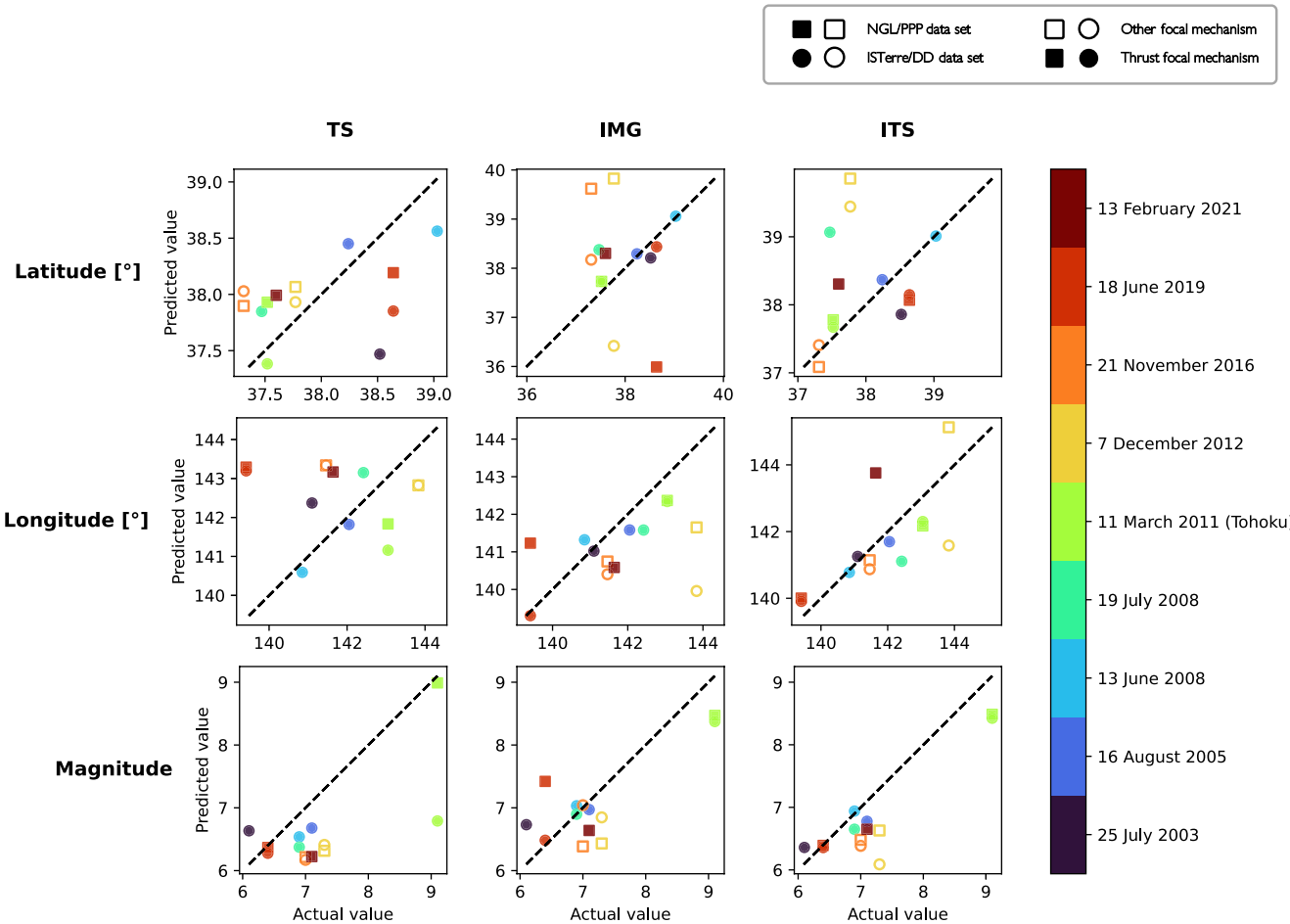

Figure 11. Actual versus predicted plot on real data from ISTerre/DD and NGL/PPP datasets. Each subplot shows the *real* versus *predicted* comparison for the estimated parameters (fault centroid latitude, fault centroid longitude, and magnitude in each row) for each of the three methods (TS, IMG, and ITS in each column). For each scatter plot, circles and squares represent predictions associated with the ISTerre/DD and NGL/PPP datasets, respectively. Filled markers represent events having a trust focal mechanism, while empty markers indicate any other focal mechanism. The solid dashed line shows the line of perfect prediction. The data points are color-coded according to the time of occurrence.

because of the Precise Point Positioning solution, which is slightly noisier with respect to the DD approach, which has been used in the noise generation phase. From these results, it seems that analyzing longer time periods (time-series-based approaches) may help reduce the error on noisier data. However, Figure 11 shows that, on average, the predictions of the image-based models, that is, IMG and ITS, are less scattered with respect to the perfect prediction line and have less variability. Thus, their performance is globally more accurate than the TS model on both datasets, in line with the results obtained on synthetic data (cf. Section 3). The presence of a larger amount of location and magnitude error in the case of TS may be probably linked to data gaps and missing stations in the real data, which worsen its resemblance to synthetic data. This also can introduce potential artifacts via the interpolation in the time domain, thus deteriorating the performance of TS, notably for the Tohoku earthquake (11 March 2011) in the NGL/PPP dataset, where the co-seismic offset may have been masked. As a result, image-based models can better deal with data gaps thanks to spatial interpolation. Hence, the amount and continuity of the data play an essential role in the final prediction accuracy, which is mitigated by the image and image time series representations. Since the models have been trained on a synthetic dataset obtained from a DD approach, we will focus on the ISTerre/DD dataset henceforward, which also has more data samples to analyze.

The events in Figure 11 have been marked with a full-colored symbol if their rupture has a thrust focal mechanism ($\phi_S = 200 \pm 40°$, $\delta = 35 \pm 30°$, $\lambda = 90 \pm 45°$). Differentiating thrust and non-thrust events is interesting to assess if the shape of the associated deformation field plays a key role in the characterization performed by image-based models, given that the model was trained on thrust events only. The results shown in Figure 11 seem to suggest

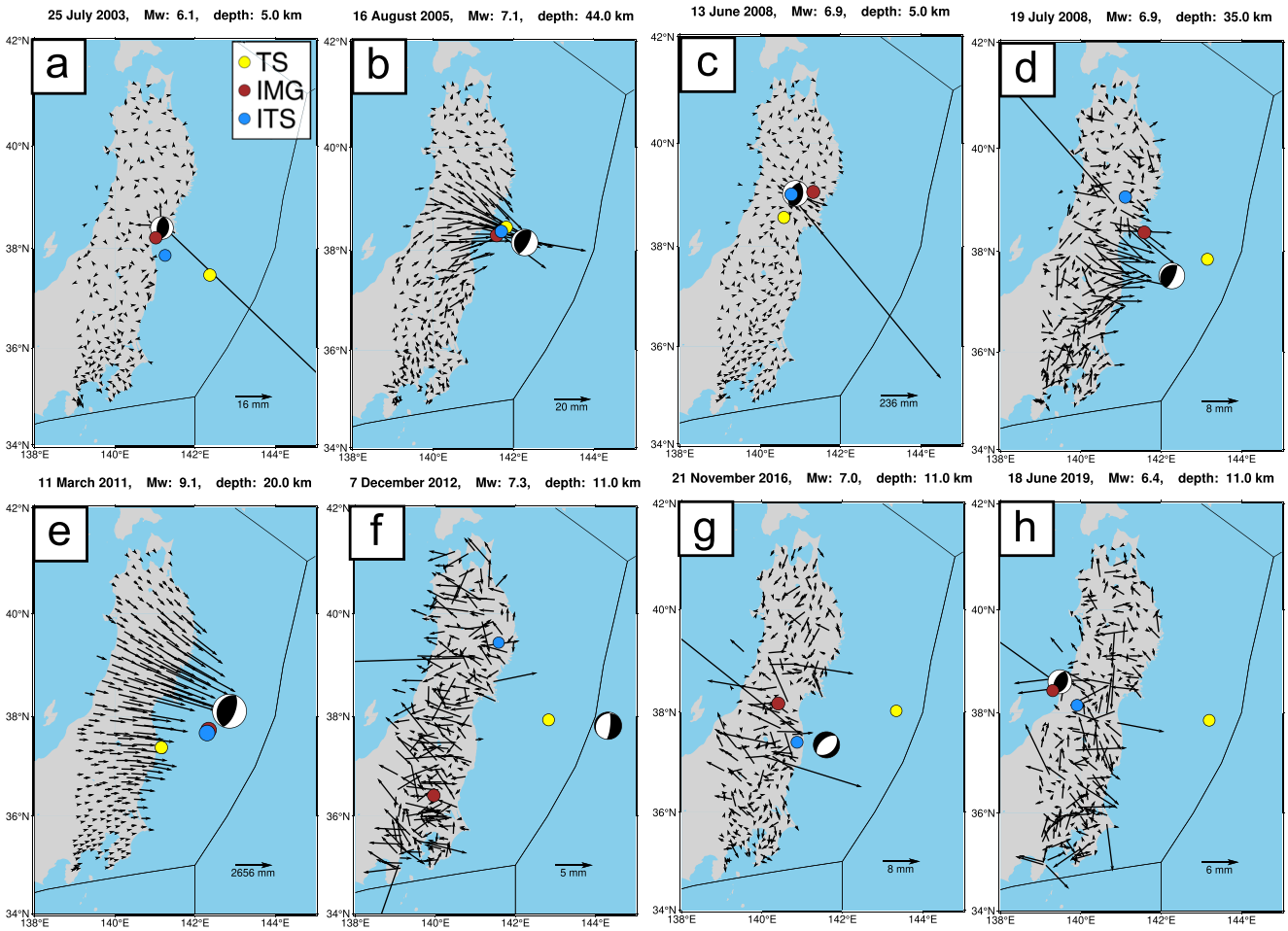

Figure 12. Displacement fields associated with the eight events of the ISTerre/DD dataset. The deformation fields have been computed by subtracting the deformation at day $t_c + 1$ and $t_c - 1$. In each subplot, the focal mechanism from the NIED catalog is shown as well as the magnitude and depth (in each title) with the yellow, brown, and blue points representing the predictions for TS, IMG, and ITS, respectively.

that the shape of the displacement field might be a relevant feature in the characterization of the location and the magnitude for IMG and ITS, since their predictions seem to depend on the nature of the focal mechanism. However, all the models have been trained on a dataset made of events having thrust focal mechanisms, thus they might not be able to generalize well in a setting that is different from the training one.

Interestingly, IMG and ITS models seem to be complementary on some events, as shown in Figures 12d, 12g and 12h. The ITS model is unable to separate the source of displacement in the 19 July 2008 event (Figure 12d) because of an outlier in the displacement field, whose influence is better mitigated by the differential approach used for IMG (cf. Figures S6–S8 in Supporting Information S1). On the contrary, ITS can more effectively retrieve the 21 November 2016 event (Figure 12g), likely thanks to the spatiotemporal approach (cf. Figures S9–S14 in Supporting Information S1), while IMG is less well performing. This seems to suggest that the two different image-based data representations carry some particular characteristics coming from the network geometry and the spatiotemporal variability of the data.

We also notice that the outlier displacement value north of the epicenter of the 19 July 2008 event (cf. Figure 12d) is actually an artifact introduced by the linear interpolation performed on the time series in presence of a large data gap. Therefore, either a more efficient method should be set up for the missing data interpolation, or artifacts should be taken into account in the training database. Accounting for the data gaps is not a trivial task and future developments should focus on this aspect, since, as we saw, the larger the data gaps, the harder the characterization.

It is worth mentioning the performance of the models on the Tohoku event (11 March 2011, $M_w = 9.1$), which is estimated as a $M_w \sim 8.5$ event by IMG and ITS, and as a $M_w \sim 9$ event by TS on the NGL/PPP dataset and as a $M_w \sim 6.8$ event on the ISTerre/DD dataset. Although these results may suggest that TS might perform well on patterns whose magnitude has never been presented to the network before, they also should be taken carefully since we cannot assess the robustness of the methods in case of testing against data having different characteristics.

In this study, we generate uniformly distributed events to train the deep learning models, yet we set the focal mechanism to be a thrust. We motivate this mainly by claiming that we are interested in characterizing thrust subduction events and thrust crustal events as well. However, one may ask how the models trained on a variety of focal mechanisms would perform on real data, with respect to the ones trained on thrust events. Results on synthetic data are presented in Figures S15–S19 in Supporting Information S1 and show that the models trained on all focal mechanisms have slightly lower performances and more variability than the models trained on thrust only. This can easily be explained by the variety of synthetic signals to be learned by the model. Since information about the focal mechanism itself is not given to the model (only magnitude and location are given), the model does not learn that those differences exist, and, as a consequence, it cannot discriminate between a signal from a thrust of a strike-slip event, even though both earthquakes generate a different deformation field at the surface. When tested on real data (Figure S20 in Supporting Information S1), we can see that the models trained on all focal mechanisms can generalize slightly better on non-thrust events. However, their performance on thrust (subduction or crustal) events is worse than the models trained on thrust events. We therefore chose to restrict our case study and the core of the discussion to a simpler case with a single focal mechanism only (thrust events).

5. Conclusions

We studied and developed an end-to-end framework for the seismic source characterization with GNSS data. We constructed three deep learning methods associated with three data representations: time series, differential images, and image time series. We train our methods on synthetic data generated to be subduction events compliant with actual events occurring in the Japan subduction zone. We tested the methods both on synthetic and real GNSS data, and we studied the performance and the sensitivity of the three methods, evidencing their strengths and their limits.

Image-based methods outperform time series-based methods on synthetic data, possibly because their associated data representations better exploit the topology of the GNSS network. The wavelength of the deformation is seemingly better constrained with images with respect to time series, the longitudinal extent of the deformation being more difficult to characterize by means of the temporal evolution only. Results on synthetic data clearly evidence a detection threshold associated with GNSS data, which is linked to the SNR, and also dependent on the depth and position of events. This allows us to partition the output space by identifying regions in which the source characterization can be performed with confidence.

The performance on real datasets is globally consistent with the results obtained on synthetic data, although it is hard to draw robust statistics due to the small amount of testing data. Image-based methods, that is, IMG and ITS, qualitatively outperform the TS approach in both real datasets, being able to retrieve the source parameters of most of the tested events, with IMG effective in characterizing most of them. The ITS model shows that the proposed spatiotemporal approach is crucial in resolving the location and magnitude of some of the real events where IMG had poor performance. This result confirms that accounting for the pre- and post-event noise level can lead ITS to a better estimate of the co-seismic offset. However, the noise characterization needs to be improved, in order to better account for outliers in GNSS time series, data gaps, and, possibly, common modes. By improving the simulation of the realistic noise, we can produce more and more real-looking synthetic data, possibly having better results on the characterization and a lower SNR threshold. Also, we might expect that the performance on real data would improve if we generated a more realistic synthetic dataset that contains also a post-seismic signal in addition to the co-seismic offset, as shown in Section 3.6. Nonetheless, the results on real data are promising and could potentially also lead to an effective analysis of slow deformation, which would benefit from the present work as well as from the potential refinements that we have listed before.

Data Availability Statement

All rough data used in this manuscript are available at https://doi.osug.fr/public/GNSS_products/GNSS.products.Japan.html (ISTerre/DD) and http://geodesy.unr.edu (NGL). All the software as well as the pre-trained deep learning models are available at: https://gricad-gitlab.univ-grenoble-alpes.fr/costangi/eq-characterization.

Acknowledgments
This work has been supported by ERC CoG 865963 DEEP-trigger. Most of the computations presented in this paper were performed using the GRICAD infrastructure (https://gricad.univ-grenoble-alpes.fr), which is supported by Grenoble research communities. All the computations needed to build images and image time series, as well as map plots, have been performed thanks to the GMT software (and its Python wrapper, pyGMT) (Wessel et al., 2019). We would sincerely thank Yosuke Aoki and the two anonymous reviewers for their valuable comments and suggestions, which helped us improve the quality of the manuscript.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: A system for large-scale machine learning. In *Osdi* (Vol. 16, No. 2016, pp. 265–283).
Agarap, A. F. (2018). Deep learning using rectified linear units (ReLu). *arXiv*. preprint arXiv:1803.08375.
Aki, K., & Richards, P. G. (2002). *Quantitative seismology*. University Science Books.
Anantrasirichai, N., Biggs, J., Albino, F., & Bull, D. (2019). A deep learning approach to detecting volcano deformation from satellite imagery using synthetic datasets. *Remote Sensing of Environment*, *230*, 111179. https://doi.org/10.1016/j.rse.2019.04.032
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
Bevis, M., & Brown, A. (2014). Trajectory models and reference frames for crustal motion geodesy. *Journal of Geodesy*, *88*(3), 283–311. https://doi.org/10.1007/s00190-013-0685-5
Blewitt, G., Hammond, W., & Kreemer, C. (2018). Harnessing the GPS data explosion for interdisciplinary science. *Eos*, *99*. https://doi.org/10.1029/2018EO104623
Blewitt, G., Hammond, W. C., Kreemer, C., Plag, H.-P., Stein, S., & Okal, E. (2009). GPS for real-time earthquake source determination and tsunami warning systems. *Journal of Geodesy*, *83*(3–4), 335–343. https://doi.org/10.1007/s00190-008-0262-5
Bock, Y., & Melgar, D. (2016). Physical applications of GPS geodesy: A review. *Reports on Progress in Physics*, *79*(10), 106801. https://doi.org/10.1088/0034-4885/79/10/106801
Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, *60*(2), 223–311. https://doi.org/10.1137/16m1080173
Bürgmann, R. (2018). The geophysics, geology and mechanics of slow fault slip. *Earth and Planetary Science Letters*, *495*, 112–134. https://doi.org/10.1016/j.epsl.2018.04.062
Chollet, F. (2021). *Deep learning with python*. Simon and Schuster.
Chollet, F., et al. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras
Dong, D., Fang, P., Bock, Y., Cheng, M., & Miyazaki, S. (2002). Anatomy of apparent seasonal variations from GPS-derived site position time series. *Journal of Geophysical Research*, *107*(B4), ETG9-1–ETG9-16. https://doi.org/10.1029/2001jb000573
Dziewonski, A. M., Chou, T.-A., & Woodhouse, J. H. (1981). Determination of earthquake source parameters from waveform data for studies of global and regional seismicity. *Journal of Geophysical Research*, *86*(B4), 2825–2852. https://doi.org/10.1029/jb086ib04p02825
Ekström, G., Nettles, M., & Dziewoński, A. (2012). The global CMT project 2004–2010: Centroid-moment tensors for 13,017 earthquakes. *Physics of the Earth and Planetary Interiors*, *200*, 1–9. https://doi.org/10.1016/j.pepi.2012.04.002
Feng, L., Hill, E. M., Banerjee, P., Hermawan, I., Tsang, L. L., Natawidjaja, D. H., et al. (2015). A unified GPS-based earthquake catalog for the sumatran plate boundary between 2002 and 2013. *Journal of Geophysical Research: Solid Earth*, *120*(5), 3566–3598. https://doi.org/10.1002/2014jb011661
Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).
gnss products. (2019). GNSS position solutions in Japan. CNRS, OSUG, ISTERRE. https://doi.org/10.17178/GNSS.products.Japan
Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
Guo, A., Ni, S., Chen, W., Freymueller, J. T., & Shen, Z. (2015). Rapid earthquake focal mechanism inversion using high-rate GPS velometers in sparse network. *Science China Earth Sciences*, *58*(11), 1970–1981. https://doi.org/10.1007/s11430-015-5174-7
Gutenberg, B. (1956). The energy of earthquakes. *Quarterly Journal of the Geological Society*, *112*(1–4), 1–14. https://doi.org/10.1144/gsl.jgs.1956.112.01-04.02
Hanks, T. C., & Kanamori, H. (1979). A moment magnitude scale. *Journal of Geophysical Research*, *84*(B5), 2348–2350. https://doi.org/10.1029/jb084ib05p02348
Haykin, S. S. (2008). *Adaptive filter theory*. Pearson Education India.
He, B., Wei, M., Watts, D. R., & Shen, Y. (2020). Detecting slow slip events from seafloor pressure data using machine learning. *Geophysical Research Letters*, *47*(11), e2020GL087579. https://doi.org/10.1029/2020gl087579
Hulbert, C., Rouet-Leduc, B., Johnson, P. A., Ren, C. X., Rivière, J., Bolton, D. C., & Marone, C. (2019). Similarity of fast and slow earthquakes illuminated by machine learning. *Nature Geoscience*, *12*(1), 69–74. https://doi.org/10.1038/s41561-018-0272-8
Hulbert, C., Rouet-Leduc, B., Jolivet, R., & Johnson, P. A. (2020). An exponential build-up in seismic energy suggests a months-long nucleation of slow slip in cascadia. *Nature Communications*, *11*(1), 1–8. https://doi.org/10.1038/s41467-020-17754-9
Ji, K. H., & Herring, T. A. (2013). A method for detecting transient signals in GPS position time-series: Smoothing and principal component analysis. *Geophysical Journal International*, *193*(1), 171–186. https://doi.org/10.1093/gji/ggt003
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
Lay, T. (2018). A review of the rupture characteristics of the 2011 Tohoku-oki Mw 9.1 earthquake. *Tectonophysics*, *733*, 4–36. https://doi.org/10.1016/j.tecto.2017.09.022
LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539
Lin, J.-T., Chang, W.-L., Melgar, D., Thomas, A., & Chiu, C.-Y. (2019). Quick determination of earthquake source parameters from GPS measurements: A study of suitability for Taiwan. *Geophysical Journal International*, *219*(2), 1148–1162. https://doi.org/10.1093/gji/ggz359
Mao, A., Harrison, C. G., & Dixon, T. H. (1999). Noise in GPS coordinate time series. *Journal of Geophysical Research*, *104*(B2), 2797–2816. https://doi.org/10.1029/1998jb900033
Marill, L., Marsan, D., Socquet, A., Radiguet, M., Cotte, N., & Rousset, B. (2021). Fourteen-year acceleration along the Japan trench. *Journal of Geophysical Research: Solid Earth*, *126*(11), e2020JB021226. https://doi.org/10.1029/2020jb021226

Moon, T. K., & Stirling, W. C. (2000). Mathematical methods and algorithms for signal processing (No. 621.39: 51 MON).
Mousavi, S. M., & Beroza, G. C. (2020). A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*, *47*(1), e2019GL085976. https://doi.org/10.1029/2019gl085976
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
Münchmeyer, J., Bindi, D., Leser, U., & Tilmann, F. (2021). Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network. *Geophysical Journal International*, *226*(2), 1086–1104. https://doi.org/10.1093/gji/ggab139
Münchmeyer, J., Bindi, D., Sippl, C., Leser, U., & Tilmann, F. (2020). Low uncertainty multifeature magnitude estimation with 3-D corrections and boosting tree regression: Application to North Chile. *Geophysical Journal International*, *220*(1), 142–159. https://doi.org/10.1093/gji/ggz416
Okada, Y. (1985). Surface deformation due to shear and tensile faults in a half-space. *Bulletin of the Seismological Society of America*, *75*(4), 1135–1154. https://doi.org/10.1785/bssa0750041135
Page, M. T., Custódio, S., Archuleta, R. J., & Carlson, J. (2009). Constraining earthquake source inversions with GPS data: 1. Resolution-based removal of artifacts. *Journal of Geophysical Research*, *114*(B1). https://doi.org/10.1029/2007jb005449
Pearson, R. K., Neuvo, Y., Astola, J., & Gabbouj, M. (2016). Generalized Hampel filters. *EURASIP Journal on Advances in Signal Processing*, *2016*(1), 1–18. https://doi.org/10.1186/s13634-016-0383-6
Prichard, D., & Theiler, J. (1994). Generating surrogate data for time series with several simultaneously measured variables. *Physical Review Letters*, *73*(7), 951–954. https://doi.org/10.1103/physrevlett.73.951
Riquelme, S., Bravo, F., Melgar, D., Benavente, R., Geng, J., Barrientos, S., & Campos, J. (2016). W phase source inversion using high-rate regional GPS data for large earthquakes. *Geophysical Research Letters*, *43*(7), 3178–3185. https://doi.org/10.1002/2016gl068302
Ross, Z. E., Yue, Y., Meier, M.-A., Hauksson, E., & Heaton, T. H. (2019). Phaselink: A deep learning approach to seismic phase association. *Journal of Geophysical Research: Solid Earth*, *124*(1), 856–869. https://doi.org/10.1029/2018jb016674
Rouet-Leduc, B., Hulbert, C., & Johnson, P. A. (2019). Continuous chatter of the cascadia subduction zone revealed by machine learning. *Nature Geoscience*, *12*(1), 75–79. https://doi.org/10.1038/s41561-018-0274-6
Rouet-Leduc, B., Hulbert, C., McBrearty, I. W., & Johnson, P. A. (2020). Probing slow earthquakes with deep learning. *Geophysical Research Letters*, *47*(4), e2019GL085870. https://doi.org/10.1029/2019gl085870
Rouet-Leduc, B., Jolivet, R., Dalaison, M., Johnson, P. A., & Hulbert, C. (2021). Autonomous extraction of millimeter-scale deformation in InSAR time series using deep learning. *Nature Communications*, *12*(1), 1–11. https://doi.org/10.1038/s41467-021-26254-3
Saad, O. M., Hafez, A. G., & Soliman, M. S. (2020). Deep learning approach for earthquake parameters classification in earthquake early warning system. *IEEE Geoscience and Remote Sensing Letters*, *18*(7), 1293–1297. https://doi.org/10.1109/lgrs.2020.2998580
Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4510–4520).
Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv*. preprint arXiv:1312.6120.
Schreiber, T., & Schmitz, A. (1996). Improved surrogate data for nonlinearity tests. *Physical Review Letters*, *77*(4), 635–638. https://doi.org/10.1103/physrevlett.77.635
Seydoux, L., Balestriero, R., Poli, P., Hoop, M. d., Campillo, M., & Baraniuk, R. (2020). Clustering earthquake signals and background noises in continuous seismic data with unsupervised deep learning. *Nature Communications*, *11*(1), 1–12. https://doi.org/10.1038/s41467-020-17841-x
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv*. preprint arXiv:1409.1556.
Smith, W., & Wessel, P. (1990). Gridding with continuous curvature splines in tension. *Geophysics*, *55*(3), 293–305. https://doi.org/10.1190/1.1442837
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.
van den Ende, M. P., & Ampuero, J.-P. (2020). Automated seismic source characterization using deep graph neural networks. *Geophysical Research Letters*, *47*(17), e2020GL088690. https://doi.org/10.1029/2020GL088690
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).
Wang, Q., Guo, Y., Yu, L., & Li, P. (2017). Earthquake prediction based on spatio-temporal data mining: An LSTM network approach. *IEEE Transactions on Emerging Topics in Computing*, *8*(1), 148–158. https://doi.org/10.1109/tetc.2017.2699169
Wessel, P., Luis, J., Uieda, L., Scharroo, R., Wobbe, F., Smith, W., & Tian, D. (2019). The generic mapping tools version 6. *Geochemistry, Geophysics, Geosystems*, *20*(11), 5556–5564. https://doi.org/10.1029/2019gc008515
Weston, J., Ferreira, A. M., & Funning, G. J. (2012). Systematic comparisons of earthquake source models determined using InSAR and seismic data. *Tectonophysics*, *532*, 61–81. https://doi.org/10.1016/j.tecto.2012.02.001
Williams, S. D., Bock, Y., Fang, P., Jamason, P., Nikolaidis, R. M., Prawirodirdjo, L., et al. (2004). Error analysis of continuous GPS position time series. *Journal of Geophysical Research*, *109*(B3), B03412. https://doi.org/10.1029/2003jb002741
Zhang, J., Bock, Y., Johnson, H., Fang, P., Williams, S., Genrich, J., et al. (1997). Southern California permanent GPS geodetic array: Error analysis of daily position estimates and site velocities. *Journal of Geophysical Research: Solid Earth*, *102*(B8), 18035–18055. https://doi.org/10.1029/97jb01380
Zhang, X., Zhang, M., & Tian, X. (2021). Real-time earthquake early warning with deep learning: Application to the 2016 M 6.0 Central Apennines, Italy earthquake. *Geophysical Research Letters*, *48*(5), 2020GL089394. https://doi.org/10.1029/2020gl089394
Zhu, W., & Beroza, G. C. (2019). Phasenet: A deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, *216*(1), 261–273.
Zhu, W., Mousavi, S. M., & Beroza, G. C. (2019). Seismic signal denoising and decomposition using deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(11), 9476–9488. https://doi.org/10.1109/tgrs.2019.2926772