DEEP LEARNING APPLIED TO THE DETECTION AND UNDERSTANDING OF SLOW SLIP EVENTS

Giuseppe Costantino

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : STEP - Sciences de la Terre de l'Environnement et des Planètes
Spécialité : Sciences de la Terre et de l'Environnement
Unité de recherche : Institut des Sciences de la Terre

APPRENTISSAGE PROFOND APPLIQUÉ À LA DÉTECTION ET LA COMPRÉHENSION DES GLISSEMENTS LENTS

DEEP LEARNING APPLIED TO THE DETECTION AND UNDERSTANDING OF SLOW SLIP EVENTS

Présentée par :

Giuseppe COSTANTINO

Direction de thèse :

Anne SOCQUET PHYSICIENNE, Université Grenoble Alpes	Directrice de thèse
Mauro DALLA MURA MAITRE DE CONFERENCES, Université Grenoble Alpes	Co-encadrant de thèse
Sophie GIFFARD-ROISIN CHARGEE DE RECHERCHE, Institut de Recherche pour le Développement	Co-encadrante de thèse

Rapporteurs :

Romain JOLIVET
PROFESSEUR DES UNIVERSITES, ENS Paris
Gregory C. BEROZA
PROFESSEUR, Stanford University

Thèse soutenue publiquement le **25 septembre 2023**, devant le jury composé de :

Anne SOCQUET PHYSICIENNE, Université Grenoble Alpes	Directrice de thèse
Romain JOLIVET PROFESSEUR DES UNIVERSITES, ENS Paris	Rapporteur
Gregory C. BEROZA PROFESSEUR, Stanford University	Rapporteur
Joan GOMBERG SCIENTIST, Washington University	Examinatrice
Jérôme MARS PROFESSEUR, Grenoble INP	Président
Diane RIVET PHYSICIENNE ADJOINTE, Université Côte d'Azur	Examinatrice

Invités :

Mauro DALLA MURA
MAITRE DE CONFERENCES, UGA
Sophie GIFFARD-ROISIN
CHARGEE DE RECHERCHE, IRD

Abstract

The discovery of slow slip events at plate boundaries in the 2000s was made possible by the continuous monitoring of ground deformation by geodetic networks and constituted a change of paradigm for the understanding of the earthquake cycle and of the mechanics of the fault interface. Twenty years later, the characterization of the full slip spectrum and the understanding of the link between slow slip and associated seismological signals are still hindered by our capacity to detect, in a systematic manner, all the slow slip events, including the smallest ones. In this Ph.D. thesis, we enforce a multidisciplinary approach to join geophysics and deep learning. By taking advantage of the explosion of large-scale data sets, the number of deep learning applications has experienced enormous growth in geosciences in recent years. While most of the works focus on seismological data, geodetic measurements are still poorly explored, and the applications of deep learning to the identification of slow slip events in the Global Navigation Satellite System (GNSS) data are still arduous, mainly because of the difficulty in dealing with the fragmented availability of catalogues and the challenges related to the spatiotemporal complexity of the noise. Here, we develop deep learning methods to detect and characterize slow slip events in non-post-processed GNSS data. We first benchmark the distillation of spatial and temporal features in GNSS data at best, and the development of deep learning strategies targeted to leverage them. We develop SSEdetector, an attentive convolutional deep-learning method to detect slow slip events in real GNSS data in Cascadia, which proves effective in revealing patterns of slow-slip-event-induced displacement with a multi-station approach. Finally, we address the problem of denoising GNSS time series as a preliminary step towards slow slip event characterization. We present SSEdenoiser, a spatiotemporal graph-based attentive denoiser that learns the latent characteristics of GNSS noise to reveal slow slip event displacement with sub-millimeter precision.

Abstract (français)

La surveillance régulière de la déformation du sol par les réseaux géodésiques a permis la découverte de glissements lents aux frontières de plaques dans les années 2000. Cela a changé la façon dont nous comprenons le cycle sismique et la mécanique de l'interface des failles. Vingt ans plus tard, notre capacité à détecter systématiquement tous les événements de glissement lent, y compris les plus petits, reste un obstacle à la caractérisation du spectre complet de glissement et à la compréhension du lien entre le glissement lent et les signaux sismologiques associés. Dans cette thèse de doctorat, nous utilisons une approche multidisciplinaire pour joindre la géophysique et l'apprentissage profond. Au cours des dernières années, le nombre d'applications d'apprentissage profond a augmenté considérablement dans les géosciences en tirant parti de la disponibilité de grandes quantités de données. Alors que la plupart des travaux se concentrent sur les données sismologiques, les mesures géodésiques sont encore peu explorées et les applications d'apprentissage profond pour identifier les glissements lents dans les données GNSS sont encore ardues à mettre en place, principalement en raison de la difficulté à traiter la disponibilité fragmentaire des catalogues et des défis liés à la complexité spatio-temporelle du bruit. Ici, nous développons des méthodes d'apprentissage profond pour détecter et caractériser les glissements lents dans les données GNSS (Global Navigation Satellite System) brutes (non post-traitées). Nous commençons par évaluer l'obtention des caractéristiques spatiales et temporelles des données GNSS et créons des stratégies d'apprentissage profond pour les exploiter. Nous développons SSEdetector, une méthode d'apprentissage profond basée sur des réseaux convolutifs et des mécanismes d'auto-attention, pour détecter les glissements lents dans les données GNSS réelles aux Cascades. Avec une approche multi-station, il se révèle efficace pour identifier le déplacement induit par les glissements lents à la surface. Enfin, comme étape préliminaire à la caractérisation des glissements lents, nous abordons le problème du débruitage des séries temporelles GNSS. Nous présentons SSEdenoiser, un débruiteur basé sur des réseaux de neurones graphiques spatio-temporels avec un mécanisme d'attention qui apprend les caractéristiques latentes du bruit GNSS pour révéler les déplacements dus aux glissements lents avec une précision submillimétrique.

Acknowledgements

At the conclusion of my Ph.D., I realize more than ever how much effort and devotion I have invested in science over the last three years. The accomplishments in this thesis are not solely attributed to my strengths, but rather, they reflect the combined contributions of the collaborative efforts of those who worked alongside me, as well as the valuable support gained from interacting with the people who surround my life.

First of all, thanks to Greg Beroza and Romain Jolivet for agreeing to review my manuscript and for your invaluable time, expertise, and constructive feedback. Thank you Greg for providing your broad and enthusiastic vision and thank you Romain for your thorough and insightful evaluation. Thanks to Jérôme Mars for agreeing to continuously evaluate my work during the Ph.D. and to serve as president of the jury. Thank you for always providing your pragmatic yet forward-looking feedback. Thank you to Joan Gomberg and Diane Rivet for being also part of the examiners' board. Thanks for the commitment that you demonstrated through your constructive feedback and your accurate remarks.

I feel grateful for the chance to work with my Ph.D. committee, made of extremely talented scientists who have always been there for guidance and support.
First, I want to thank Anne Socquet, my Ph.D. director. I feel really lucky to have met you during my master's internship and glad that you proposed this Ph.D. to me right after. You possess the extraordinary ability to see things that other people just overlook. And from this scientific sense of smell, among your greatest talents, I have always tried to learn and capture the most subtle nuances. Not only do you have exceptional professional and scientific skills, but also personal and social qualities that are rare to find, especially in the increasingly competitive and tough world of science. It is a blessing to know that there are people like you. I owe a lot to you for the scientist I have become.
Thanks to Mauro Dalla Mura, my Ph.D. co-supervisor. You have been the bridge to this journey in science back at the time of my master's internship. Thank you for always encouraging me to be factual and punctual in science.
Thanks to Sophie Giffard-Roisin, my Ph.D. co-supervisor. I have loved this Ph.D. experience together. You are a comforting and optimistic person and your support has been fundamental, especially in stressful situations when you were encouraging

and guiding me to find a solution that has always materialized. Even in stressful moments, you have always helped me to take a step behind and see things from an external perspective. Your “no worries, you will find a way out" has always come at the right moment. Thank you for helping me find a balance between my imposter syndrome and my endless perfectionism.
Thanks to Mathilde Radiguet and David Marsan, your scientific support during these three years has been just amazing.

Thanks to the *cycle* team for all the moments of conviviality that we have shared, for this lovely tradition of the Friday team coffee, for our numerous team pizzas, and just for the time spent together. Thanks to all of you for being a happy sparkling presence in this lab during these years. It will be hard to leave such an amazing group.

Thanks to all the people I met at ISTerre: each of you has given me something that I will bring with me.

Among all the ISTerre people, some of them became my closest friends and people I share my life with. Thank you to my dream team: Anuar, Audrey, Béné, Diego, Javiera, Juliette, Léo, Mirko, Renata, Shan. Thank you for your time together, the most precious gift that people can give each other. Even though you don’t agree (not yet!), I still think that we can study the nucleation of earthquakes in ground coffee and build artificial-intelligence-empowered wind turbines.

Thank you Juliette for being an incredible office mate and friend. We started our Ph.D. at the same time three years ago and, since then, we shared all the highs and lows that a Ph.D. involves. I am grateful that you have always been there and always found the time for me, especially when I needed it most. Besides emotional support, you have also given me impressive scientific feedback, during our endless sessions by the whiteboard: I admire your mental flexibility and your ability to discuss deep learning with me as if you have been doing that for ages.
Thank you Diego for being my favourite *cahuinero*. In such a short time, you have represented outstanding emotional support and have shown me the meaning of friendship through a strong emotional connection. Thank you for your support during the writing of the manuscript and for encouraging me to trust the process and, most importantly, to trust myself.
Thanks Mirko for being my daily dose of Neapolitan and for our countless coffee breaks. Thank you for being an unwavering and comforting presence and always ready to offer assistance in any situation.
Thank you Renata for being a fabulous personal trainer and essential support.

Thank you Béné for being an amazing partner during our field trip in Peru last year and such a joyful person.

Thank you Ivana for bringing out the best in me.

Thank you Giuseppe M. for being more than a friend. Words are not enough to describe what you are to me and how grateful I am to have you in my life.

Thank you Luigi for being a true friend. Being your best man was the best gift I could have received.

Thanks to my family, love nest forever.

Contents

Dedicated to my parents
Ai miei genitori

C’est la nuit qu’il faut croire à la lumière

1 General introduction

From the perspective of the geological phenomena occurring at the human time scale, earthquakes are among the most harmful processes. Our planet has witnessed some of the most destructive earthquakes during the past years, most of which occurred in subduction zones, *i.e.*, where two tectonic plates meet and slide, one beneath the other. Subduction (megathrust) earthquakes have been among the deadliest events, causing tens of thousands of fatalities (Lay, 2015), such as the 2004 M_w 9.4 Sumatra (Ammon et al., 2005), 2010 M_w 8.8 Maule (Delouis et al., 2010), 2011 M_w 9.0 Tohoku (Suzuki et al., 2011) earthquakes. Thus, it is essential to understand the processes leading to stress release at subduction zones.

At the beginning of the 2000s, the continuous monitoring of ground displacement with the Global Navigation Satellite System (GNSS) empowered the discovery of slow slip events (SSEs) in many subduction zones, such as Cascadia (North America) (Dragert et al., 2001b; Rogers & Dragert, 2003a), Nankai (Japan) (Obara et al., 2004; Ozawa et al., 2001), New Zealand (Douglas et al., 2005; Wallace & Beavan, 2010), Mexico (Kostoglodov et al., 2003; Lowry et al., 2001a). Like earthquakes, slow slip events also contribute to releasing stress on the fault, through a slow slip on the subduction interface lasting a couple of days to a couple of months, yet without generating seismic waves. For this reason, geodetic data (*e.g.*, here GNSS position time series) yields the sole access to their identification. Moreover, slow slip events are often accompanied by a diversity of seismic signals, such as non-volcanic tremors (Dragert et al., 2004; Obara, 2002), low-frequency earthquakes[1] (LFEs) (Beroza & Ide, 2011) and seismic swarms[2] (Hirose et al., 2014), revealing latent relationships between seismic and aseismic phenomena. Most intriguingly, they can induce frictional instabilities of close asperities and influence the nucleation of earthquakes. Even though aseismic slip has been geodetically observed before large earthquakes (Radiguet et al., 2016; Socquet et al., 2017), the question still persists whether slow slip events may be responsible for promoting the conditions leading to the nucleation of megathrust earthquakes.

[1] Tremors and LFEs are weakly-emergent micro-seismicity or micro-seismicity with low-frequency content (1-10 Hz).

[2] Seismic swarms are seismic sequences in which there are no clear mainshock-aftershock patterns.

There are still several unanswered questions that need to be addressed. Are slow slip events happening everywhere? When they are not detected, is that due to a limit of detection? If not present, why are there regions where slow slip events do not occur? What is the relationship between slow slip events and the accompanying seismicity? Is there any causal link? What is the mechanics of slow slip events? In what they differ from regular earthquakes? What are the conditions that promote slow slip? Systematic and complete mapping of slow slip events is key to addressing these questions. SSE catalogues in given areas would allow for inter- and intra-event analyses and extraction of scaling laws. Obtaining scaling laws is essential to better understand the physics of slow slip events, as this can provide insights into the mechanics of fault slip, its geometry and kinematic, and similarities with regular earthquakes, towards a better comprehension of the underlying conditions that promote the nucleation, growth, and propagation of slow slip (Bletery et al., 2017; Gomberg et al., 2016b), also related to the frictional properties of the subduction interface, such as differences in stress conditions (C. Scholz, 1968; Tan et al., 2019) and fault roughness (Goebel et al., 2017; Pétrélis et al., 2023).

The major barrier impeding our ability to find answers to our questions lies in the difficulty of revealing slow slip events in the geodetic measurements. The signature of aseismic transients of deformation in GNSS time series is, for most of the events, hidden in the noise (Frank, 2016; Rousset et al., 2017b). Without sophisticated techniques to unveil the tiniest occurrences, our ability to comprehensively analyze the seismic spectrum and understand the relationship between seismic and aseismic phenomena, at both short and large space and time scales, remains fragmentary.

The primary objective of this thesis is to employ a comprehensive multidisciplinary approach and leverage advanced deep-learning methodologies, coupled with realistic physical simulations, to systematically map and analyze slow slip events occurring in subduction zones. Numerous geophysical investigations concentrate on individual events or specific tectonic regions, by visual examination of the data and specialized modeling techniques with specific calibration of the parameters (e.g. Bletery & Nocquet, 2020; Ozawa et al., 2013; Wallace, 2020). Another research direction focuses on systematically characterizing slow slip events with the dual purpose of constructing event catalogues for inter-event comparison and extraction of scaling laws, as well as enhancing the signal-to-noise ratio to detect and characterize events that are at the limit of detection capabilities (e.g. Frank & Brodsky, 2019; Ide, Beroza, et al., 2007a; Michel, Gualandi, & Avouac, 2019a). We align with the latter approach and our primary objective is to address the observational gap concerning small slow slip events with the help of deep learning techniques, in order to shed

light on the behaviour of slow slip events and their relationships with the seismic rupture and the underlying processes leading their nucleation and propagation.

The application of deep learning methods in seismology has experienced significant growth in recent years, leveraging the availability of large data sets. However, the majority of studies in this field have primarily focused on seismic data (e.g. Münchmeyer et al., 2021; Zhu & Beroza, 2019). Despite being a valuable source of information, GNSS data is still little used. The principal obstacle preventing consistent extraction of slow-slip-driven displacement in GNSS time series is the ensemble of perturbations affecting the measures of the position, *e.g.*, orbit computation errors, atmospheric perturbations, seasonal variations, and tectonic signals. Most of them are hard to model and can have a significant impact on the GNSS precision. All approaches, at present, attempt to model those contributions to first clean the data. Yet, this process can introduce biases and artifacts. This is why in this manuscript we investigate how to identify slow slip events from raw (non-post-processed) GNSS time series. In addition, we focus on analyzing geophysical networks with a multi-station approach instead of classical single-station analysis, since slow slip events produce spatially-coherent signals that need to be captured by multi-station strategies. To the best of our knowledge, this is the first time that multi-station deep-learning methods are employed for slow slip event identification with GNSS data. Although this presents a challenge, it allows us to maintain the integrity of the raw data and shift the ability to extract small signals buried in the noise to the methodology itself. Also, the scarcity of SSE catalogues represents a complication for deep learning, which we solve by developing a strategy to generate physics-based synthetic data to train advanced deep learning models.

Deep learning reveals promising to tackle this problem, because it can capture complex patterns in the GNSS data, typical of slow slip events. Also, it can autonomously extract relevant features from raw data, which is essential, since the spatiotemporal features associated with slow slip events are still poorly understood to have them as an *a priori* knowledge base. However, using existing methods is not adequate in our case, due to the challenge of working with multivariate data (*e.g.*, spatial and temporal dimensions, multiple components), thereby requiring the development of tailored methodologies, which must effectively deal with noisy GNSS measurements as well as missing data and uneven spatial coverage of GNSS stations.

With the developed approaches, we are now able to automatically process continuous GNSS data in order to detect, localize and retrieve the displacement produced by slow slip events by effectively dealing with the pervasive presence of geodetic noise, sparse GNSS measurements and missing data to extract millimeter-scale

displacement that was buried in the noise so far. Our original contributions hold potential interest for a broader audience beyond the geophysical domain, as they serve as a foundation for building models in custom applications, that encompass areas such as sensor-network time series analysis, building realistic data sets for end-to-end learning and the development of physic-based deep learning models.

This manuscript is organized as follows. Chapter 2 presents the multidisciplinary background needed for the rest of the manuscript. First, we provide an overview of the seismic cycle and slow slip events, by describing their observation, implications on the seismic cycle, and associated detection methods. Following, machine-learning and deep-learning background is provided, as well as a bibliography on their applications in solid earth sciences. Each of the models used in the manuscript has been explained to provide the reader with short context and easier comprehension.

In chapter 3, we describe earthquake characterization by means of deep learning in GNSS time series, as our pioneering work towards slow slip event analysis. In this chapter, we discuss different strategies to leverage GNSS data to make it effectively work with deep learning, as well as developing deep learning methods targeted to each GNSS-based data representation. We develop a training database of synthetic earthquakes and we test the performance of the trained deep-learning models against real earthquakes from Honshu, Japan.

In chapter 4, we present our deep-learning-based method to detect slow slip events in real GNSS time series in Cascadia in 2007-2022. We develop SSEgenerator, a method to generate realistic GNSS time series containing synthetic slow slip signals, and SSEdetector, an attentive multi-station deep-learning model trained to identify the signature of transients of aseismic deformation.

In chapter 5, we first discuss slow slip event characterization with spatiotemporal deep-learning methods. In the second part, we address the problem of slow slip characterization by first denoising the GNSS data by developing more complex methods capable to learn the spatiotemporal structure of the GNSS noise in order to effectively output clean real GNSS data in Cascadia in 2007-2022. Based on clean data, we further characterize the retrieved events and we provide preliminary scaling laws, in the direction of a systematic characterization and better understanding of slow slip events.

Finally, in chapter 6, we draw general conclusions and provide further perspectives.

Background

2

2.1 Subduction zones and seismic cycle

The Earth's surface is divided into 14 main tectonic plates. The mantle convection induces the relative movement of plates between each other of a few centimeters per year (*convergence rate*). Plates can meet at their boundaries (*plate margins*) in three main mechanisms. Plates can collide or approach in convergent margins (as in the case of the subduction), depart at divergent margins or they slide between each other at transform boundaries. In convergent margins, two continental plates can collide (orogenesis). Alternatively, as shown in Figure 2.1, in the case of an oceanic and a continental plate (or two oceanic plates), the denser (subducting plate) plunges below the other (overriding plate). This phenomenon is called subduction.

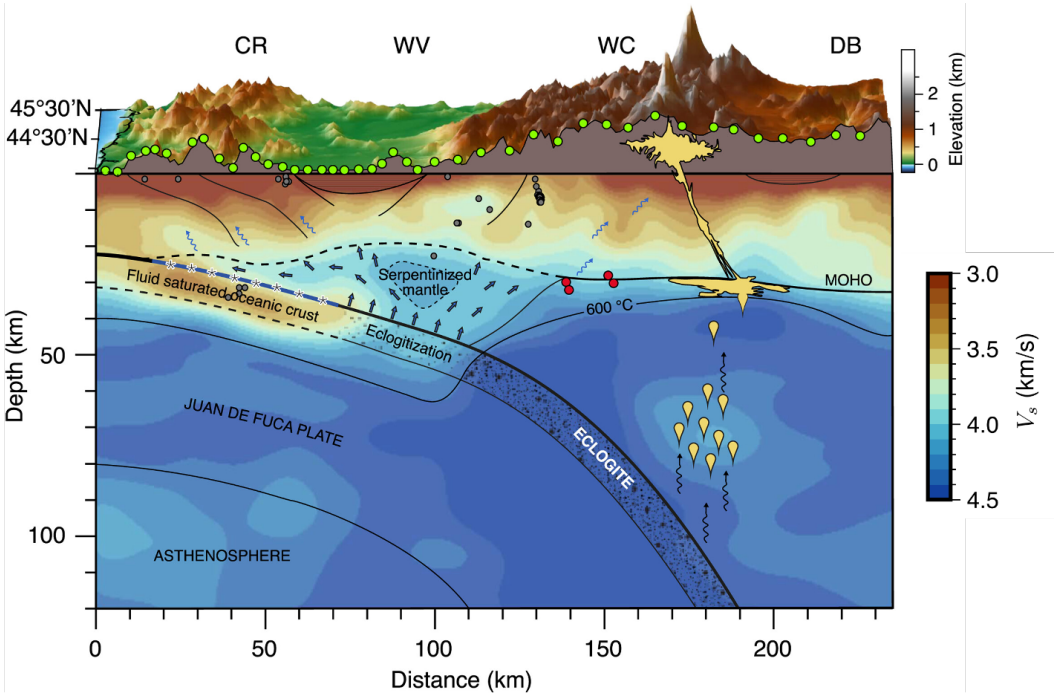


Fig. 2.1.: Illustration of the Cascadia subduction zone, after (Kan et al., 2023). Here, the Juan de Fuca plate subducts beneath the North American plate. The S-wave velocity V_s from the model of Kan et al., 2023 is shown by the heatmap.

The subduction interface (where the two plates are stuck together) is subject to stress accumulation. This stress is suddenly released by earthquakes. Subduction zones are thus areas where intense seismic activity concentrates. Because of that, the largest and the most violent earthquakes have been observed in subduction zones, producing large-scale deformation and harm (Lay, 2015), such as major earthquakes in the last decade: 2004 M_w 9.4 Sumatra (Ammon et al., 2005; Chlieh et al., 2008; Ishii et al., 2005; Vigny et al., 2005), 2010 M_w 8.8 Maule (Delouis et al., 2010; Y.-n. N. Lin et al., 2013; Lorito et al., 2011; Moreno et al., 2010; Vigny et al., 2011), 2011 M_w 9.0 Tohoku (Asano et al., 2011; Lay, 2018; Ozawa et al., 2012; Simons et al., 2011; Sun et al., 2014; Suzuki et al., 2011) earthquakes.

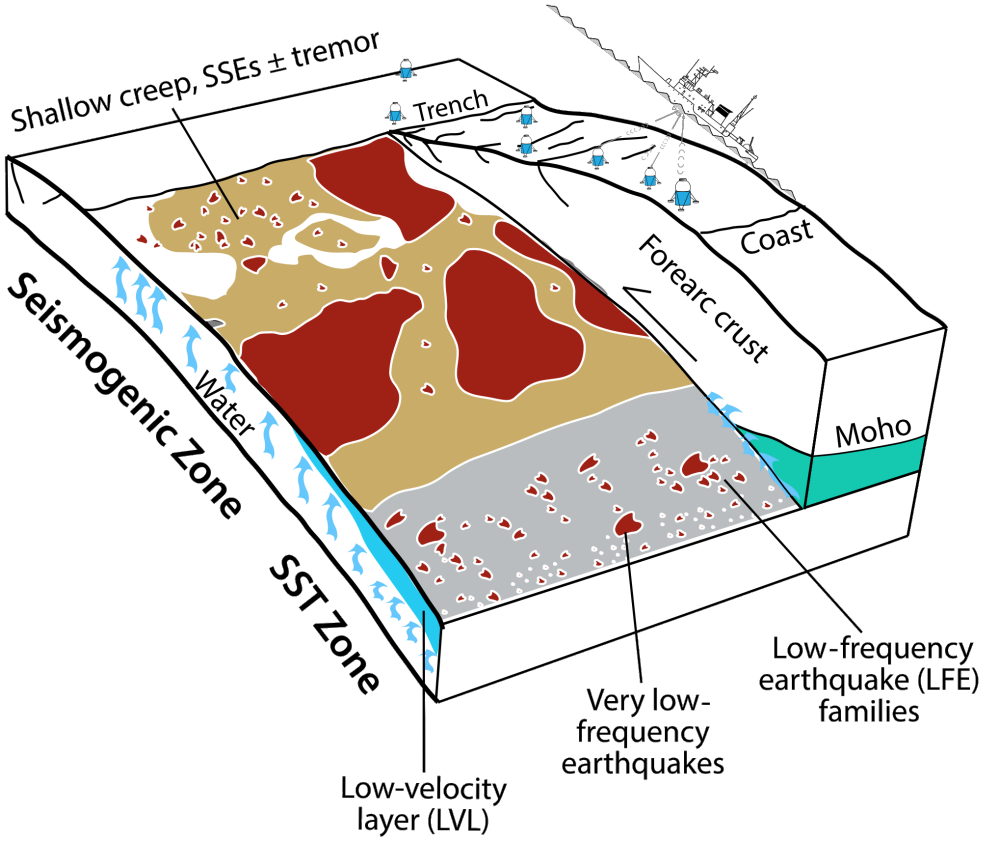


Fig. 2.2.: Slip behavior along the subduction interface, after (Behr & Bürgmann, 2021).

Subduction zones are highly dynamic geological regions on the Earth's surface. Yet, the mechanisms of deformation at the scale of the subduction still need to be fully understood, as well as their role in the seismic cycle. In the 1660s, laboratory experiments highlighted that rocks usually do not slide smoothly, but they rather exhibit an unsteady behaviour, probably due to frictional variations along the surface (Brace & Byerlee, 1966). During the sliding (loading phase), frictional stress increases (*stick*), which is abruptly released (failure) when it reaches a critical value (*slip*).

This simple model (*stick-slip*) has thus been used to explain how earthquakes occur and behave: during the *stick* phase, corresponding to the *interseismic* phase, the fault interface is loaded and frictional stress increases, to be released during the *slip* phase, which can be identified as the *co-seismic* phase. However, the slip on the fault interface is not uniform, as shown in Figure 2.2. The subduction stability (proneness to earthquakes) varies as a function of depth and temperature. At shallow depths, the stable creep seems to be associated with the presence of unconsolidated granular material (C. H. Scholz, 1998) and low pressure. Sediment consolidation as well as changes in the properties of the minerals might explain the transition zone between the stable creep zone and the unstable seismogenic zone (Marone & Saffer, 2007). In the seismogenic zone, rocks exhibit a transition from brittle to ductile behaviour, also subject to higher pressure with respect to the shallower brittle zone. At the same time, the interactions between fluids and asperities[1] along the subduction interface, as well as frictional variations, make the seismogenic zone a complex and still partially understood area. At higher depths, because of the high pressure and temperature conditions, the rocks tend to exhibit plastic behaviour, and quasi-stable creep is found. However, complex phenomena, belonging to the family of (quasi-) aseismic events, can occur until the purely stable slip region is found.

The frictional properties of the subduction interface exhibit a cyclic behaviour: stress is accumulated over a certain amount of time, during which the static friction coefficient grows. When the static friction exceeds a certain threshold, that is when the ratio between the shear stress and the normal stress exceeds a limit value, earthquakes can nucleate, which usually corresponds to a *velocity weakening* regime: the dynamic friction coefficient decreases with the amount of slip, which means that, once an earthquake starts, it will have a rapid and unbounded propagation. Since the frictional resistance decreases during the propagation, this is equivalent to a positive-feedback system that boosts the earthquake rupture. After the earthquake, new stress can build up again (*fault healing*), to be released with the next earthquake. The way stress accumulates and drops suggests a cyclic behaviour. However, this has not to be interpreted as a mechanism having a well-defined periodicity.

The aforementioned behaviour can be explained by the *Elastic Rebound* theory (Lawson & Reid, 1908). During the stress accumulation phase, plates elastically deform and, when an earthquake occurs, the stored energy is elastically released. In its original formulation, the elastic rebound theory explained the seismic cycle as a two-stage recurring process, consisting of an *interseismic phase* and a *co-seismic phase*. Starting from the late 1990s, the deployment of Global Navigation Satellite System

[1] An asperity is an area on an active fault with increased friction (locked) (Lay & Kanamori, 1981).

(GNSS) networks initiated the continuous monitoring of ground displacement, significantly contributing to modern geodesy.

Interseismic phase The interseismic phase is the longest phase of the seismic cycle, which can last from decades to centuries up to millennia (McCaffrey, 2007). During this phase, the subduction interface is locked and subject to stress loading, which results in elastic deformation of the overriding plate, which moves landwards, as seen from GNSS measurements.

Co-seismic phase The co-seismic phase corresponds to the sudden release of stress, which builds up during the interseismic phase and can last from a few seconds to a few minutes. This is accommodated through the elastic deformation of the overriding plate in the opposite direction with respect to the interseismic phase, *i.e.*, towards the trench.

The co-seismic phase has been largely studied because of the high seismic hazard associated with large earthquakes and also because it can help better understand the mechanical properties of the seismogenic zone. However, the use of Global Navigation Satellite System (GNSS) data has resulted in a change of paradigm for the study of the interseismic phase as well as for the discovery of other phenomena that enriched the spectrum of signals associated with the seismic cycle.

Post-seismic phase One of the first discoveries made possible by geodesy is the *post-seismic* phase, which was not present in the original elastic rebound theory. Heki et al., 1997 observed what they call a silent fault slip following the 1994 M_w 7.6 Sanriku-Haruka-Oki earthquake. Interestingly, the post-seismic deformation following the earthquake can help in rethinking the seismic cycle with respect to its original formulation. In fact, on one hand, the post-seismic phase further accommodates the deformation after a strong and sudden stress drop (*relaxation*), but it also corresponds to further moment release, which can have the same order of magnitude as the co-seismically-released moment, with a duration from days to years. The post-seismic relaxation can be described by three main phenomena: *afterslip*, *viscoelastic relaxation*, *poroelastic rebound*. The afterslip is mainly associated with an aseismic slip on the region surrounding the seismic rupture and is associated with the first phase of the post-seismic relaxation phenomenon. The viscoelastic relaxation is associated with the mantle and the way it responds to the co-seismic solicitation. It can last from years to several decades and can have a large-scale spatiotemporal

evolution. The poroelastic rebound is linked to pore fluid re-equilibration after the co-seismic stress changes.

2.2 The discovery of slow slip events and their implications for the seismic cycle

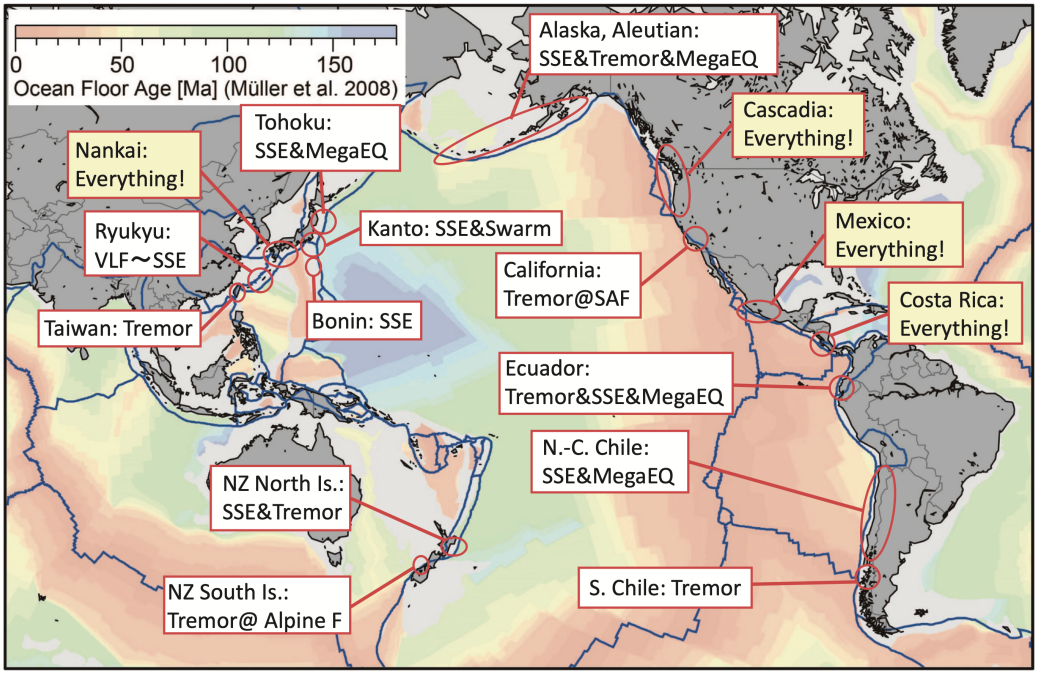


Fig. 2.3.: Worldwide occurrence of slow slip events and associated seismicity, from Obara, 2020. For each fault system, several phenomena can be observed and measured, such as slow slip events ("SSE") tremors and low-frequency earthquakes ("Tremor"), very low-frequency earthquakes ("VLF"), major earthquakes ("MegaEQ") and seismic swarms ("Swarm"). "Everything!" indicates that SSEs, tremors, LFEs and VLFEs are observed. The ocean floor age is represented in colors (Müller et al., 2008). "SAF" corresponds to the San Andreas Fault and "Alpine F." to the Alpine Fault in New Zealand. "N.-C. Chile" and "S. Chile" stay for North-Central Chile and South Chile.

The first observations of transient slip were captured by creepmeters and strainmeters, such as the post-seismic phase of the 1987 Superstition Hills earthquake (Bilham, 1989) and the slow earthquake sequence in December 1992 on the San Andreas Fault (Linde et al., 1996). Also, there is a reference to a "fault creep event" in 1975 by King et al., 1975, although the data should be interpreted with caution. A few years later, thanks to GNSS measurements, this family of slow transient events has been identified in subduction zones worldwide, such as Cascadia (Brudzinski &

Allen, 2007; Dragert et al., 2001b; Kao et al., 2009; Miller et al., 2002; Rogers & Dragert, 2003a), Nankai (Hirose et al., 1999; Hirose et al., 2023; Miyazaki et al., 2003; Obara et al., 2004; Ozawa et al., 2001), New Zealand (Douglas et al., 2005; McCaffrey et al., 2008; Wallace & Beavan, 2010), Mexico (Kostoglodov et al., 2003; Larson et al., 2004; Lowry et al., 2001a; Radiguet et al., 2011; Radiguet et al., 2012b; Vergnolle et al., 2010), Ecuador-Peru (Vaca et al., 2018; Vallée et al., 2013; Villegas-Lanza et al., 2016), Boso Peninsula (Fukuda, 2018), Alaska (Peterson & Christensen, 2009), and Costa Rica (Jiang et al., 2012), as shown in Figure 2.3. The most commonly accepted nomenclature for these events is *slow slip events* (SSEs), to denote its transient and episodic nature (see Figure 2.4 for an example).

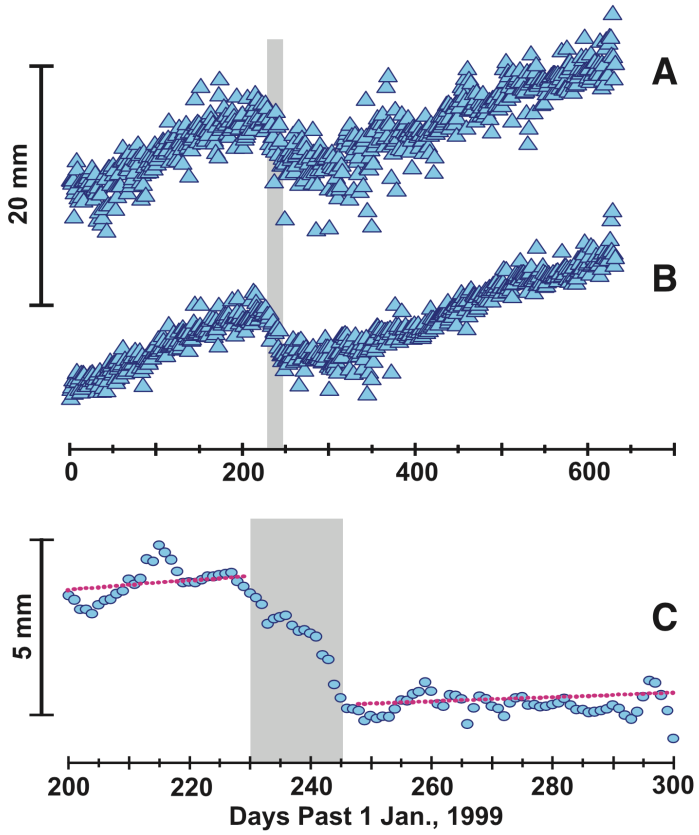


Fig. 2.4.: GNSS time series showing the 1999 Cascadia slow slip event, after (Dragert et al., 2001b). The event can be clearly identified by the reversal of surface motion in the time series. (a) shows the raw time series, (b) is obtained by a 5-day smoothing and (c) shows a zoom on the transient.

Slow slip events occur most probably on the subduction interface (e.g. Brown et al., 2009; Dragert et al., 2001b; La Rocca et al., 2009; Obara et al., 2004; Shelly, 2010; Shelly, Beroza, Zhang, et al., 2006) and have the same mechanism as earthquakes, that is shear slip on faults. However, unlike them, they last from a few days to months or years (Beroza & Ide, 2011; Ide, Beroza, et al., 2007a; Peng & Gomberg, 2010).

Since they are slowly slipping, they do not radiate detectable seismic waves (Beroza & Ide, 2011). For this reason, the exploration of SSEs can be exclusively pursued through geodesy-based approaches (*e.g.*, GNSS, InSAR) and the examination of any accompanying seismological signals. Slow slip events are often associated with non-volcanic tremors (episodic tremor and slip, ETS) (Bartlow et al., 2011; Bartlow et al., 2014; Dragert et al., 2004; Ide, Shelly, et al., 2007; Obara, 2002; Schwartz & Rokosky, 2007a; Wech & Bartlow, 2014), low-frequency earthquakes (LFEs) (Beroza & Ide, 2011; Frank, 2016; Ide, Beroza, et al., 2007a; Shelly, Beroza, Ide, & Nakamula, 2006) and seismic swarms (seismic sequences with no prevailing mainshock) (Hirose et al., 2014; Lohman & McGuire, 2007; Segall et al., 2006), suggesting that most of them occur below the seismogenic zone. Non-volcanic tremors were observed for the first time in 2002 by Obara, 2002. They are low-amplitude seismic signals with frequencies between 1 and 10 Hz. Tremors can be seen as a persistent, continuous signal, propagated by S-wave velocity, yet without clear P- and S-wave arrivals. They are usually relocated through cross-correlation techniques. Obara, 2002 determines that the observed tremors should lie at about 30 km depth, at the base of the seismogenic zone, and postulates that they could have been generated by fluid generated by the slab dehydration. Shelly, Beroza, Ide, and Nakamula, 2006 observed low-frequency signals (1-10 Hz) buried into the non-volcanic tremor signals, suggesting that tremors can be seen as the superposition of many LFEs.

Slow slip events revolutionized our understanding of plate motion. As seen in the previous section, the seismic cycle was thought of as a rather simple process where plates accumulate stress during the interseismic phase and release it during the co-seismic rupture. Geodesy, as well as more sensitive instruments, helped to shed light onto the interseismic and post-seismic phases, providing evidence of intertwined relationships between earthquakes and slow transients, as well as the associated seismological signals. While the interseismic period was considered a sill phase in between two major earthquakes, Frank, 2016; Jolivet and Frank, 2020; Jolivet et al., 2013; Linde et al., 1996; Melnick et al., 2017; Michel, Gualandi, and Avouac, 2019a; Obara and Kato, 2016; Rousset et al., 2017b; Rousset et al., 2016 showed that the interseismic loading is not constant over space and time, but is modulated by aseismic transients of deformation. However, the main challenge associated with the identification of slow slip events lies indeed in their detection in the geodetic data, since the slow slip signal goes unnoticed in the GNSS noise (Frank, 2016; Rousset et al., 2017b).

Slow slip events are characterized by a slip instability that propagates along a fault, whether spontaneous or not. As opposed to regular earthquakes, slow slip may be

thought of as being associated with a velocity-strengthening regime: once aseismic slip starts, the friction coefficient grows over time, thus the motion will rapidly arrest. However, the mechanics and governing factors of slow slip events are still poorly understood, as are their possible similarities with regular earthquakes. Hence, it is not to be excluded that slow slip events may be the spontaneous weakening of the fault zone, which cannot occur in velocity-strengthening materials. For this reason, slow slip implies velocity weakening to allow for the nucleation of an instability. Although the link between seismic and aseismic processes is still unclear, slow slip events can have magnitudes similar to regular earthquakes: they thus contribute equally to the seismic cycle. Slow slip events can be used as a proxy for the frictional state of active faults and are responsible for stress redistribution, affecting the stability of nearby asperities, and influencing or inhibiting the earthquake nucleation depending on the stress conditions and frictional state (Behr & Bürgmann, 2021; Rousset, 2019; Weng & Ampuero, 2022). In fact, several lines of evidence suggest that the nucleation, propagation, and arrest of the seismic rupture is moderated by slow slip events (e.g. Avouac, 2015; Bürgmann, 2018; Jolivet & Frank, 2020; Obara & Kato, 2016). An acceleration of the moment release before large earthquakes has been observed in geodetic data, suggesting that slow slip may trigger the nucleation of large earthquakes (Mavrommatis et al., 2014; Radiguet et al., 2016; Schurr et al., 2014; Socquet et al., 2017). Yet, geodetic observations remain rare. An acceleration of the background seismicity (Marsan et al., 2017; Marsan et al., 2013), as well as changes in the seismicity rate (Bouchon et al., 2013; Bouchon et al., 2016), have been associated with aseismic slip, yet the question remains whether aseismic slip may be the driving mechanism of gradual fault weakening prior to large earthquakes (Bowman & King, 2001), or whether a cascade of failures might trigger the mainshock (Dodge et al., 1996).

2.2.1 SSE characteristics: what is known so far?

As presented in the previous section, slow slip events occur over a few days to a few years and, although they do not radiate seismic waves, are often associated with seismicity. For example, in Nankai, slow slip events have been observed to last 1 day (Sekine et al., 2010) (short-term) as well as 5 years (Obara, 2011) (long-term). In Cascadia, SSEs are mostly episodic (short-term) and last about 20 days and are accompanied by non-volcanic tremors (Rogers & Dragert, 2003a), that are often used as a proxy for slow slip events (Wech et al., 2009). Also, SSEs occur periodically. Short-term SSEs in Cascadia seem to have a recurrence interval of about 9 months (Aguiar et al., 2009; Brudzinski & Allen, 2007), while, in Nankai,

short-term SSEs have recurrence intervals spanning from 70 to 200 days (Obara et al., 2010; Sekine et al., 2010) and long-term SSEs seem to be recurring every 10 years (Obara, 2011).

A slow slip event can be modeled as a semicircular crack that first grows updip and downdip until bounded by the transition from rate-weakening to rate-strengthening regions. Then, it propagates along strike (Dal Zilio et al., 2020). The rupture velocity on the interface is not homogeneous, that is from 1 and 10 km/day (Dal Zilio et al., 2020; Schwartz & Rokosky, 2007a). On the surface, the displacement associated with the slow slip is between a few millimeters to a few centimeters (major SSEs). Thus, the use of GNSS time series is essential to reveal the deformation associated with small slow slip events, since the millimetric precision of GNSS positioning.

Obara, 2002 observed that SSE occurrence is related to the movement of fluid generated by slab dehydration. The hypothesis is that fluids may reduce the normal stress on the interface, thus promoting fault shear slip. For example, Hawthorne and Rubin, 2013 observed that small stress perturbations, *e.g.*, tidal stressing, may modulate SSE recurrence. These stress perturbations may be driven by heterogeneities on the fault surface. Also, a high Poisson's ratio (V_p/V_s), that is the ratio between p-wave and s-wave velocities, has been observed in subduction zones near the slab interface at 30-50 km depth (see Figure 2.1) (Audet et al., 2009; Kan et al., 2023; Shelly et al., 2007). Shelly et al., 2007 find clusters of LFEs in high V_p/V_s zones, suggesting that they can be used as a proxy for slow slip manifestation. In fact, several pieces of evidence suggest that a hydrologic control might be one of the factors driving slow slip occurrence (Beroza & Ide, 2011; Frank, Shapiro, et al., 2015; Shelly, Beroza, Ide, & Nakamula, 2006).

Ide, Beroza, et al., 2007a grouped both seismic and geodetic observations to classify the novel slip mode of SSEs compared to fast (regular) earthquakes. They also group LFEs and SSEs as belonging to the same fault shearing mechanism. On top of the cubic moment-duration scaling law $M_0 \sim T^3$ typical of fast earthquakes (Kanamori & Anderson, 1975), they suggest that the moment-duration scaling law is typically linear $M_0 \sim T$. Other works also suggest that the exponent for the duration may range from 1 to 2 (e.g. H. Gao et al., 2012; Liu, 2014). Peng and Gomberg, 2010 questioned the fast and slow earthquakes subdivision and suggested that natural phenomena rather share a continuum of slip modes. They also complement the original data set from Ide, Beroza, et al., 2007a with additional geodetic observations, such as afterslip or landslides, suggesting that the gap between slow and fast slip modes in the original formulation may be due to current observation ability. Gomberg et al., 2016b suggest that a single model of slip can reconcile the observations of fast

and slow earthquakes, from bounded to unbounded growth styles as a function of the slip area. More recent observations suggest a cubic scaling law for SSEs, such as Michel, Gualandi, and Avouac, 2019b in Cascadia, Takagi et al., 2019 in Nankai, and Frank and Brodsky, 2019 in Mexico, providing further evidence that earthquakes and slow slip events may be more similar than we think. In this direction, Dal Zilio et al., 2020 point out that, even though such observations follow a cubic scaling law, SSE may likely not be associated with magnitude-invariant average rupture velocity and stress drop, as in the case of regular earthquakes. Thus, the same scaling law can arise from different physical properties, which have to be further studied.

2.2.2 SSE detection methods

As seen in section 2.2, the difficulty in identifying slow slip events in geodetic time series is related to the signal amplitude with respect to the noise. Rousset et al., 2017b use template matching to correlate GNSS time series with synthetic SSE templates obtained from the expected theoretical displacement (Okada, 1985). In order to have robust detection, they compute a weighted average of the correlation with the expected theoretical displacement to emphasize SSE displacement at stations that are supposed to record higher signal amplitude. With their method, they detect 28 slow slip events of magnitude between 6.3 and 7.2 in 2005-2014 in the Guerrero segment of the Mexican subduction zone.

Nishimura, 2014, 2021; Nishimura et al., 2013 use the Akaike Information Criterion (AIC) to detect short-term SSEs in the Japanese subduction zone. They first remove common modes from the GNSS time series via spatial filtering. They use a 180-day sliding window to compute the AIC difference between the data and two possible models: (1) either a straight line (2) or a Heaviside function centered in the window. They select the best model, *e.g.*, the model corresponding to the lowest value of AIC. They detect 133 slow slip events of magnitude between 5.5 and 6.3 in Nankai from 1996 to 2012, 130 events of magnitude between 5.8 and 6.8 along the Ryukyu trench (Japan) from 1997 to 2014, and 176 events of magnitude between 5.3 and 7 in the Kanto and Tokai regions (Japan) from 1994 to 2020.

Michel, Gualandi, and Avouac, 2019a, 2019b use the variational Bayesian Independent Component Analysis (vbICA) to first decompose the GNSS time series into Independent components. They correct the time series for all signals that do not correspond to slow slip events. They compute the slip deficit induced by slow slip and the long-term slip deficit related to the coupling, by focusing on periods of negative slip deficit rate. They consider having a detection when the slip deficit rate

is less than 40 mm/year. They detect 40 slow slip events in Cascadia from 2007 to 2017 of magnitude from 5.3 to 6.8.

2.3 Seismo-tectonic context

In this manuscript, we mostly focus on the detection and characterization of slow slip events along the Cascadia subduction zone, which we present in section 2.3.1. In chapter 3, we also address the problem of earthquake characterization in Honshu, North Japan, which we briefly present in section 2.3.2.

2.3.1 Cascadia subduction zone

The Cascadia subduction zone is located in North America. The subduction margin extends for about 1100 km, from Vancouver Island (Nootka Fault) to north California (Mendocino triple junction), as shown in Figure 2.5. Cascadia is a case of warm and young (< 10 Ma) subduction margins. Here, the Juan de Fuca plate subducts beneath the North American plate with a convergence rate between 30 and 40 km per year (DeMets & Dixon, 1999; Hyndman et al., 1997; McCaffrey, 2007). Over the past $\sim$ 10,000 years, the Cascadia margins have experienced several megathrust earthquakes, as inferred from turbidite sediments and coastal geological evidence (Goldfinger et al., 2017; Goldfinger et al., 2012), with magnitudes up to 9 (Leonard et al., 2010). In 1700, the last M_w 9 Cascadia megathrust earthquake occurred, which produced tsunami waves, recorded in Japan (Satake et al., 2003). At the present time, the sole seismologically active zones are the northern and southern segments (see Figure 2.5) (K. Wang & Tréhu, 2016). According to Obana et al., 2015, the absence of interplate seismic activity may indicate that the megathrust has fully healed and locked since the 1700 event, suggesting a relatively lower level of structural heterogeneity, such as subducting seamounts.

Episodic slow slip events have been observed along the margins of the Cascadia subduction zones, most of them accompanied by non-volcanic tremors, LFEs and VLFEs (Calvert et al., 2020; Chaudhuri & Ghosh, 2022; Ghosh et al., 2015; Rogers & Dragert, 2003a; Wech et al., 2009). Calvert et al., 2020 show that in many subduction zones, such as Cascadia, a layer of anomalously low seismic wave velocities has been observed near the episodic-tremor-and-slip area, suggesting that high pore-fluid pressure may weaken the megathrust (see Figure 2.1 and the detail

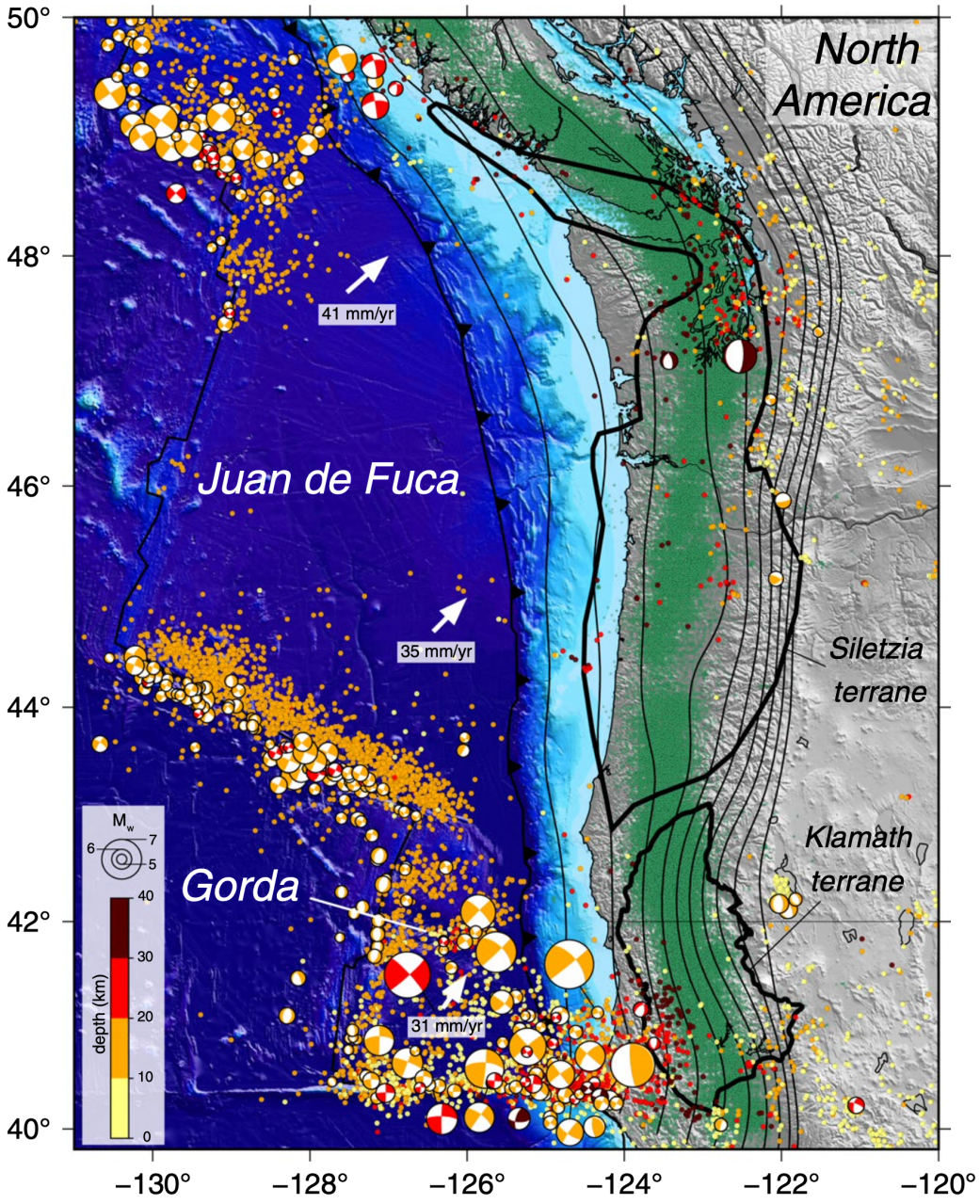


Fig. 2.5.: Tectonic setting of the Cascadia subduction zone, from McKenzie et al., 2022. Seismicity from 1976 to 2021 is shown ($M_w > 3$). The MORVEL (Mid-Ocean Ridge VELocity) Juan de Fuca-North America plate motions are shown (DeMets et al., 2010). $M_w > 5$ earthquakes are shown with their focal mechanism and are color-coded by depth (USGS and GCMT earthquake catalogs). Green transparent onshore points show tremor locations from 2016 to 2021 from the Pacific Northwest Seismic Network (PNSN). 10 km slab contours (Hayes et al., 2018) are shown as thin black lines.

of the fluid-saturated oceanic crust) (Calvert et al., 2020; Frank, Shapiro, et al., 2015; Kan et al., 2023).

The ETS zone in the subduction presents an along-dip segmentation of slip and it disconnected from the seismogenic zone, acting as a buffer zone that isolates the seismogenic and the SSE zone, and temperature and lithological constraints might explain the existence of two separate zones of unstable frictional sliding (X. Gao & Wang, 2017; Michel, Gualandi, & Avouac, 2019a). Michel, Gualandi, and Avouac, 2019a find that the locking zone is closer to the coastline, at depths between 10 and 30 km, where the slab is supposed to be creeping. Also, the absence of seismicity in this zone suggests that the interseismic loading has not yet compensated for the stress drop due to the 1700 megathrust earthquake and also that interplate seismicity may penetrate at depths deeper than the locked zone.

2.3.2 Tectonic context of Honshu, North Japan

At the intersection of the Japan archipelago, four tectonic plates converge: the North American plate (NAM), the Pacific plate (PAC), the Philippine Sea plate (PHS), and the Eurasia plate (EUR), as shown in Figure 2.6 (EUR is not present in the figure). The PAC plate subducts beneath the NAM plate along the Japan Trench with a convergence rate of 76 mm/yr. The PHS plate subducts beneath the NAM plate along the Sagami Trough at 26 mm/yr. Along the Nankai Trough, the PHS plate subducts beneath the EUR plate at 56 mm/yr, and the EUR plate subducts beneath the NAM plate along the Japan Sea Margin at 22 mm/yr. The plate convergence rates are taken from Marill et al., 2021.

The PAC/NAM and PHS/NAM have hosted the greatest earthquakes in Japan, such as the 1994 M_w 7.7 Sanriku earthquake (A. Ito et al., 2004), the 2003 M_w 8 Tokachi earthquake (Yamanaka & Kikuchi, 2003) and the 2011 M_w 9 Tohoku earthquake (Hooper et al., 2013) as for the Japan Trench, and 1703 M_w 8.2 and 1923 M_w 7.9 Kanto earthquakes (Komori et al., 2017; Shishikura, 2014) and the 1946 M_w 8.4 Nankai earthquake (Baba et al., 2002) as for the Sagami Trough.

In chapter 3, we focus on the Japan Trench. The PAC/NAM subduction is colder and older (130-140 Ma, from Müller et al., 2008) than the Cascadia subduction zone (see section 2.3.1). It presents along-strike structural heterogeneity and a complex segmentation of slip, different than the along-dip segmentation of the Nankai Tough and the Cascadia subduction zone, where ETS is found. In the Japan Trench, no regular SSEs are observed, which do not seem to be correlated with any

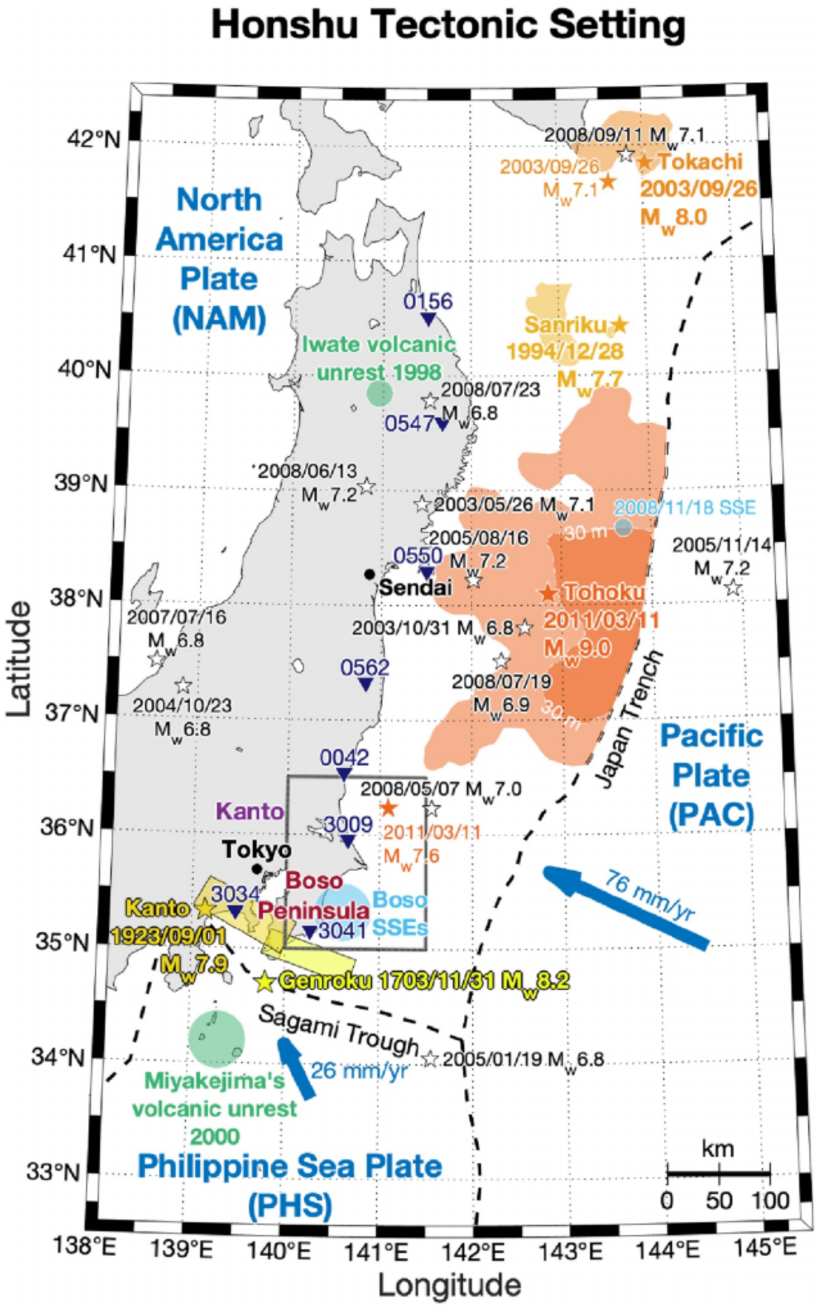


Fig. 2.6.: Tectonic setting of the Honshu island, from Marill et al., 2021. The colored stars represent historically and instrumentally recorded great earthquakes ($M_w > 7.7$) as well as major aftershocks (rupture areas are also indicated by the same color as the corresponding stars) (Hooper et al., 2013; A. Ito et al., 2004; Komori et al., 2017; Shishikura, 2014; Yamanaka & Kikuchi, 2003). The light blue circle near the Boso peninsula shows the Boso SSE rupture area (Fukuda, 2018), and the one at 38°N 143°E represents the 2008 SSE (Y. Ito et al., 2013). The green circles show the 2000 Miyakejima (Cattania et al., 2017) and the 1998 Iwate volcanic unrest (Miura et al., 2000). Plate motions are deduced from Nishimura et al., 2007 Euler's poles.

seismological signals. While there are a few observations of SSEs in the Northern and Southern parts of the Japan trench, the central segment presents a lack of observations, probably due to less pore fluid (Nishikawa et al., 2019).

2.4 Fundamentals of machine learning

When we talk about Machine learning, we refer to mathematical and statistical methods employed to "learn" specific patterns from data to solve a specific problem. In this manuscript, we will refer to machine learning (and deep learning, in the following sections) models to indicate *supervised* learning methods, which are one of the learning strategies. In supervised learning, the data is considered to be a set of pairs $\{x_i, y_i\}$, where x_i is an *input* sample and y_i is the corresponding desired *output*. A simple high-level description is provided by the following equation:

$$y_i = f_\Theta(x_i) \tag{2.1}$$

where the function $f_\Theta(\cdot)$ is the representation of a specific model. The subscript Θ represents the parameters of the model. A machine learning model takes a set of inputs x_i and optimizes (*learns*) the model parameters Θ to build a hidden function f_Θ that (ideally) maps each input to the corresponding output. In most machine learning methods, the training can be thought of as an iterative process where the parameters Θ are optimized according to some loss or error metrics (*e.g.*, mean squared error), as described in the following pseudo-code:

```
function training(x, y, N) {
    f_theta = f_0
    for iteration in 1, ..., N {
        y_pred = f_theta(x)
        error = E(y, y_pred)
        f_theta = update_parameters(f_theta, error)
    }
}
```

Listing 2.1: High-level supervised training procedure.

where N is the number of iterations, E is the chosen error metric (also called *loss function*) and the function `update_parameters` updates the model parameters Θ according to the error (*misfit*) at the current iteration. A large number of machine learning models are optimized according to versions of the gradient descent algorithm. In this case, the parameters are updated as:

$$\Theta_t \leftarrow \Theta_{t-1} - \eta \nabla E(\Theta_{t-1}) \tag{2.2}$$

where t and $t-1$ indicate the current and the previous iterations and $\eta > 0$ is the *step size* or *learning rate*.

In addition to the parameters Θ, a model usually also has *hyperparameters*. While parameters Θ are optimized during training, hyperparameters are linked to the structure of the function f and the algorithm configuration. For instance, the learning rate η, the number of iterations N and the general form or architecture of f are hyperparameters and they need to be adjusted in order to have the best performance. They are usually optimized through an experimental process called *hyperparameter tuning*.

Once the best hyperparameters and parameters Θ^* are selected, the predictions are computed as:

$$\hat{y} = f_{\Theta^*}(x). \tag{2.3}$$

To be noted, testing and assessing the performance of the trained model on the samples used for *training* the model can lead to a biased estimate since the model parameters have been optimized on this data specifically. Yet, testing on training data does not tell anything about the generalization ability of the model, *i.e.*, the ability to perform reasonably well over **unseen** data. The same logic applies to the hyperparameter tuning, which should be performed on a separate data set for the tuned model to generalize better over new data. For these reasons, the data is usually split into three *independent* sets:

- **training data**. The data that is used during the *training phase*. It is used to compute the training error or misfit E, serving to iteratively optimize the model's parameters Θ.

- **validation data**. The data that is used to tune the model's hyperparameters.

- **test data**. The data that is used to assess the performance of the trained model during the inference phase.

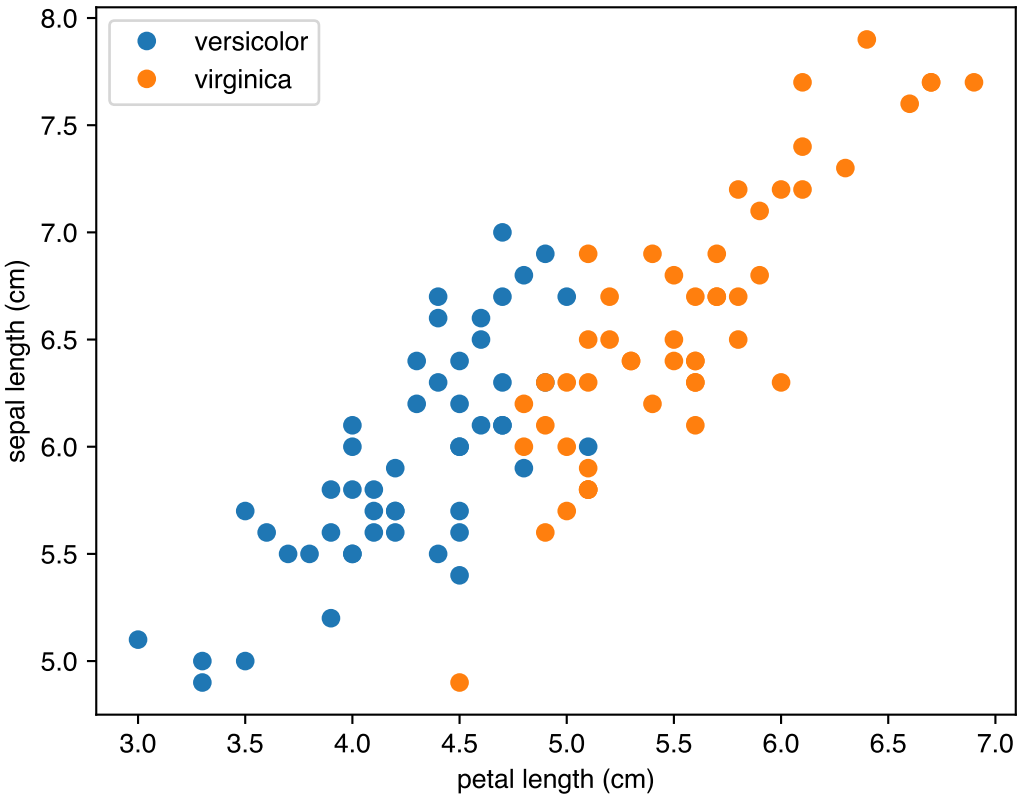


Fig. 2.7.: Example of a subset of the iris data set. Each point represents the abstraction of an iris *sample*, as a function of two features: the petal length (x-axis) and the sepal length (y-axis). Blue and orange points represent two different types (classes) of sample: versicolor and viriginica, respectively.

2.4.1 A real-world problem addressed by machine learning

In 1936, R. Fisher introduced the iris flower data set. This data set consists of 50 samples of three different species of iris flower. For each flower, four structural properties were collected: length and width of petals and sepals. These four properties are commonly called *features*. Features represent an abstract representation of the problem of interest and are supposed to contain all the relevant information to address the task under consideration. Each sample can thus be represented as a point in a vector space, that we call *feature space*. In our case, each flower is a point in a four-dimensional space. For simplicity, suppose that we only retain two features, the petal length and the sepal length, as shown in Figure 2.7. We also assume here that we only have two species (*classes*) in the data set: versicolor and virginica.

Suppose that we want to train a machine learning model to classify the iris species. Such a model will receive some pairs (x_p, x_s) in input, where x_p and x_s are the petal and sepal lengths, respectively, and will have a binary output $\hat{y}$, either "versicolor" or

"virginica". We can encode these categorical output variables into numerical values, such as 0 (for versicolor) and 1 (for virginica). We can use a simple model (also called *logistic regression*) to discriminate between the two classes. The model first computes a hypothesis z on the input data:

$$z = \Theta \cdot x = \theta_0 + \theta_1 x_p + \theta_2 x_s \tag{2.4}$$

and then computes a prediction $\hat{y}$ as:

$$\hat{y} = f_\Theta(x) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.5}$$

where $\Theta \cdot x$ indicates the scalar product between Θ and x and σ is the sigmoid function. The model has thus three parameters $(\theta_0, \theta_1, \theta_2)$ that will be optimized during the training by minimizing a specific loss function.

Suppose that, after the training, the parameters $\Theta^* = (\theta_0^*, \theta_1^*, \theta_2^*)$ have been optimized to the values $(-15.29, 3.75, -0.48)$. The hypothesis $z^* = \Theta^* x = 0$ corresponds to a straight line in the feature space that acts as a *decision boundary* (corresponding to $\hat{y} = \frac{1}{2}$):

$$\theta_0^* + \theta_1^* x_p + \theta_2^* x_s = 0 \tag{2.6}$$

as shown in Figure 2.8. When $z^* > 0$, the model will classify a sample as *virginica*, *versicolor* otherwise. In this case, a new iris having a petal length of 5 cm will be classified as *virginica* if the sepal length $x_s > -\frac{\theta_1}{\theta_2} x_p - \frac{\theta_0}{\theta_2} = 7.1$ cm.

In general, a machine learning model $f_\Theta(\cdot)$ is fed some input features and outputs one or more variables corresponding to the predicted output. For this reason, (supervised) machine learning can be resumed as "learning from examples". Two types of machine learning problems exist:

- **classification**: the target is a discrete variable (*e.g.*, 0, 1, ..., n) which is not ordered, as in the case of iris species classification;
- **regression**: the target is a continuous variable, *e.g.*, the estimation of the total size of the plant, or the altitude at which the plant was found, etc.

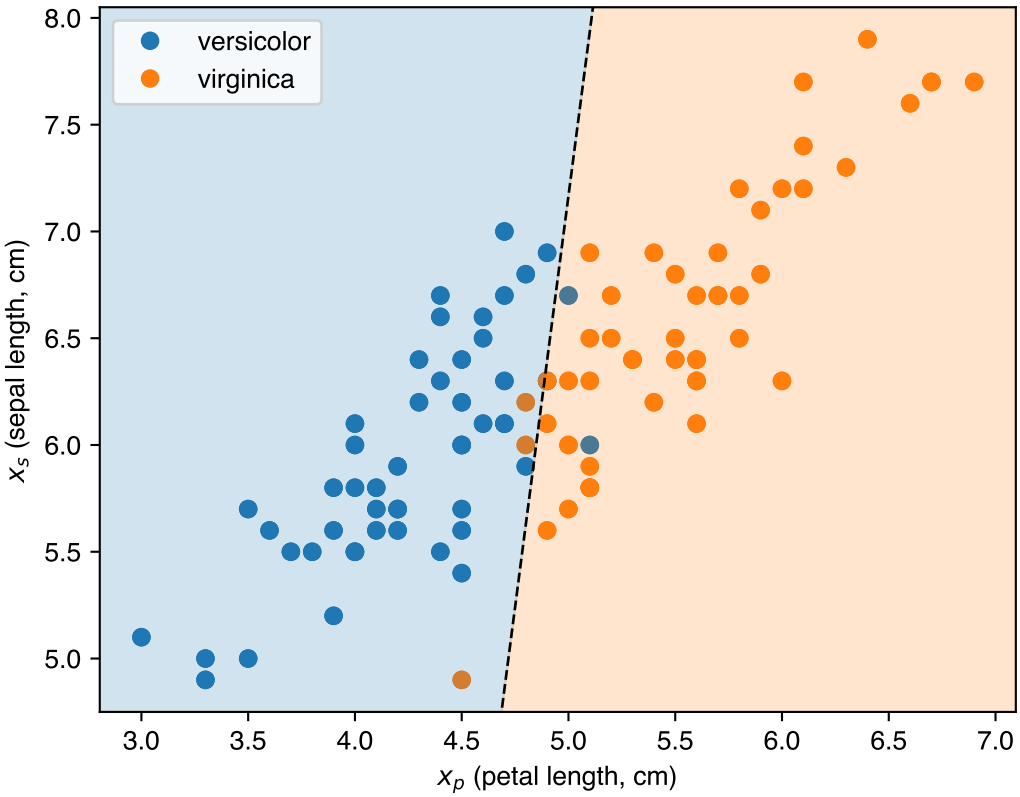


Fig. 2.8.: Example of linear classification of two iris species using equation 2.5 as the decision rule. Each point represents an iris sample as a function of the petal length (x-axis) and the sepal length (y-axis). Blue and orange points represent two different classes. The dashed straight line indicates the decision boundary (equation 2.8 and the blue and orange regions indicate when a sample will be classified as versicolor and virginica, respectively.

On top of deep learning models, which will be developed later, many other machine learning methods have been found to be performing well when features can be seen as a list of variables (such as in the Iris dataset): we can note, in particular, random forests (Breiman, 2001) and gradient boosted trees, *e.g.*, XGBoost (Chen & Guestrin, 2016).

2.4.2 Machine learning in geosciences

Machine learning has recently gained popularity in geosciences thanks to successful applications, notably in seismology and earthquake engineering. Among them, Zou et al., 2021 use attributes (features) extracted from seismic data to infer the subsurface porosity. In the frame of earthquake engineering and hazard assessment, Mangalathu et al., 2020 compare several machine learning methods to assess the

spatial extent and the severity of building damage, Y. Zhang et al., 2018 assess the post-earthquake building structural safety using sequential ground motions, and Ghimire et al., 2022 focuses on post-seismic damage on urban/regional scale after the 2015 M_w 7.8 Gorkha Nepal earthquake.

In seismology, Hibert et al., 2017 use the random forest method to automatically classify rockfalls and volcano-tectonic earthquakes at the Piton de la Fournaise volcano. Soubestre et al., 2018 use a network-based method for detecting and classifying seismovolcanic tremors by exploiting the daily array covariance matrix. Li et al., 2018 use the feature extracted from earthquake and noise waveforms by a Generative Adversarial Network (GAN) to train a random forest model targeted to seismic wave discrimination. P. Shi et al., 2021 apply unsupervised machine learning techniques to better understand the evolution of seismic wavefield properties, from wavefield features and covariance matrix analysis. Aden-Antoniów et al., 2022 use an adaptable random forest model for the declustering of earthquake catalogues. Rouet-Leduc et al., 2019 use the random forest model applied to seismic recordings to infer the GNSS displacement rate throughout the slow slip cycle and Hulbert et al., 2020 use gradient-boosted trees to infer slow slip timing from statistical features extracted from seismic waveforms.

In the frame of laboratory earthquakes, Rouet-Leduc et al., 2017 use the random forest model to predict the time before a failure by using acoustic emissions, Bolton et al., 2019 use unsupervised machine learning to identify patterns in the acoustic signals and precursors to labquakes. Rouet-Leduc et al., 2018 use gradient-boosted trees to estimate the fault friction from continuous seismic signals. Hulbert et al., 2019 use the same method to predict the timing and duration of laboratory earthquakes. Asim et al., 2017 test several machine learning models, such as tree-based methods and random forest, to predict the earthquake magnitude.

2.5 Towards deep learning: neural networks

many problems have complex data as input, that are not easily convertible to a list of features: we can think of an input being a sequence (of text, of time points such as a time series), or an image. The deep learning family of methods has the advantage to incorporate feature extraction in the model: this is what made them very effective.

The linear model $y = f_\Theta(x) = \Theta \cdot x$ presented in the previous sections can be thought of as the building block of more sophisticated models, typically called (artificial)

neural networks. Neural networks owe their name to biological neural networks because of their complex structure, which resembles the interconnections between biological neurons.

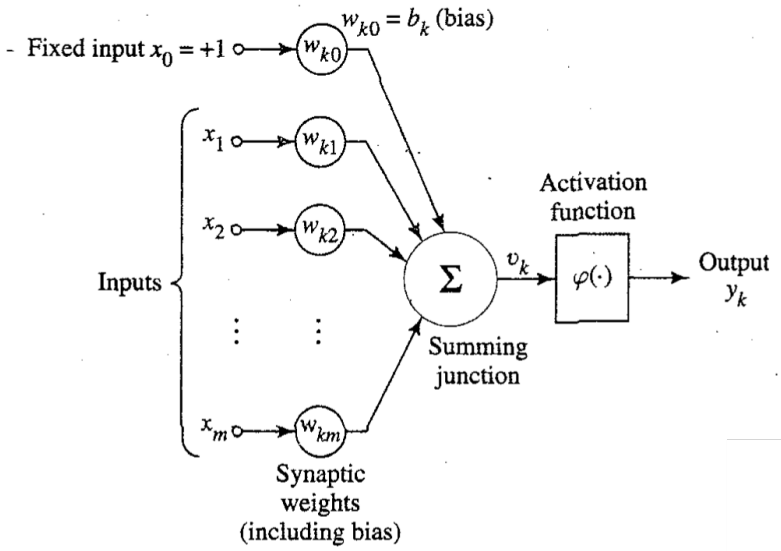


Fig. 2.9.: Example of artificial neuron, from (S. Haykin, 1998).

An artificial neuron is a (generally nonlinear) processing unit and its structure is presented in Figure 2.9. A neuron basically performs a logistic regression: a linear combination of the inputs followed by a nonlinear *activation function* φ, *e.g.*, a sigmoid function:

$$y = f_\Theta(x) = \varphi\left(\sum_{i=1}^{n} \theta_i x_i + \theta_0\right) \tag{2.7}$$

where n is the input size. Equation 2.7 can be made more compact by transforming the input $x = (x_1, x_2, ..., x_n)$ into $(1, x_1, x_2, ..., x_n)$ such that the bias can be embedded in the scalar product:

$$y = f_\Theta(x) = \varphi\left(\sum_{i=0}^{n} \theta_i x_i\right) \tag{2.8}$$

The activation function can be, in principle, any differentiable nonlinear function. If φ is chosen as a sigmoid, this is equivalent to equation 2.5. The weights Θ, as before, represent the model parameters and are optimized during training.

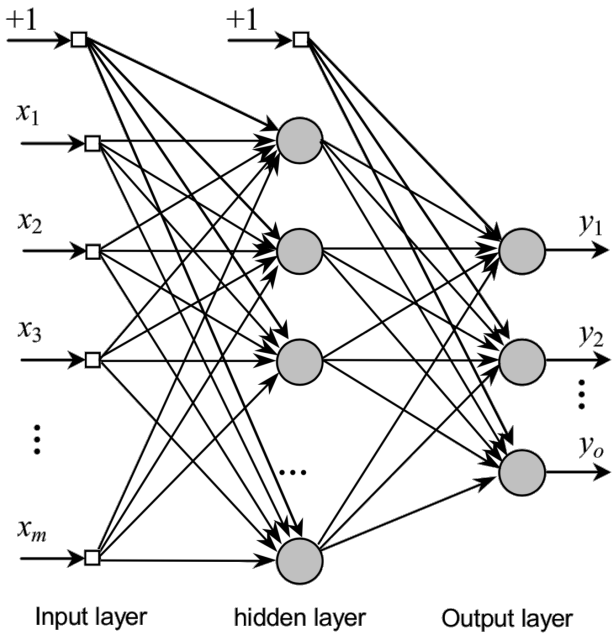


Fig. 2.10.: Example of multi-layer perceptron, edited after (De Castro, 2006).

In neural networks, neurons are connected to each other, as shown in Figure 2.10, showing an example of a neural network with one hidden layer[2]. In this case, the model structure $f_\Theta(\cdot)$ consists of an input layer, a *hidden layer* and an output layer and is described by:

$$y_k = f_\Theta(x) = \varphi\left(\sum_{j=0}^{m} \theta_{k,j}\, \varphi\left(\sum_{i=0}^{n} \theta_{j,i} x_i\right)\right) \tag{2.9}$$

where n is the number of inputs and m is the number of neurons in the hidden layer.

As seen in Figure 2.8, a single neuron can express the decision boundary for classification[3] as a hyperplane. When combining multiple layers (*e.g.*, more hidden neurons), as in equation 2.9, the decision boundary can become more complex (intuitively, the composition of several hyperplanes can result in a nonlinear shape), which can help better separating the different classes. In the case of the iris classification, a curve could have helped to better separate the two classes instead of a straight line.

[2]We refer to a layer as a group of neurons.
[3]Here we talk about classification, yet the same states for regression tasks (*e.g.*, nonlinear function approximation).

A remarkable theoretical result achieved for neural networks is known as *universal approximation theorem*, whose informal definition is provided below.

Theorem $\forall \varepsilon > 0, f(x), \exists$ a neural network $f_\Theta(x)$ such that

$$\|f(x) - f_\Theta\|_\infty < \varepsilon. \tag{2.10}$$

The universal approximation theorem, however, states that a neural network exists that solves any task. Yet, nothing is said about how difficult it might be to train, nor what are its hyperparameters and how to optimize them (*e.g.*, number of hidden neurons). Also, a deeper neural network, *i.e.*, with more hidden layers may be easier to train and lead to better performance.

For many years, 3-layer neural networks (also called multilayer perceptrons, MLP) have been used, with limited emphasis on deeper architectures. Reconsidering deep networks, with proper structural modifications, led to deep learning.

2.6 Fundamentals of deep learning

The universal approximation theorem, as seen in the previous section, does not provide any guarantee that a "shallow" network would achieve better performance than a deeper one with different hyperparameter configurations. Also, the way those models work assumes the presence of features previously extracted from data (*feature engineering*). During the training phase, they can reproject the data onto a (typically) higher-dimensional space, hence why the universal approximation theorem, yet they cannot enrich nor learn a new representation for that data. Also, the choice of features is one of the most important tasks and the selection of the right one is not trivial. In fact, the feature selection is usually delegated to experts in the specific domain. Once specialists have selected a good set of features, machine learning experts can rely on those to build specific models. However, this process could need to be reiterated because of domain-related caveats that need to be taken into account.

One of the breakthroughs empowered by deep learning is the possibility to build models that are trained on *raw data*, instead of features obtained from it. In this way, these models can extract knowledge from the data and, most intriguingly, they can distillate the best features according to the data itself and the learning task

(*representation learning*). This has been enabled by the availability of large data sets, making deep learning one of the state-of-the-art frameworks for statistical learning.

2.6.1 Convolutional Neural Networks

One of the most employed deep learning models today is Convolutional Neural Networks (CNNs), which make use of convolution as the main operation. This form of deep learning is particularly relevant for large but *ordered* input data, such as images or time series. Given two real discrete signals $x(n)$ and $y(n)$, with $n \in \mathbb{Z}$, their discrete convolution is defined as:

$$(x * y)(n) = \sum_{u=-\infty}^{\infty} x(u)y(n-u) \tag{2.11}$$

Suppose to have a one-dimensional kernel $K(m)$ having values in a finite domain $m \in (-M, ..., -1, 0, 1, ..., M)$. The size of the kernel is $2M + 1$ and is assumed odd for convenience. The convolution between x and K can be rewritten as:

$$(x * K)(n) = \sum_{m=-M}^{M} K(m)x(n-m) \tag{2.12}$$

where the commutative property was applied.

The idea behind this is to slide the kernel on the signal and cross-correlate[4] the two, with the kernel values (weights) being learned during training.

One of the strengths of deep learning, enforced by convolutional networks, is the **weight sharing**: the convolution weights are shared for all the input values, as shown in Figure 2.11 (top). This results in fewer computations with respect to a *fully-connected* (equivalent to an MLP, see the previous section and Figure 2.10) configuration, as shown in Figure 2.11 (bottom). This is also motivated by the arrangement of neurons in the visual cortex of mammals, which inspires CNNs. A given set of N different kernels $\left\{K_i(n)\right\}_{i=1}^{N}$ can be convoluted with the inputs $x(m)$ to obtain a set of signals $\left\{h_i(n)\right\}_{i=1}^{N} = \left\{(x * K_i)(n)\right\}_{i=1}^{N}$. The signals $h_i(m)$ are typically called *feature maps* and can be further transformed by an activation function, as:

[4]Cross-correlation becomes a convolution when one of the two signals is not flipped.

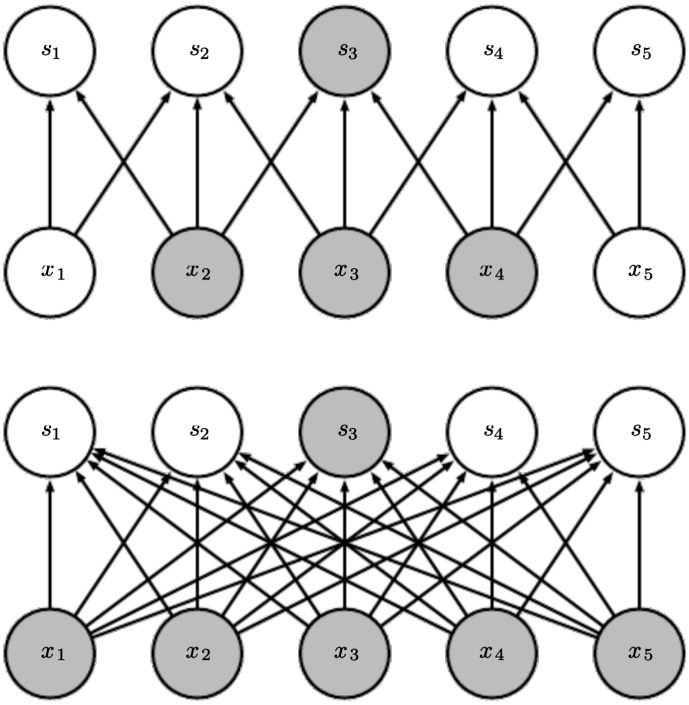


Fig. 2.11.: Example of *weight-sharing*, after (Goodfellow et al., 2016). (top) a neural network with weight sharing, (bottom) a fully-connected neural network.

$$h_i(m) = f_{K_i}(m) = \varphi\left((x * K_i)(n)\right) = \varphi\left(\sum_{m=-M}^{M} K_i(m)x(n-m)\right). \tag{2.13}$$

Although equation 2.13 is quite similar[5] to equation 2.8, there is a substantial difference between a fully-connected layer and a convolutional layer. Since a neuron is connected to all the previous neurons in a fully-connected configuration, the number of parameters (weights) to be learned increases quadratically with the input (or hidden-layer) size. In convolutional networks, the number of learnable parameters is proportional to $(2M+1)N$, that is the kernel size multiplied by the number of feature maps. Since the filter size is usually small, the number of parameters depends mostly on the number of features computed in a layer. This also tells us that, contrary to fully-connected networks, convolutional networks can process inputs of arbitrary dimensions, to parity of parameters: this makes CNN flexible to many types of data (*e.g.*, satellite images or seismic recordings).

The convolution is a **local** operation. However, a sequence of more convolutional layers can increase the **receptive field**, as shown in Figure 2.12. As a result,

[5]We tried to be consistent with equation 2.3. It should be noted that the parameter vector Θ is now replaced with the kernel weights K_i for the i-th feature map.

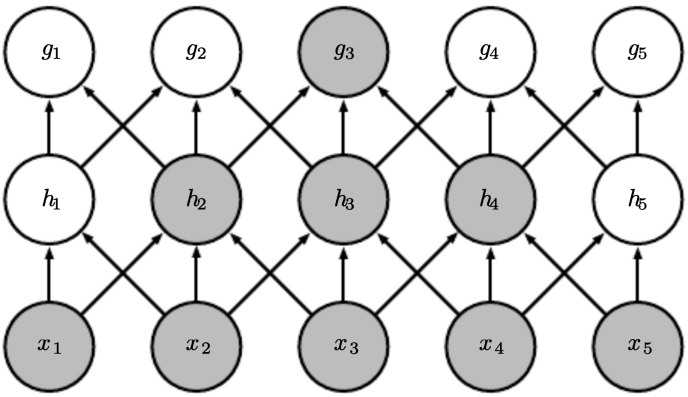


Fig. 2.12.: Example of the receptive field in a CNN, after (Goodfellow et al., 2016).

deeper convolutional networks can leverage long-range information. Also, because the convolution filters are learned, a convolutional network can extract low-level information first (*e.g.*, edges, corners, as in the case of images, or peaks, spectral information in the case of 1D signals), all the way to more complex and abstract characteristics (*e.g.* object parts, shapes, textures as for images, or temporal-spectral relationships, events, as for 1D signals) as the depth of the network increases. For this reason, CNNs are also called *feature extractors* or *feature detectors*.

Convolutional networks make extensive use of **pooling** operations, which are novel components introduced in deep learning. The idea of pooling is to replace a local neighborhood of the input (or hidden representations) with summary statistics (Goodfellow et al., 2016), such as max-pooling or average-pooling. A pooling kernel slides onto the data and outputs the maximum or the average value at a certain location. With a kernel of size 2, the output will have half the length of the input. This can be seen as a downsampling operation, which can result in improved computational efficiency since the network will have half the samples to process in the next layer. Thanks to downsampling, convolutional neural networks can capture multi-scale features, since the result of convolutions made with a kernel of a given size applied to smaller inputs can be seen as the same as sliding a bigger filter to the original input. With pooling, the filter size can remain smaller, with the advantage of having a larger receptive field, thus improving feature extraction by computing features at both local and global scales. Moreover, pooling kernels do not have any trainable parameters, therefore they do not add any complexity to the network.

2.6.2 Recurrent neural networks

Recurrent Neural Networks (RNNs) are a class of deep learning models specifically designed to analyze *sequential data*, *i.e.*, data in which samples are ordered in time. For example, the value of a time-varying signal $x(t)$ at time t is generally dependent on (possibly all) the previous samples:

$$x(t) = f(x(t-1), x(t-2), ...). \tag{2.14}$$

A typical example of a sequence is text. In sentences, the semantics associated with a given word is dependent on the previous words. Hence, to understand the meaning of a sentence, it is usually necessary to read it entirely.

In order to model this desired behaviour, RNNs introduce the idea of *state* $h(t)$, which is an intrinsic variable of the RNN system and can be thought of as a "memory" about past information. The idea behind this is that the sequence is parsed and, for each sample, the internal state of the system is *updated*:

$$h(t) = f\left(h(t-1), x(t); \Theta\right). \tag{2.15}$$

This equation can be thought of as the update rule of the internal state of an RNN, based on the current input and the previous state, which is conditioned on past inputs and past state values. The basic RNN structure (or "cell") can be given by a simple linear update and the output for the *k*-th neuron can be written according to the following (simplified) equation (Goodfellow et al., 2016):

$$y_k(t) = \varphi_1\left(w_o \varphi_2(w_h h_k(t-1) + w_i x_k(t)\right) \tag{2.16}$$

where φ_1 and φ_2 are nonlinear activation functions, w_o, w_h, w_i are learnable weights associated with hidden-output, hidden-hidden and input-hidden connections, respectively.

No details will be provided here about how to train recurrent networks. However, it should be noted that the kind of linear state update ($h_k(t) = w_h h_k(t-1)$) provided in equation 2.16 can raise some stability issues, especially for long sequences, where the network parameter update process can be harmed by unstable gradient propagation. Because of this, modern recurrent architectures integrate a *gating* mechanism to modulate the flow of information: long short-term memory (LSTM)

and gated recurrent units (GRU). Here, the GRU architecture will be presented, which is simpler than LSTM and will be used in the following chapters.

2.6.2.1. Gated Recurrent Units

The idea of gated RNNs such as LSTMs or GRUs is that the network can select how much information to retain in the state, how much to discard, and how much to propagate to the next stage. As a simple analogy, the working mechanism of such RNNs is similar to water, which can flow or be shut off by means of faucets during its flow. These faucets, or gates, are **learned** during the training, which means that the network can learn how to adaptively filter out information.

The GRU is equipped with two types of gates: an *update* gate $\mathbf{u}(t)$ and a *reset* gate $\mathbf{r}(t)$, defined as:

$$\mathbf{u}(t) = \sigma\left(\mathbf{W}_u \mathbf{x}(t) + \mathbf{U}_u \mathbf{h}(t-1)\right) \tag{2.17}$$

$$\mathbf{r}(t) = \sigma\left(\mathbf{W}_r \mathbf{x}(t) + \mathbf{U}_r \mathbf{h}(t-1)\right) \tag{2.18}$$

where σ is the sigmoid function, and $\mathbf{W}_u, \mathbf{U}_u, \mathbf{W}_r, \mathbf{U}_r$ are learnable weight matrices. A *candidate hidden state* can be defined, which is the hidden state subject to the reset gate, as:

$$\tilde{\mathbf{h}}(t) = \tanh\left(\mathbf{W}_h \mathbf{x}(t) + \mathbf{U}_h \left(\mathbf{r}(t) \odot \mathbf{h}(t-1)\right)\right) \tag{2.19}$$

where $\odot$ represents the Hadamard (or element-wise) product, and $\mathbf{W}_h, \mathbf{U}_h$ are learnable weight matrices. Finally, the update rule for the hidden state, subject to both gates, is defined as:

$$\mathbf{h}(t) = \mathbf{u}(t) \odot \mathbf{h}(t-1) + (1 - \mathbf{u}(t)) \odot \tilde{\mathbf{h}}(t). \tag{2.20}$$

With this rule, the state is updated both by propagating a fraction of the previous state and by conditioning the previous state with current inputs and the previous state. The network learns how to find a balance between the two gates, which simultaneously controls how much to forget about the previous state and how to update the decision on the current state. The idea is that the network can learn

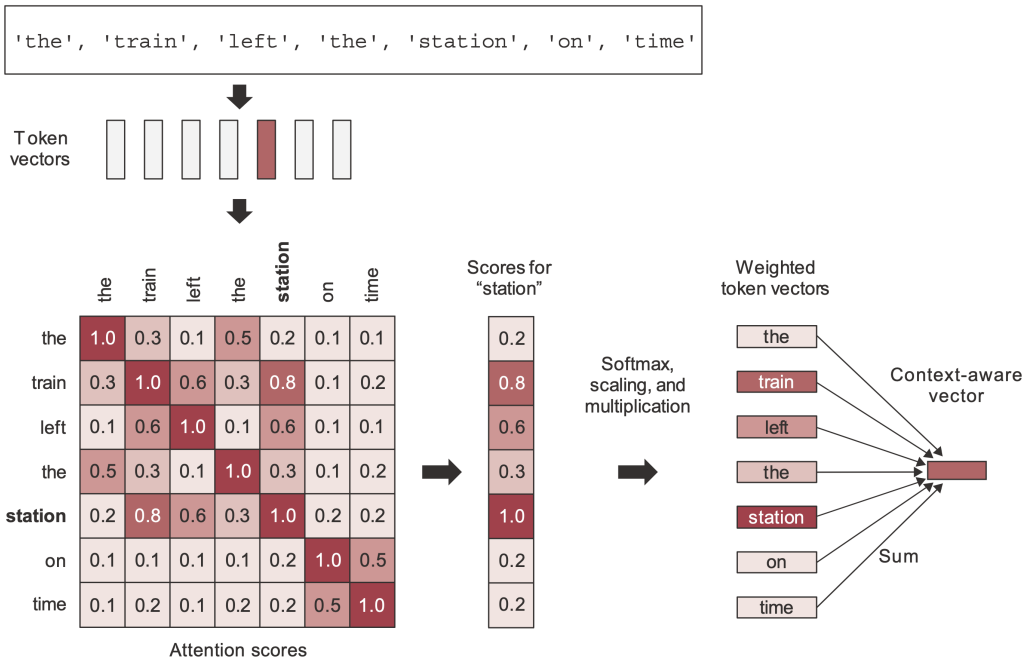


Fig. 2.13.: Example of self-attention, after (Chollet, 2021).

to create paths between the input sequence and the state update, where only the most relevant information is forwarded and retained, enabling the model to capture long-term dependencies in the data more effectively.

2.6.3 Transformer neural networks

Recurrent neural networks can be effective at modeling sequences, such as time series or text. However, when the sequence is long, some issues arise, which are not fully addressed by LSTMs or GRUs. For example, when modeling a long text sequence, we may wonder whether scanning the entire sentence is useful to detect long-term dependencies between words. In fact, a word at the end of the sentence may only be related to the context provided at the beginning of the sentence. Transformers introduced the concept of **self-attention**, which resulted in a change of paradigm with respect to previous sequence modeling methods, significantly outperforming them in most of the scenarios. The idea behind self-attention is that not every word in a sentence equally contributes to the global semantics: a score could be assigned to each word to discriminate between what is important and what can be neglected, *i.e.*, to know what is the most relevant context associated with it. Figure 2.13 schematically explains this idea. In the sentence "the train left the station on time", the context associated with the word "station" can be almost fully

captured only by the words “train” and “left” (regardless of the word “station” itself). This is done by relying on three main ingredients:

- **query**: a specific element that we want to obtain information about, *e.g.*, the word “station” in our example;
- **key**: keys are associated with every word (often called *tokens*) in the sentence and encode their features;
- **value**: as the keys, they encode features associated with the tokens, but they focus on their *meaning*.

The intuition behind this mechanism can be explained as a lookup table, which is a table that relates some (key, value) pairs. Given a question (the query) related to the word that we are investigating, we obtain its representation (key) to access the table, to find its value. As it can be inferred from Figure 2.13, this table is somehow built upon the similarities between the words, that is the sentence with itself (hence the name *self-attention*). First, the (scaled) similarities between the queries and keys are computed:

$$\mathbf{S} = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \tag{2.21}$$

where d_k is the key (as well as the query) dimension, and $\mathbf{Q}, \mathbf{K}$ are matrices representing the query and the key, respectively. Then, the attention weights α are computed:

$$\boldsymbol{\alpha} = \text{softmax}(\mathbf{S}) \tag{2.22}$$

where the softmax operation is applied along the query sequence dimension, such that:

$$\text{softmax}(\mathbf{S}_{i,j}) = \frac{e^{S_{i,j}}}{\sum_j e^{S_{i,j}}}. \tag{2.23}$$

Finally, the output of the self-attention layer is the linear combination of the values, weighted by the attention coefficients:

$$\mathbf{y} = \boldsymbol{\alpha}\mathbf{V} \tag{2.24}$$

where V is a matrix representing the value.

The $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ matrices are obtained from the input through a linear projection:

$$\begin{aligned} \mathbf{Q} &= \mathbf{W}_q \mathbf{x} \\ \mathbf{K} &= \mathbf{W}_k \mathbf{x} \\ \mathbf{V} &= \mathbf{W}_v \mathbf{x} \end{aligned} \tag{2.25}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ are learnable matrices. This allows rewriting the output of the self-attention layer as a function of the inputs:

$$\mathbf{y} = f_W(\mathbf{x}) = \text{softmax}\left(\frac{(\mathbf{W}_q \mathbf{x}) \cdot (\mathbf{W}_k \mathbf{x})^T}{\sqrt{d_k}}\right) \cdot (\mathbf{W}_v \mathbf{x}) \tag{2.26}$$

Thanks to the three learned projections $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$, the self-attention mechanism can learn what are the relationships in the data and, most importantly, learn how to distinguish the most relevant ones from the rest, which translates in assigning a higher weight to the former and penalizing the latter.

A Transformer layer is then followed by further projections and normalizations, which we will not discuss here. The reader can refer to the complete Transformer architecture in (Vaswani et al., 2017).

By construction, the self-attention mechanism is permutation invariant, which means that it focuses on the content of tokens rather than their position in the sequence. However, the sequential structure of the data should be enforced. To address that, *positional encoding* is typically adopted (Vaswani et al., 2017). The idea is to leverage some ancillary information on the position $\mathbf{p} = (p_0, p_1, ..., p_k)$ and to define an encoding $\mathcal{E}$ which maps each position to a vector. Hence, the encoded positions are added to the inputs:

$$\mathbf{x}' = \mathbf{x} + \mathcal{E}(\mathbf{p}). \tag{2.27}$$

2.6.4 Graph neural networks

Convolutional neural networks are powerful models, able to extract information from the data from both local and global scales. Yet, in order for CNNs to be employed,

the input data needs to be **structured**. Since CNNs are based on the idea of sliding a filter onto the data, this means that all data points must be arranged as a uniform grid (*e.g.*, images). However, in some applications, the input data is not structured on a regular grid: for example, in the case of a network of sensors (*e.g.*, meteorological or ground motion measurements). In the scope of this manuscript, a GNSS network is employed. In principle, it is not possible to perform 2D convolutions on GNSS data, since, unlike interferograms, GNSS stations are sparse and not equally spaced. To try to mitigate this issue, (shared) 1-dimensional CNNs can be used (convolution in the time domain) on each station separately. Another solution is to spatially interpolate the data to create gridded maps (or image time series) where the 2-dimensional convolution is possible, as we detail in section 3.2.2.3. However, those techniques are domain-dependent and thus do not adapt to all types of data. Also, they can be computationally expensive and may suffer from data contamination (*e.g.*, in the case of GNSS interpolation on a regular grid, artifacts can be introduced and noise can be enhanced by the interpolation). To solve this issue, graph neural networks (GNNs) extend the convolution operation on data that is not structured as a regular grid, which can be modeled as **graphs**.

2.6.4.1. Graphs

A graph is a mathematical data structure fully described by a pair of sets $(\mathcal{V}, \mathcal{E})$ called *vertices* or *nodes* and *edges*. Given a set of nodes $\mathcal{V} = (v_1, v_2, ..., v_n)$, the edges are defined as connections between two nodes: $\mathcal{E} \subseteq \{(u, v) \mid (u, v) \in [\mathcal{V}]^2\}$, where $[\mathcal{V}]^2$ represents the family of unordered-2-element subsets of $\mathcal{V}$. For example, if $\mathcal{V} = \{1, 2, 3\}, [\mathcal{V}]^2 = \{(1, 2), (1, 3), (2, 3)\}$. Two nodes connected by an edge are called *adjacent* or *neighbours*. Such a graph is called *undirected* since the relationships between nodes don't have a direction, that is there is no "source" and "destination": the edge (u, v) is equal to the edge (v, u). For the rest of the chapter, we will talk about undirected graphs only.

Figure 2.14 shows an example of an undirected graph, containing 5 nodes and 6 edges, with $\mathcal{V} = \{A, B, C, D, E\}$, $\mathcal{E} = \{(A, B), (A, C), (B, B), (B, C), (C, D), (D, E)\}$. The edge (B, B) is called a *self-loop*.

A graph can be fully described by its *adjacency matrix*. The adjacency matrix is a square matrix $A \in \mathbb{R}^{N \times N}$, where N is the number of nodes. A is symmetric in the case of undirected graphs. Each element of A is such that $A_{i,j}$ is 0 or 1 depending on if the edge (i, j) exists. If a weight is associated with each edge (weighted graph), then $A_{i,j} = e_{i,j}$. Edge weights $e_{i,j}$ can be thought of as a measure of the strength of

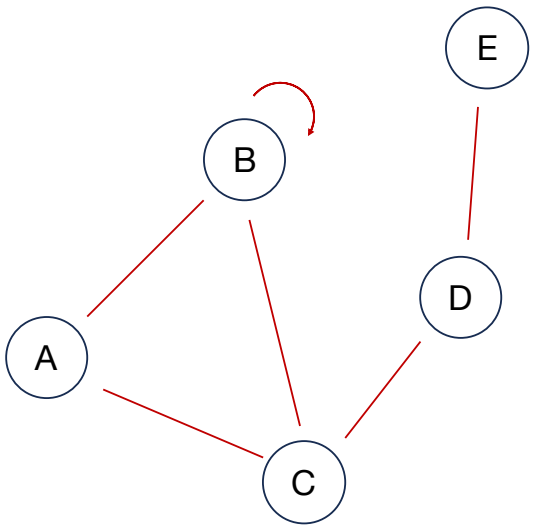


Fig. 2.14.: Example an undirected graph.

the connection between two nodes, depending on the specific application. In the example of Figure 2.14, the (unweighted) adjacency matrix is:

$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{2.28}$$

2.6.4.2. Graph neural networks

Graph neural networks (GNNs) extend the convolution operation on graphs, which can be thought of as a generalization of structured data, *e.g.*, images. GNNs generalize the convolution operation via *message passing*. The idea of message passing is that the features of the nodes are updated by taking into account the features of neighbouring nodes, as:

$$y_u = \text{update}(x_u, \underset{v \in \mathcal{N}(u)}{\text{aggregate}}(\text{message}(x_u, x_v, e_{u,v}))) \tag{2.29}$$

where $\mathcal{N}(u)$ indicates the neighbourhood of the node u, *i.e.*, its adjacent nodes. The message$(\cdot)$ function indicates how the features associated with the nodes u and v are combined (by taking into account a possible edge weight $e_{u,v}$). The aggregate$(\cdot)$

function is permutation-invariant and aggregates the aforementioned messages to a new aggregated representation that captures the collective information from the neighborhood. The update$(\cdot)$ function is responsible for updating the features of each node based on the aggregated messages received from neighboring nodes.

Many GNN layers have been developed after message passing with specific characteristics. Here, we focus on the **Graph Convolutional Network** (GCN). The key idea behind graph convolution is to use the graph structure to define a local receptive field for each node, similar to how convolutional layers operate on regular grids (*e.g.*, image-like data). Graph convolution involves aggregating features from neighboring nodes, applying a learnable transformation, and updating the node's representation, as:

$$y = f_W(X) = \mathbf{AXW} \tag{2.30}$$

where $\mathbf{A}$ is the adjacency matrix, $\mathbf{X}$ the input features and $\mathbf{W}$ a learnable weight matrix. The matrix multiplication $\mathbf{AX}$ performs neighborhood selection: for a given node in the column of $\mathbf{X}$, the multiplication with the respective row of the adjacency matrix, containing its neighbours, computes the sum of the node features and all its neighbours. By further multiplying by $\mathbf{W}$, the model can learn the best connections as well as (typically) expand the data dimensionality, allowing for more complex data representations.

The adjacency matrix A has the property that the matrix A^k, obtained by raising A to the power k, contains 0 everywhere except when two nodes u and v are reachable in k hops, *i.e.* if a path of length k exists between them. This means that, by increasing the number of GCN layers, it is possible to increase the receptive field of a node.

In practice, the adjacency matrix (see equation 2.30) is normalized for better numerical stability, but this will not be discussed here.

2.6.5 Spatial-temporal graph neural networks

Graph neural networks are usually used with univariate features, for example, a list of characteristics for each node. However, many applications are multi-variate. In the case of this Ph.D. thesis, we work with three-dimension input data, that can be thought of as a tensor of shape (station, time, component). Standard graph neural networks can still be used in this context, *e.g.*, by applying a GNN to stations and components, yet without taking the time dimension into account. Spatial-temporal

graph convolutional networks (STGCN) have been developed to extend classical GNNs to spatiotemporal data. The key idea is that spatial (GNN-like) and temporal modules (CNNs, RNNs, etc.) are intertwined to perform joint spatial and temporal analysis. We give further details in section 5.2.1.

2.6.6 Deep learning in solid earth sciences

The number of deep learning applications in seismology is rapidly growing (Mousavi & Beroza, 2022), probably thanks to the flexibility of deep learning in analyzing large-scale data sets as well as extracting meaningful features from raw data, as seen in the previous sections. Here we review some of the recent advances in deep learning applied to solid earth sciences. For a more complete overview, the reader can refer to (Bergen et al., 2019; Kong et al., 2019a; Mousavi & Beroza, 2022, 2023).

Most of the works in seismology make use of convolutional neural networks to analyze continuous seismic waveforms to address several tasks such as earthquake detection and phase selection. Zhu and Beroza, 2019 use a deep learning approach based on CNNs to pick the arrival times of both P and S waves in seismic data. Mousavi et al., 2020a also focus on P- and S-wave picking by combining convolutional, recurrent, and transformer neural networks. For further examination, please refer to (Münchmeyer et al., 2022). Ross, Yue, et al., 2019 use recurrent neural networks to build a deep learning method for phase association. Another direction is earthquake source parameter estimation, *e.g.*, Münchmeyer et al., 2020; Saad et al., 2020. Münchmeyer et al., 2021 use a CNN- and transformer-based neural network for the location and magnitude estimation from real-time seismic waveforms. van den Ende and Ampuero, 2020 also focus on earthquake location and magnitude estimation with a simple graph neural network using a $\max(\cdot)$ operation to aggregate features at each station. X. Zhang et al., 2022 use a graph neural network with a convolutional neural network to extract temporal features prior to the GNN module.

Some notable work is carried out in geodesy as well. Anantrasirichai et al., 2018 use a CNN-based network to detect volcano deformation using synthetic data and Rouet-Leduc et al., 2021a use a CNN-based autoencoder to denoise SAR interferograms to recover deformation associated with slow slip events. Rouet-Leduc et al., 2020 use a convolutional neural network trained on spectrograms to detect tremor activity as a proxy for slow slip. In the frame of denoising, Zhu et al., 2019 used a convolutional autoencoder to denoise and decompose seismic signals based on spectrograms.

Thomas et al., 2023 developed a method based on (Zhu et al., 2019) to denoise high-rate-GNSS time series.

3

Seismic source characterization from GNSS data using deep learning

This chapter is based on the article published in the *Journal of Geophysical Research (JGR): Solid Earth* as:

Costantino, G. and Giffard-Roisin, S. and Marsan, D. and Marill, L. and Radiguet, M. and Dalla Mura, M. and Janex, G. and Socquet, A. (2023). *Seismic Source Characterization From GNSS Data Using Deep Learning*, Journal of Geophysical Research: Solid Earth, 128(4), e2022JB024930, https://doi.org/10.1029/2022JB024930.

In this chapter, we present a deep learning approach to identify and characterize earthquakes on- and off-shore Honshu (Japan) with the GEONET GNSS network. We propose three data representations and corresponding deep-learning models to tackle this problem. In addition to time series, we rearrange GNSS data into images and image time series, to capture the spatial and spatiotemporal variability, respectively. We show that our preferred model, ITS, based on a transformer neural network, can detect earthquakes in GNSS data down to magnitude 6 in real data.

Abstract The detection of deformation in 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 spatiotemporal 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 spatiotemporal domains, respectively. Thereafter, we design and develop a specific deep-learning model for each data set. 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 spatiotemporal 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 detecting and characterizing fast or slow earthquakes.

Plain language summary The continuous monitoring of ground displacement with Global Navigation Satellite System (GNSS) 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 spatiotemporal variability, respectively. We train our methods on synthetic data since real data sets 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 towards the study of the slow deformation.

3.1 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 position time series. Seismic source characterization consists of the inversion of the location, magnitude, 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, catalogues having low magnitudes of completeness can be obtained. Thus, catalogues 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 the 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 modelling methods with a fine-tuning of the parameters (Blewitt et al., 2009; Feng et al., 2015; Guo et al., 2015; J.-T. 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 catalogues 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. Catalogues 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 catalogues of unprecedented density (Kong et al., 2019a; Mousavi et al., 2020a; Ross, Yue, et al., 2019; Seydoux et al., 2020; Zhu & Beroza, 2019; Zhu et al., 2019), earthquake early warning (Münchmeyer et al., 2021; Saad et al., 2020; X. Zhang et al., 2021), prediction of ground deformation (Kong et al., 2019a; Mousavi et al., 2020a), 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 B. He et al., 2020; Hulbert et al., 2019; Hulbert et al., 2020; Rouet-Leduc et al., 2019; Rouet-Leduc et al., 2020, with notable applications to InSAR data by Anantrasirichai et al., 2019; Rouet-Leduc et al., 2021a. 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 analyse 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, *i.e.*, 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 labelled. The data ground truth comes from seismic catalogues, 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 catalogues. The strengths and the pitfalls of the presented methods are discussed by envisioning some possible strategies to improve the results.

3.2 Methods

3.2.1 Background work and positioning

3.2.1.1. 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 non-trivial 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.

3.2.1.2. Followed approach

An overview of the proposed methodology is shown in Figure 3.1. 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 $\langle$ input, desired output $\rangle$ 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

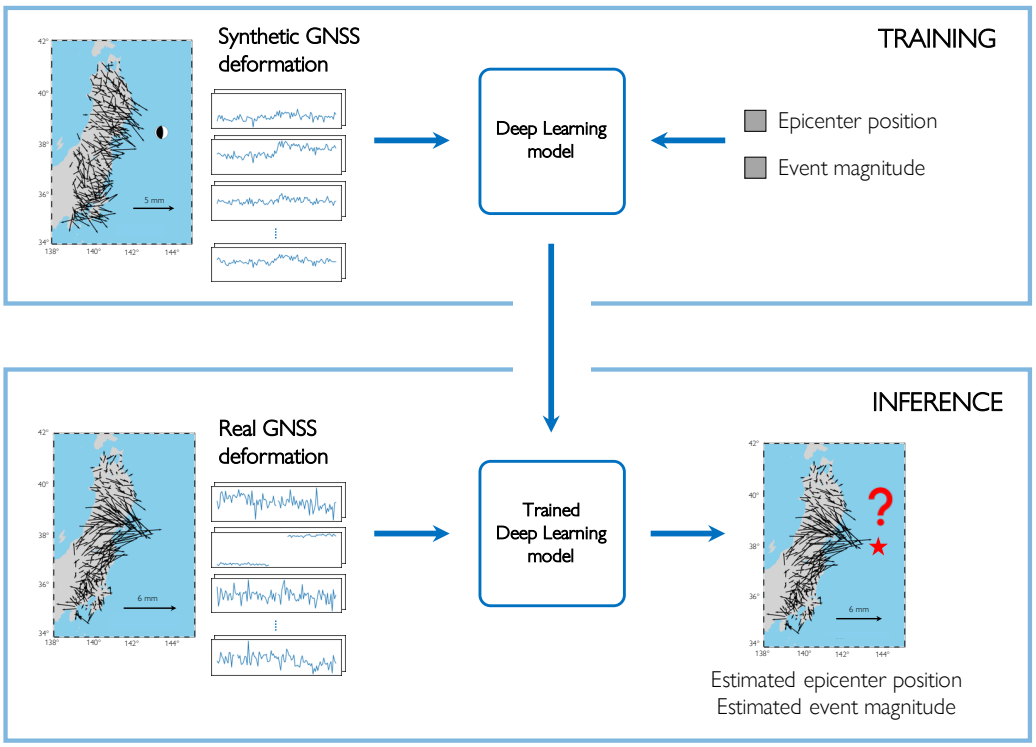


Fig. 3.1.: Schema of the proposed workflow, summarizing the training and the inference phases. A given deep learning model is trained by providing an input and a desired output. Here we use GNSS data as input and a triple consisting of (fault centroid latitude, fault centroid longitude, magnitude) as 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.

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, *i.e.*, 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 afterwards. 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.

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 3.2.2.

3.2.2 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 data set 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:

$$\mathbf{x} = \mathbf{s} + \boldsymbol{\varepsilon} \tag{3.1}$$

with $\mathbf{s}$ the synthetic signal (cf. section 3.2.2.1) and $\boldsymbol{\varepsilon}$ the noise term (cf. section 3.2.2.2).

3.2.2.1. Synthetic displacement

We obtain the synthetic displacement signals $\mathbf{s}$ 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):

$$M_0 = 10^{1.5m+9.1}\ \mathrm{N}\cdot\mathrm{m} \tag{3.2}$$

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: $\phi_S \sim \mathcal{U}(160°, 240°)$, $\delta \sim \mathcal{U}(20°, 30°)$, $\lambda \sim \mathcal{U}(75°, 100°)$. Static stress drop $\Delta\sigma$ is assumed to be a lognormal random variable with an average value of 3MPa and a standard deviation of ± 30 MPa. A circular crack is assumed with radius R computed as (Aki & Richards, 2002):

$$R = \left(\frac{7}{16}\frac{M_0}{\Delta\sigma}\right)^{1/3} \tag{3.3}$$

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

$$\pi R^2 = L \cdot W \tag{3.4}$$

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):

$$\bar{u} = \frac{16}{7\pi}\frac{\Delta\sigma}{\mu}R \tag{3.5}$$

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: (1) allowing for a complete variety of focal mechanisms would dramatically increase the size of the training database, (2) 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 3.10).

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 3.2.1.2 and Figure 3.1). 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.

3.2.2.2. 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.

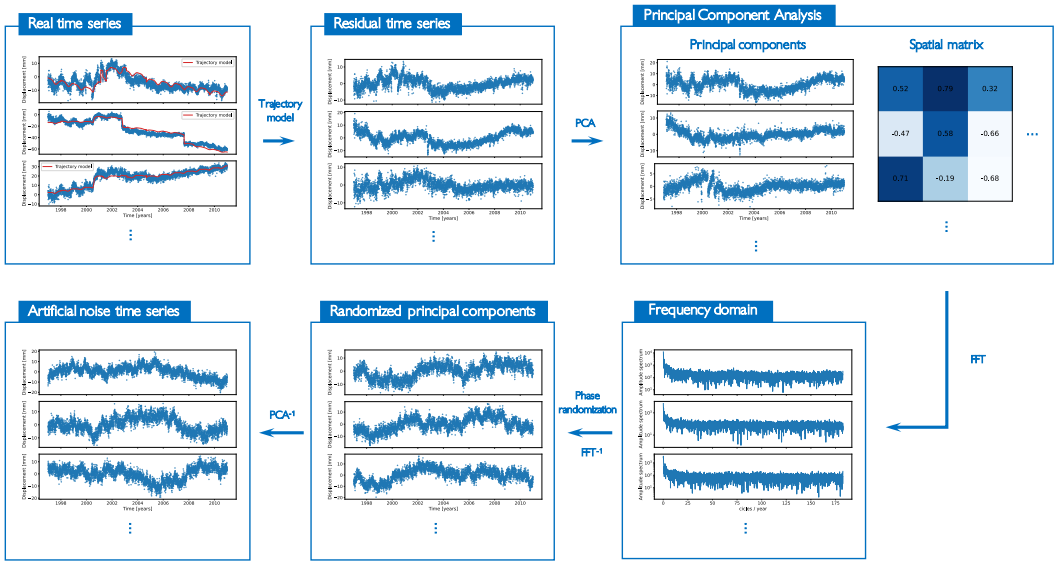


Fig. 3.2.: Schema of the artificial noise time series generation. Only 3 (notably stations 3032, 3033 and 3034) of the 300 GEONET GNSS stations are shown for better visualization (N-S component). (1) Raw GNSS time series are scanned through the quadratic trajectory model from (Marill et al., 2021). (2) 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 3.2.2.2) 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*.

Realistic perturbations, *i.e.*, 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–days windows, by taking into consideration all the stations at the same time. Afterwards, 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 ϵ will have, on average, the same spatial covariance. Moreover, we can build new noise samples by randomizing the phase, since the Power Spectral Density (PSD) of the transformed samples and the actual ones will be asymptotically equivalent. A schema of the artificial noise generation is given in Figure 3.2.

3.2.2.3. 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 3.3. 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 (1) it is noisier compared to the horizontal components and (2) 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 3.2.2.2). We add a Heaviside step to simulate the co-seismic displacement (Bevis & Brown, 2014), with the onset time (cf. t_c in Figure 3.3) being

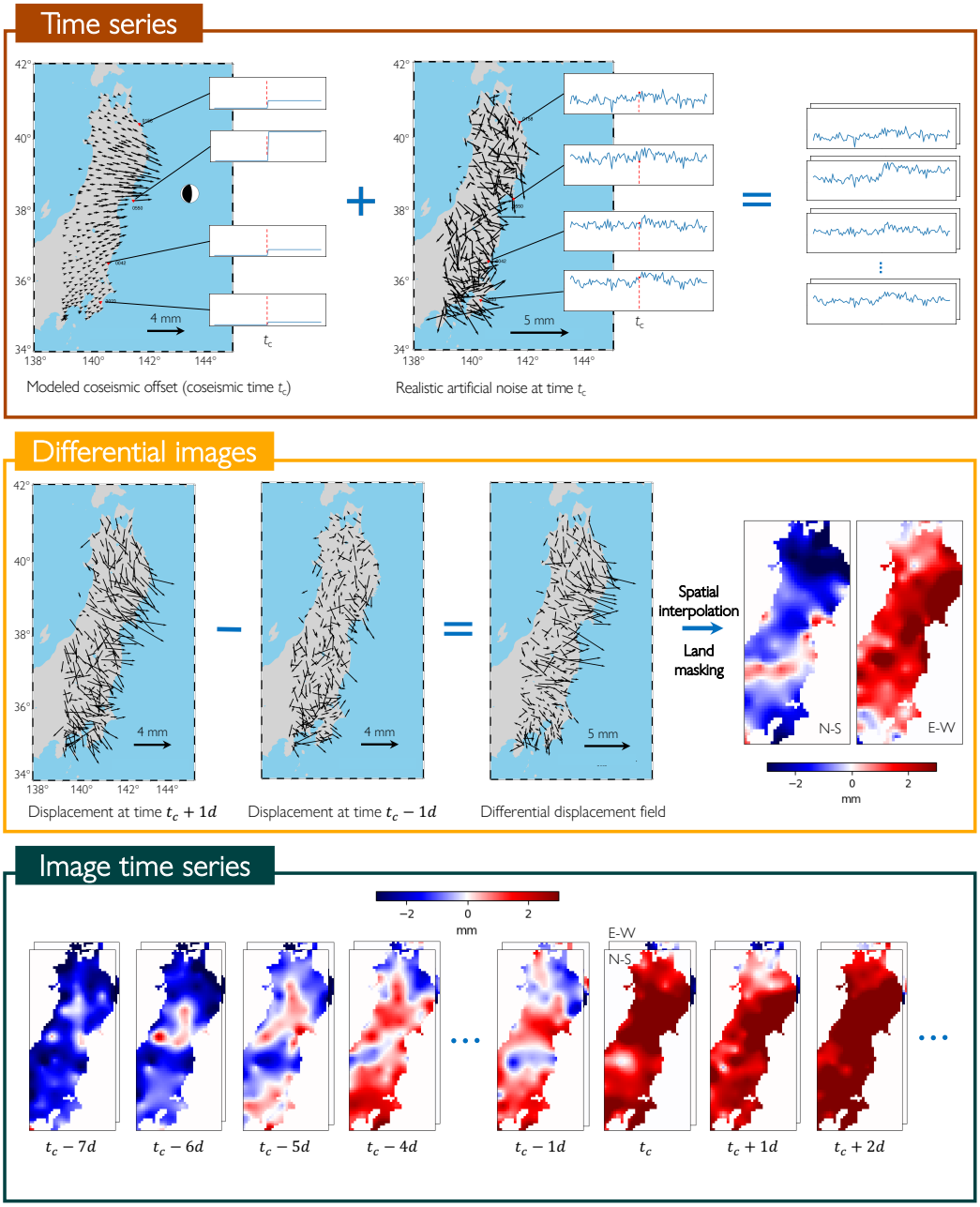


Fig. 3.3.: Outline of the three employed data representations. Each arrangement is designed for a specific deep learning model (cf. Figure 3.4 with corresponding colors). The data–arrangement procedure is shared between synthetic and real data, except for time series, which are directly available from GNSS recordings. **(time series)** is associated with the TS model. Synthetic position time series are built by adding a modeled signal (cf. section 3.2.2.1) to a realistic noise time series (cf. section 3.2.2.2) by imposing the time of the co-seismic offset to be at the center of the window (cf. section 3.2.2.3). **(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, 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.

at the center of the window. The step amplitude for each station depends on the modeled displacement (cf. section 3.2.2.1). 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 minutes (≈ 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. Afterwards, 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 $\mathbf{D}$ 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 k-th 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 minutes (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 7 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 repre-

sented by tensor $\mathbf{T} \in \mathbb{R}^{M \times I \times J \times 2}$, with M the length of the image time series and $\mathbf{T}(t_i, \mathbf{S}(k)) = \mathbf{X}(k, t_c + i)$, $i \in (-\lfloor \frac{M}{2} \rfloor, ..., 0, ..., \lfloor \frac{M}{2} \rfloor)$.

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

3.2.3 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 for geodetic data, in order to best address our specific problem. A graphical outline of the methods is provided in Figure 3.4.

3.2.3.1. 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; Kong et al., 2019a. 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 3.4. 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. Afterwards, 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 (ReLu) (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

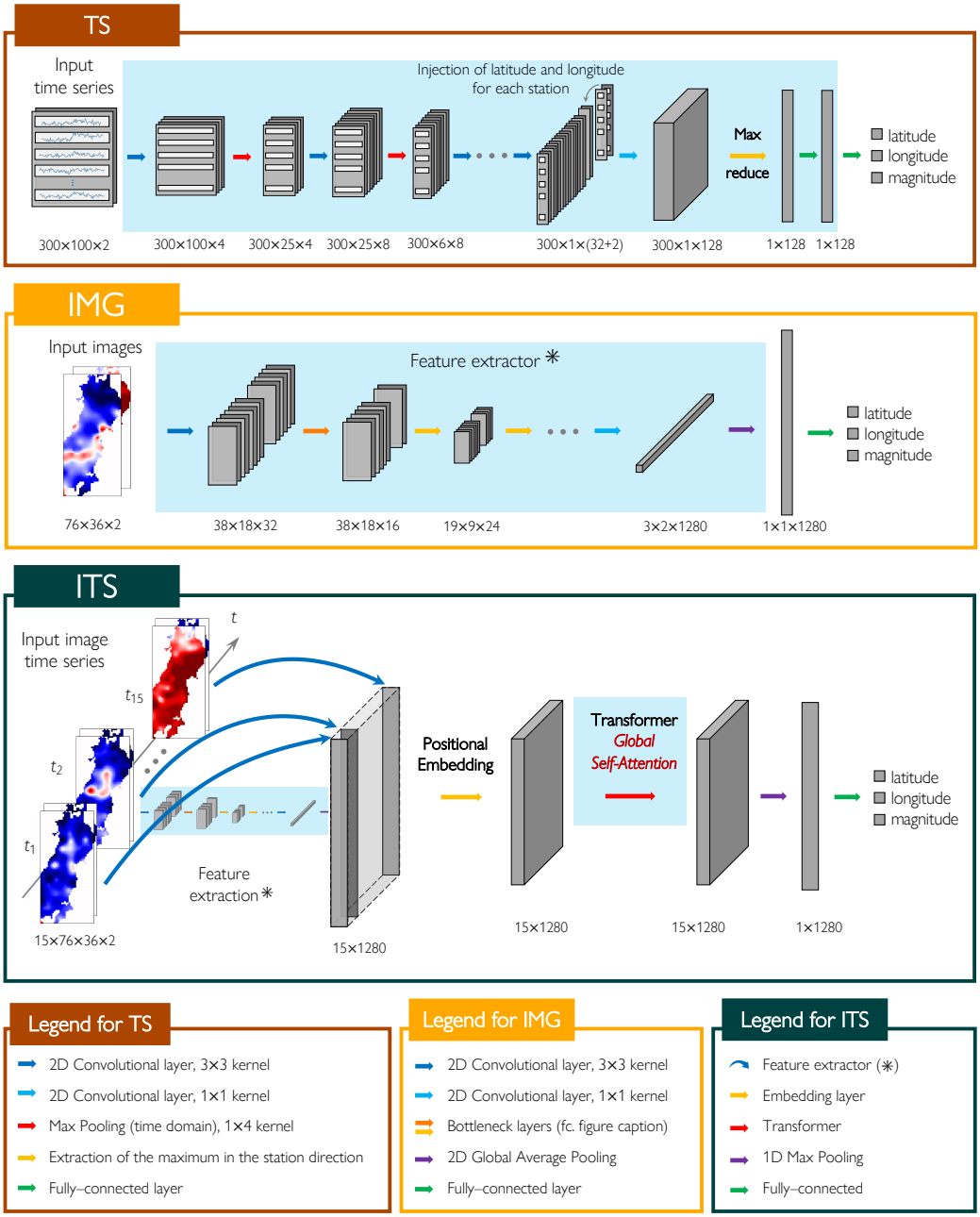


Fig. 3.4.: The three reference deep learning methods designed in this work. Shaded cyan rectangles represent existing state-of-the-art models. Such models have been slightly modified or adapted, where specified (cf. section 3.2.3). Further details, such as dropout layers, stride and activation functions, 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 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 3.2.3.2) 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., 2020a)

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.

3.2.3.2. 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 3.4. 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.

3.2.3.3. 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 behaviour of the data (Bergen et al., 2019; Q. Wang et al., 2017), as well as Transformers, which have overcome the former becoming the reference methods in the state–of–art (Mousavi et al., 2020a; 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.

The ITS architecture is presented in the third box of Figure 3.4. 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, *i.e.*, 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. Afterwards, we stack all the feature vectors in one matrix to be exploited by the Transformer layer, as shown in the third box of fig 3.4. 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 learnt during the training phase. After the embedding layer, we use a Transformer equipped with additive self–attention, as in (Mousavi et al., 2020a). 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 3.2.3.1) 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 learnt 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.

3.2.4 Implementation and training details

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

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\frac{1}{d}\sum_{i=1}^{N}\sum_{j=1}^{d}(y_{i,j} - \hat{y}_{i,j})^2 \tag{3.6}$$

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.

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).

3.3 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 data sets under consideration are the same for all the models and differ only in their input representation.

Tab. 3.1.: 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	$\mathbf{52.65 \pm 32.34}$	$\mathbf{0.08 \pm 0.05}$

Table 3.1 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:

$$E_p^i = \sqrt{\left(X_i - \hat{X}_i\right)^2 + \left(Y_i - \hat{Y}_i\right)^2} \tag{3.7}$$

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 (MAE) for the magnitude, which is computed for each sample as:

$$E_m^i = |m_i - \hat{m}_i| \tag{3.8}$$

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).

3.3.1 Analysis of the performance

Figure 3.5 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

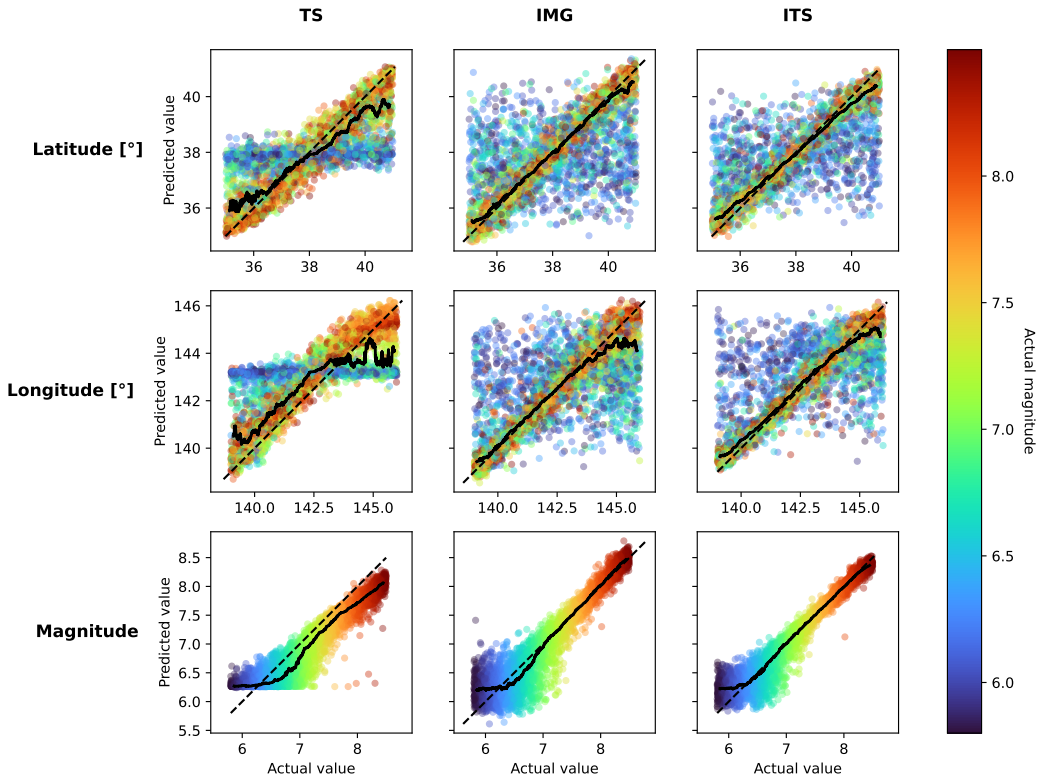


Fig. 3.5.: Comparison of the performance of the tested models at inference time. TS, IMG and ITS models are shown on columns respectively. For each row, latitude, longitude and magnitude predictions are reported, respectively. Each point of the scatter plots represents a test sample, whose magnitude 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.

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 behaviour 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 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, *i.e.*, 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 3.5). We can derive a resolution limit for the ITS model: $M_w^{ITS} = 6.2$.

From the latitude and longitude prediction, *i.e.*, 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 behaviour 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 (S. S. 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 towards 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.

3.3.2 Spatial variability of the location error

Figure 3.6 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. 3.3). 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 behaviour, since, as the magnitude increases, the highest errors are pushed towards 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

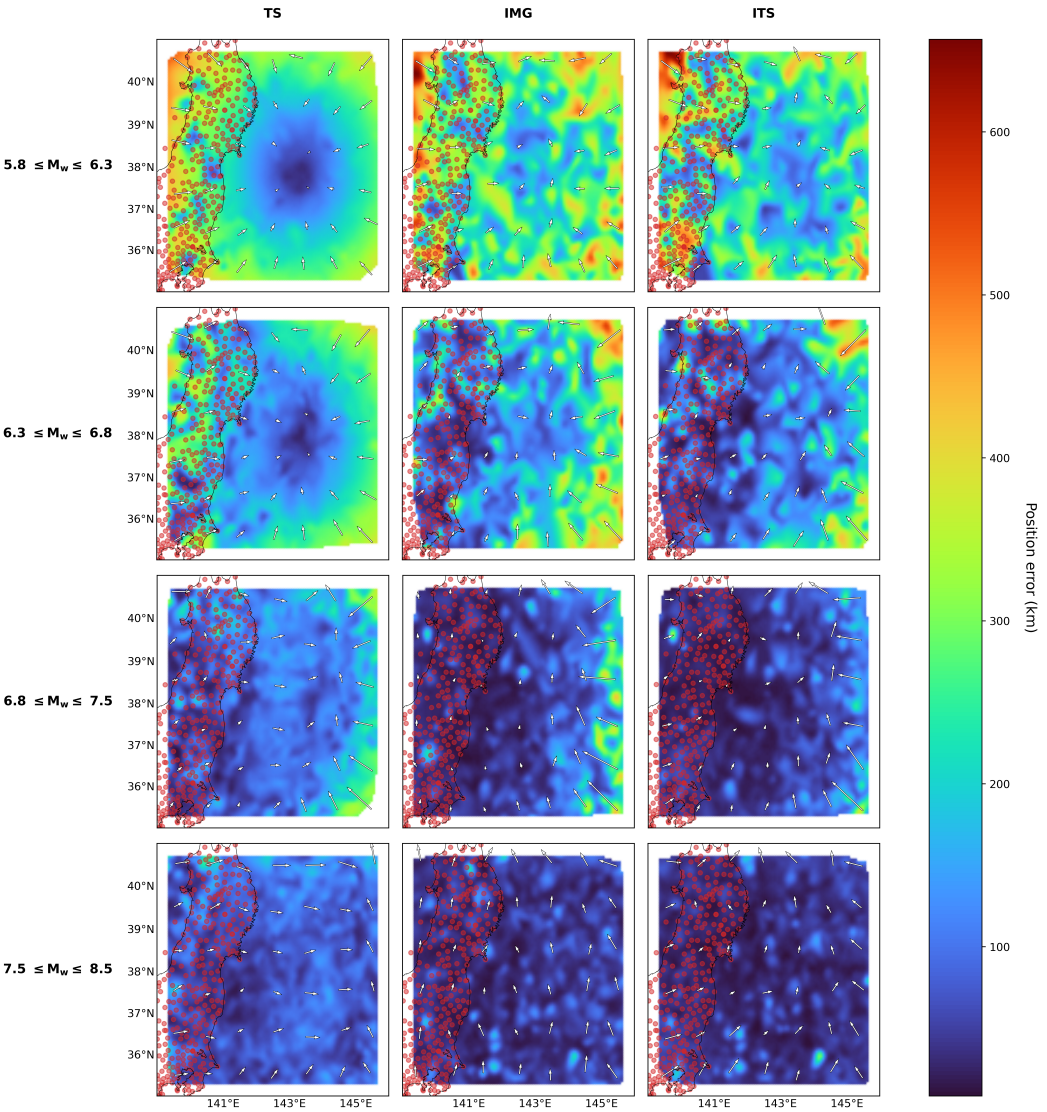


Fig. 3.6.: Comparison of the location error of the tested models, reported in the columns. Each subplot shows the location error associated with the test samples, interpolated on a grid whose corresponding spatial coordinates are indicated along the axes. Magenta data points represent the position of 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.

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 3.6 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 3.5 and 3.9). 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.

3.3.3 Influence of the distance from the GNSS network on the predictions

Figure 3.7 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 3.7). The solid lines correspond to the median for several magnitude ranges (cf. Figure 3.6).

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, 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 behaviour at small magnitudes or because the error at

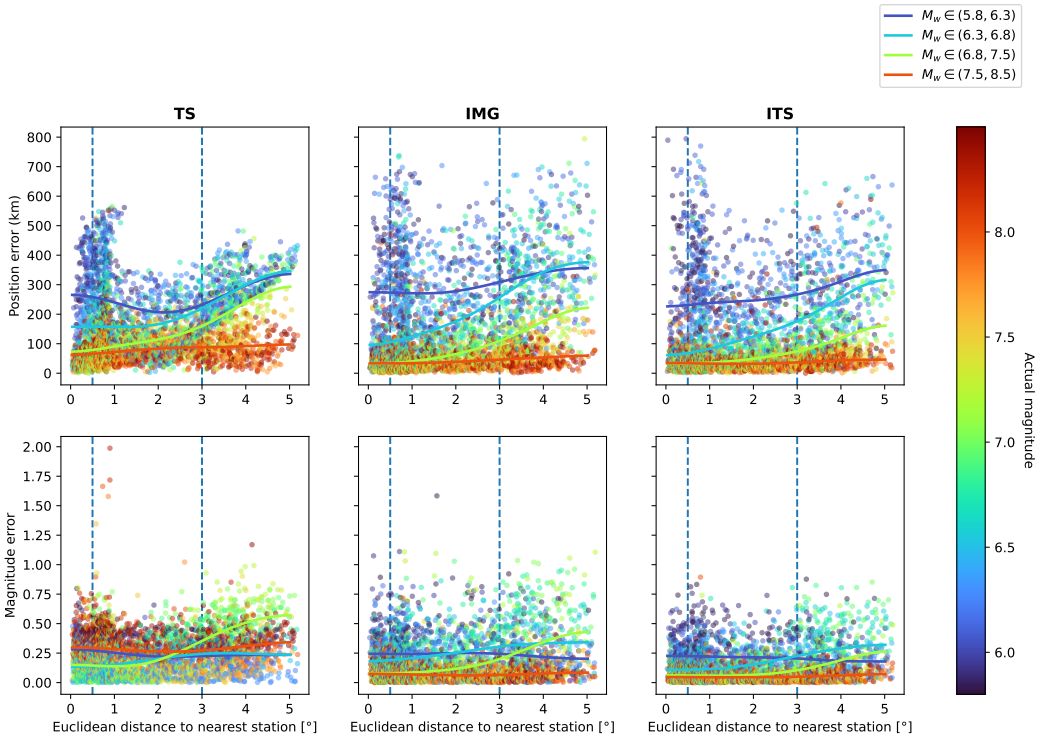


Fig. 3.7.: Comparison of errors as a function of the distance to the nearest GNSS station. The deep learning models are shown in columns, while the rows indicate position and magnitude errors, respectively. Each scatter plot depicts errors as a function of the Euclidean distance to the nearest GNSS station, 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.

Tab. 3.2.: Magnitude thresholds of ITS estimated against the synthetic test set.

	depth $\leq$ 30 km	30 km $<$ depth $\leq$ 60 km	60 km $<$ depth $\leq$ 100 km
near field	6	6.2	6.5
interm. field	6.8	6.8	7
far field	7.5	7.5	7.8

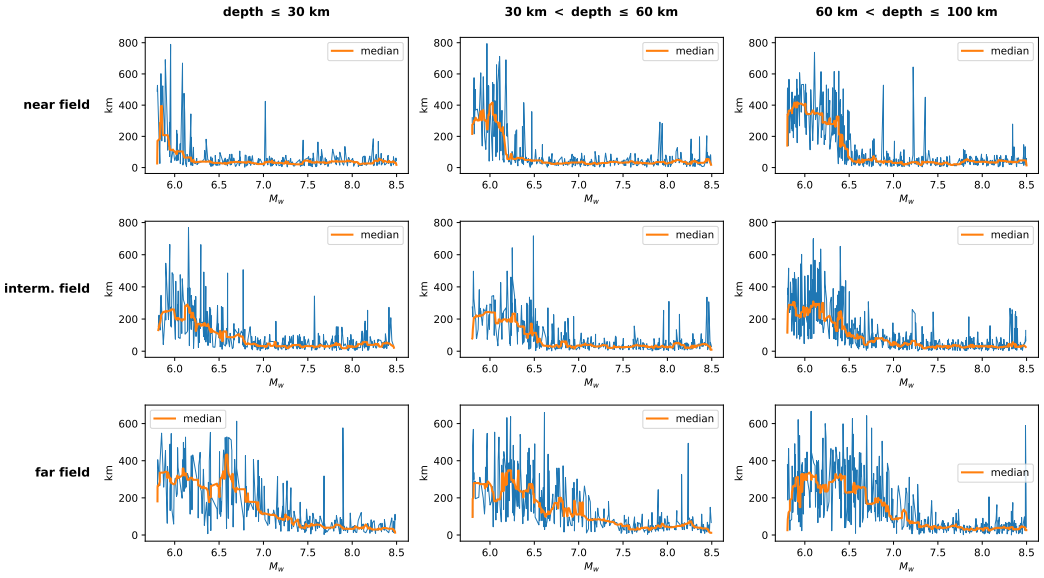


Fig. 3.8.: Position error, computed for each test sample, as a function of the magnitude (x-axis), the depth range (columns), and the distance range (cf. Figure 3.7) with respect to the GNSS network (rows) for ITS. The orange solid line represents the result of a median smoothing by employing a kernel size of 15 points.

high magnitudes does not depend on the distance to the 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 3.5) 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.

3.3.4 Magnitude threshold estimation from ITS localization error

Since we will test the deep learning models on real data (cf. section 3.4), we define here a criterion to assess whether a characterization coming from a learning model is reliable. Figure 3.8 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, *i.e.*, 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 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 3.2 and will be used in section 3.4 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.

3.3.5 Interplay between depth and magnitude in the magnitude resolution

Figure 3.9 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 3.6). 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 3.5). 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 network, the trend does not

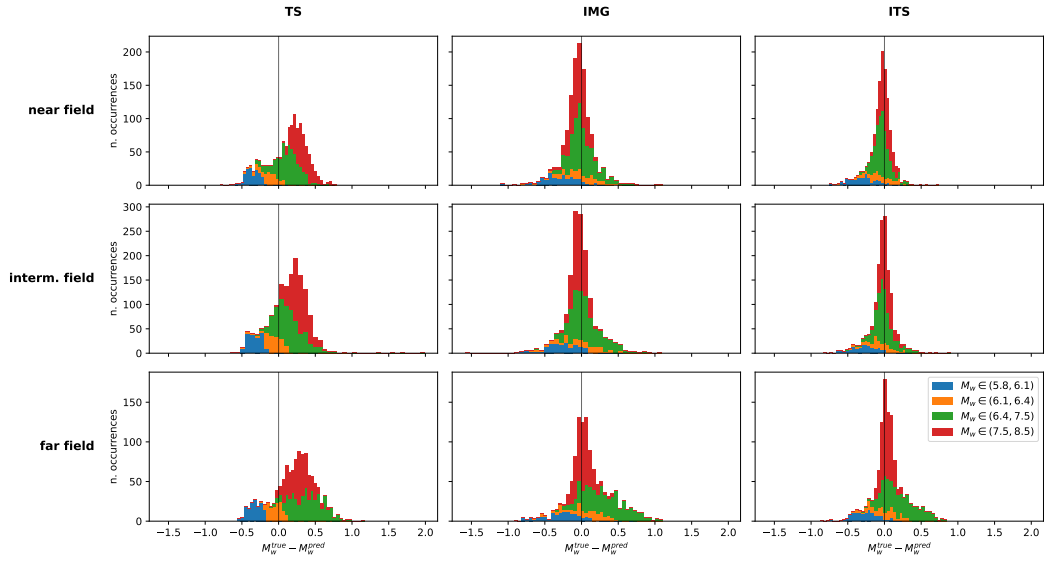


Fig. 3.9.: Cumulative histograms of the magnitude difference $M_w^{true} - M_w^{pred}$ for the three models (columns) as a function of the distance range (cf. Figure 3.7) with respect to the GNSS network (rows) and for different magnitude ranges (color). Vertical lines individuate the zero.

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 3.5). 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. Figure 3.5 and 3.6). 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.

3.3.6 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:

$$p_s(t) = H \log_{10}\left(1 + \frac{t - t_c}{\tau}\right) \tag{3.9}$$

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

$$H \sim \mathcal{U}(0.5c, 1.5c) \tag{3.10}$$

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:

$$\tau \sim \mathcal{U}(1, 50) \text{ days.} \tag{3.11}$$

The results are shown in Figures S1 – S5. 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 3.5, 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.

3.4 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 catalogued earthquake, we run our processing pipeline (details in section 3.4.1) to obtain the estimated fault centroid location and magnitude, which we compare with the

catalogued ones, in order to analyze the performance of the tested deep learning models in presence of raw noisy GNSS data.

3.4.1 Data processing

The seismic catalogue selection for real events in Japan has been conducted as follows. The F-Net catalogue 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 3.2.2.1) 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 shown that its actual value is rather 9.1. For this reason, we replaced the catalogued magnitude with the correct value. After this pre-processing, 84 events are present in the catalogue.

Two GNSS data sets 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 catalogue, a window of 100 days, centered onto the co-seismic offset (cf. section 3.2.2). We considered a 100–day 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–day 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-

Tab. 3.3.: 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 3.10). These results should be taken carefully, since they have been obtained on a very limited number of samples, notably 5 for ISTerre/DD and 3 for NGL/PPP.

	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	$\mathbf{0.11 \pm 0.08}$
IMG	$\mathbf{52.47 \pm 21.82}$	$\mathbf{0.13 \pm 0.09}$	140.98 ± 61.49	0.63 ± 0.17
ITS	76.73 ± 21.62	0.25 ± 0.14	$\mathbf{101.92 \pm 10.04}$	0.45 ± 0.16

seismic offset when that information may be missing. Finally, the data is detrended, *i.e.*, the linear trend is subtracted for every 100–day stack.

After the previous processing, the ISTerre/DD and the NGL/PPP data sets contain 69 and 51 labelled time series. We used the magnitude thresholds obtained for ITS (cf. Table 3.2) to differentiate the theoretically-characterizable events from the rest, as shown in Figure 3.10, that is if magnitude, depth and position of the events are such that they satisfy the experimentally–derived relationships detailed in Table 3.2. We found 8 (3 of which are thrust subduction events and 3 are thrust crustal events) and 5 (2 of which are thrust subduction events and one is a thrust crustal event) characterizable events for ISTerre/DD and NGL/PPP data sets, 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 catalogued 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) catalogue (https://www.globalcmt.org/) (Dziewonski et al., 1981; Ekström et al., 2012).

3.4.2 Results and discussion

The quantitative results are shown in Table 3.3, while Figure 3.11 shows the performance of the tested methods on the two real data sets. The displacement fields associated with all the characterizable events in the ISTerre/DD dataset are represented in Figure 3.12.

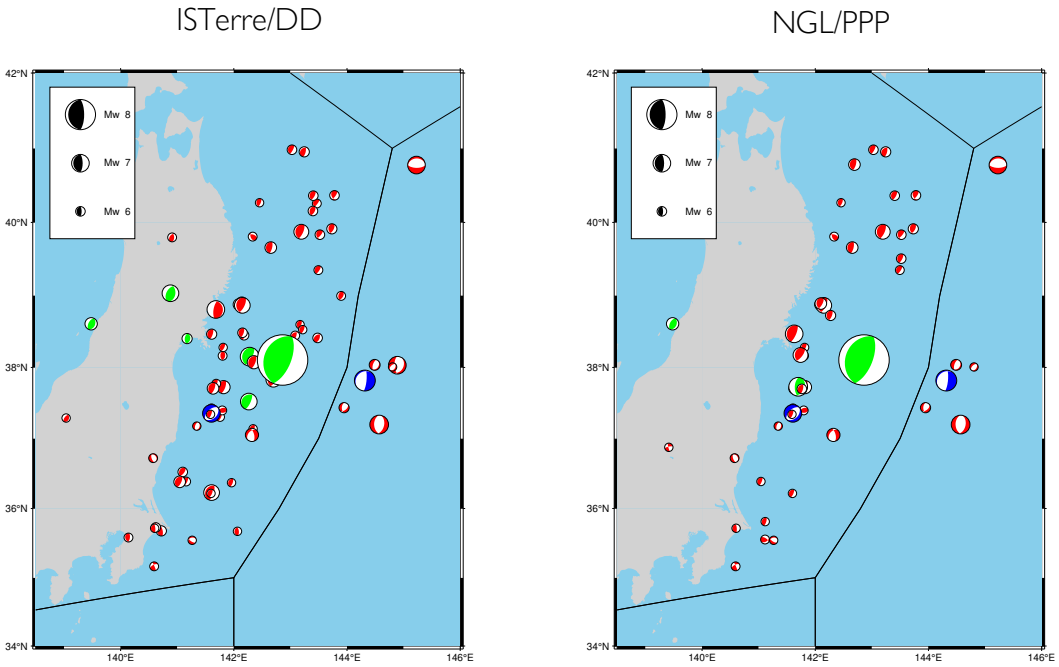


Fig. 3.10.: Seismic catalogue associated with the ISTerre/DD and NGL/PPP data sets, respectively. ISTerre/DD data set contains 69 events ranging from 1998 to 2019, while NGL/PPP set contains 51 events ranging from 2009 to 2021. Focal mechanisms are shown for each event and their size is proportional to the magnitude, according to the legend at the top left. Based on the "characterizability" thresholds from Table 3.2, 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.

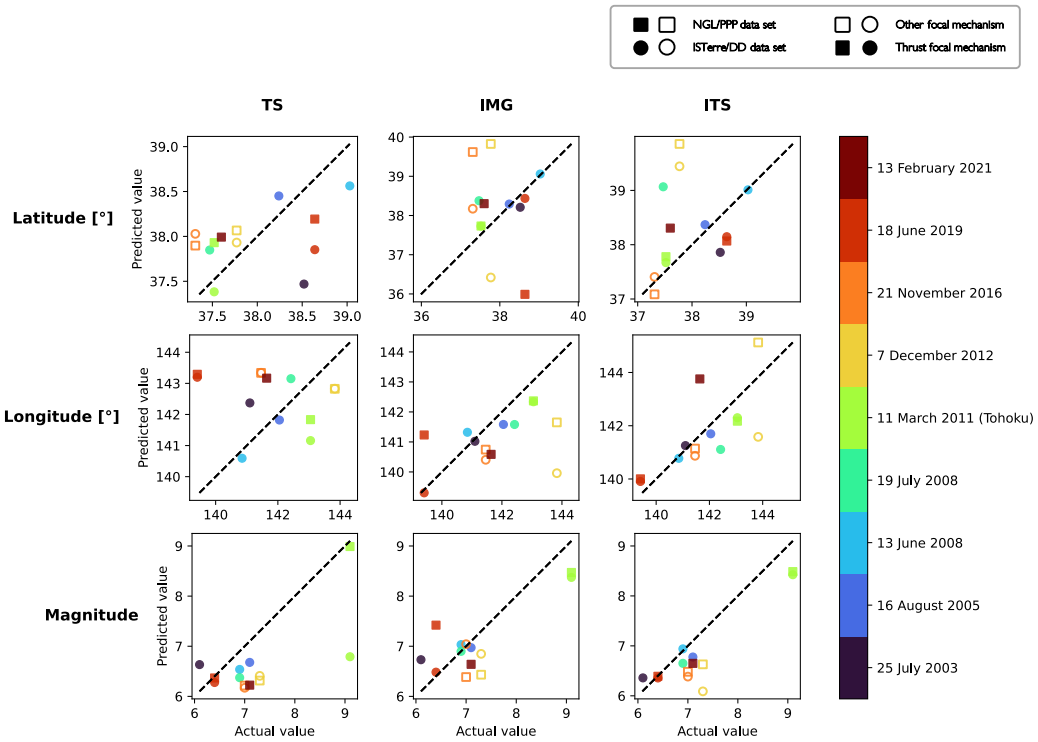


Fig. 3.11.: Actual *vs* predicted plot on real data from ISTerre/DD and NGL/PPP data sets. Each subplot shows the *real vs predicted* comparison for the estimated parameters (fault centroid latitude, fault centroid longitude and magnitude in each row) for each of the three methods (TS, IMG and ITS in each column). For each scatter plot, circles and squares represent predictions associated with the ISTerre/DD and NGL/PPP data sets, 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.

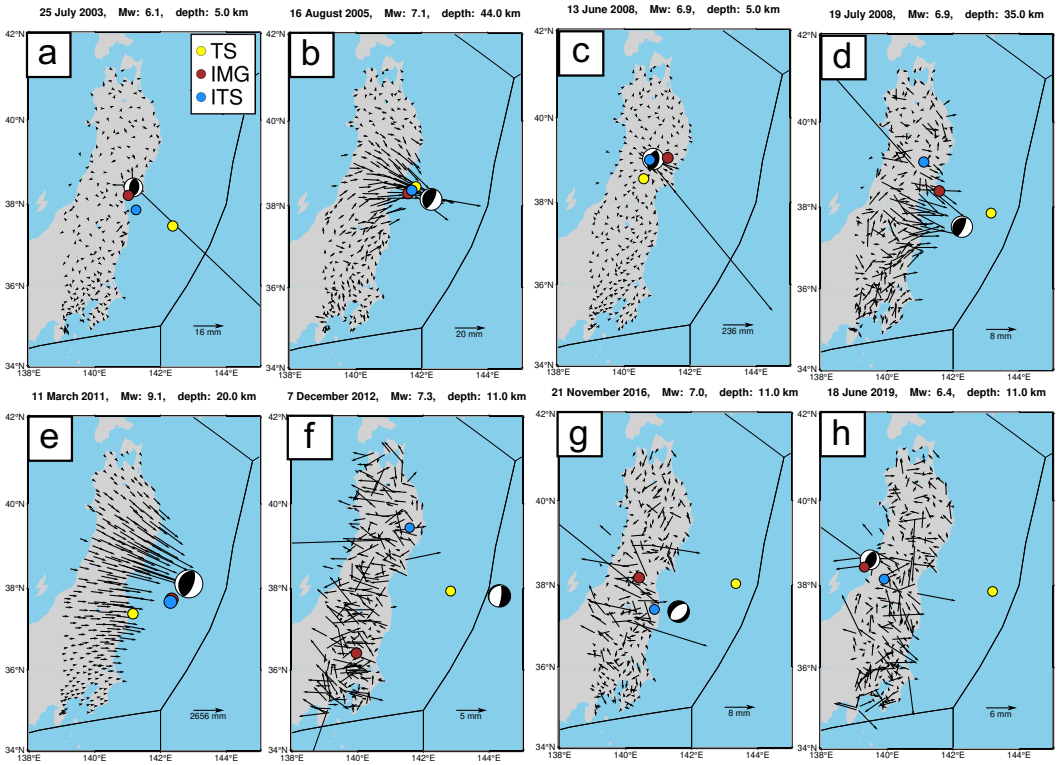


Fig. 3.12.: Displacement fields associated with the eight events of the ISTerre/DD data set. The deformation fields have been computed by subtracting the deformation at day $t_c + 1$ and $t_c - 1$. In each subplot, the focal mechanism from the NIED catalogue is shown as well as the magnitude and depth (in each title) with the yellow, brown and blue points representing the predictions for TS, IMG and ITS, respectively.

From the numerical results listed in Table 3.3, 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 data set in terms of median prediction error, with time–series–based models (*i.e.*, TS and ITS) being more accurate on the NGL/PPP data set on magnitude and location accuracy, respectively. All the models have a larger error associated with the NGL/PPP dataset, probably 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 3.11 shows that, on average, the predictions of the image–based models, *i.e.*, 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 data sets, in line with the results obtained on synthetic data (cf. section 3.3). 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 data set, 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 data set obtained from a DD approach, we will focus on the ISTerre/DD data set henceforward, which also has more data samples to analyze.

The events in Figure 3.11 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 3.11 seem to suggest 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 data set 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 Figure 3.12 (d), (g) and (h). The ITS model is unable to separate the source of displacement in the 19 July 2008 event (Figure 3.12 (d)) because of an outlier in the displacement field, whose influence is better mitigated by the differential approach used for IMG (cf. Figure S6 – S8). On the contrary, ITS can more effectively retrieve the 21 November 2016 event (3.12 (g)), likely thanks to the spatiotemporal approach (cf. Figure S9 – S14), 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 3.12 (d)) 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 data set and as a $M_w \sim 6.8$ event on the ISTerre/DD data set. 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 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), 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).

3.5 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 data sets 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, *i.e.*, IMG and ITS, qualitatively outperform the TS approach in both real data sets, 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 data set that contains also a post-seismic signal in addition to the co-seismic offset, as shown in section 3.3.6. 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.

4

Slow slip detection with deep learning in multi-station raw geodetic time series validated against tremors in Cascadia

This chapter is based on the article in review at *Nature Communications Earth and Environment* as:

Costantino, G. and Giffard-Roisin, S. and Radiguet, M. and Dalla Mura, M. and Marsan, D. and Socquet, A. (2023). *Slow slip detection with deep learning in multi-station raw geodetic time series validated against tremors in Cascadia*, in review.

The reader will find the methods at the end of the chapter, according to the paper's structure.

In this chapter, we present SSEgenerator and SSEdetector, a deep learning-based approach to detect slow slip events in GNSS time series in the Cascadia subduction zone. SSEgenerator is used to generate the training database, made of synthetic GNSS time series. These time series contain synthetic noise, and a subset of them also include synthetic slow slip events. In real GNSS data from 2007 to 2023, our method detects 78 slow slip events which correlate well with tremor episodes. The 87% of previously detected SSEs are retrieved and a fraction of new detections may be associated with new, previously undetected, events.

Abstract Slow slip events (SSEs) originate from a slow slippage on faults that lasts from a few days to years. A systematic and complete mapping of SSEs is key to characterizing the slip spectrum and understanding its link with coeval seismological signals. Yet, SSE catalogues are sparse and usually remain limited to the largest events, because the deformation transients are often concealed in the noise of the geodetic data. Here we present a multi-station deep learning SSE detector applied blindly to multiple raw (non-post-processed) geodetic time series. Its power lies in an ultra-realistic synthetic training set, and in the combination of convolutional and attention-based neural networks. Applied to real data in Cascadia over the period 2007-2022, it detects 78 SSEs, that compare well to existing independent benchmarks: 87.5% of previously catalogued SSEs are retrieved, each detection falling within a peak of tremor activity. Our method also provides useful proxies on the SSE duration and may help illuminate relationships between tremor chatter and the nucleation of the slow rupture. We find an average day-long time lag between the slow deformation and the tremor chatter both at a global- and local-temporal scale, suggesting that slow slip may drive the rupture of nearby small asperities.

4.1 Introduction

Slow slip events (SSEs) generate episodic deformation that lasts from a few days to years (Behr & Bürgmann, 2021). Like earthquakes, they originate from slip on faults but, unlike them, do not radiate energetic seismic waves. In the mid-1990s, Global Navigation Satellite System (GNSS) networks started to continuously monitor the ground displacement, providing evidence that SSEs are a major mechanism responsible for the release of stress in plate boundaries, as a complement to seismic rupture (Dragert et al., 2001a; Ide, Beroza, et al., 2007b; Lowry et al., 2001b; Mousavi et al., 2020b; Schwartz & Rokosky, 2007b). This constituted a change of paradigm for the understanding of the earthquake cycle and of the mechanics of the fault interface. Twenty years later, the characterization of the full slip spectrum and the understanding of the link between slow slip and the associated seismological signals are hindered by our capacity to detect slow slip events in a systematic manner, more particularly those of low magnitude (typically lower than M_w 6), even though a systematic and complete mapping of SSEs on faults is key for understanding the complex physical interactions between slow aseismic slip and earthquakes. Indeed, the small deformation transients associated with an SSE are often concealed in the noise (Frank, Radiguet, et al., 2015; Rousset et al., 2017a), making it difficult to precisely characterize the slip spectrum and provide fruitful insights into the

fault mechanics (Gomberg et al., 2016a; Hawthorne & Bartlow, 2018; Ide, Beroza, et al., 2007b). Studies dealing with the detection and analysis of SSEs often rely on dedicated signal analysis, involving visual inspection of the data, data selection, denoising, filtering, geodetic expertise, dedicated modeling methods with a fine-tuning of the parameters, and also often complementary data such as catalogues of tremor or low-frequency earthquakes (LFEs) (Bartlow et al., 2011; Frank & Brodsky, 2019; Frank, Radiguet, et al., 2015; Michel, Gualandi, & Avouac, 2019; Radiguet et al., 2012a). Tremors and LFEs are LFEs are weakly emergent micro-seismicity, or micro-seismicity with low frequency content often accompanying SSEs in certain subduction zones (Obara, 2002; Rogers & Dragert, 2003b).

The development of in-situ geophysical monitoring generates nowadays huge data sets, and machine learning techniques have been largely assimilated and used by the seismological community to improve earthquake detection and characterization (Kong et al., 2019b; Mousavi et al., 2020b; Woollam et al., 2022; Zhu & Beroza, 2019), generating catalogs with unprecedented high quality (Ross, Trugman, et al., 2019; Tan et al., 2021) and knowledge shifts (Ross et al., 2020; Tan & Marsan, 2020). However, up to now, such techniques could not be successfully applied to the analysis of geodetic data and slow slip event detection because of two main reasons: (1) too few true labels exist to train machine learning-based methods, which we tackle by generating a realistic synthetic training data set, (2) the signal-to-noise ratio is extremely low in geodetic data (Costantino et al., 2023b; Rouet-Leduc et al., 2021b), meaning that we are at the limit of detection capacity. One possibility is to first pre-process the data to remove undesired signals such as seasonal variations, common modes, post-seismic relaxation signals (via denoising, filtering, trajectory modeling or independent-component-analysis-based inversion), but this is at the cost of possibly corrupting the data. Trajectory models are often used to subtract the contribution due to seasonal variations in the GNSS time series. Yet, this can lead, in some cases, to poor noise characterization since seasonal variations can have different amplitudes in the whole time series and a signal shape that is not necessarily fully reproduced by a simple sum of sine and cosine functions. This approach can thus introduce some spurious transient signals which could be erroneously modeled as slow slip events. Another possibility would be to use independent-component-analysis-based methods to extract the seasonal variations. However, the extracted components might contain some useful signals as well as some noise characteristics which would not be removed (e.g., specific harmonics or spatiotemporal patterns). Hence, the main motivation of this work is that we do not want to model any GNSS-constitutive signals, but use the non-post-processed GNSS time series as they are, by relying on a deep learning model which should

be able to learn the noise signature, and therefore separate the noise from the relevant information (here, slow slip events). In order to develop an end-to-end model (which does not require any manual intermediate step) capable of dealing with non-post-processed geodetic measurements, it is necessary, on one hand, to set up advanced methods to generate realistic noise, taking into account the spatial correlation between stations as well as the large number of data gaps present in the GNSS time series. On the other hand, it involves developing a specific deep learning model able to treat multiple stations simultaneously, using a relevant spatial stacking of the signals (driven by our physics-based knowledge of the slow slip events) in addition to a temporal analysis. We address these two major drawbacks in our new approach and present SSEgenerator and SSEdetector, an end-to-end deep learning-based detector, combining the spatiotemporal generation of synthetic GNSS time series containing modeled slow deformation (SSEgenerator), and a Convolutional Neural Network (CNN) and a Transformer neural network with an attention mechanism (SSEdetector), that proves effective in detecting slow slip events in raw GNSS position time series from a large geodetic network containing more than 100 stations, both on synthetic and on real data.

Overall strategy

In this study, we build two tools: (1) SSEgenerator, which generates synthetic GNSS time series incorporating both realistic noise and the deformation signal due to synthetic slow slip events along the Cascadia subduction zone, and (2) SSEdetector, which detects slow slip events in GNSS time series by means of deep learning techniques. The overall strategy is to train SSEdetector to reveal the presence of a slow slip event in a fixed-size temporal window, here 60 days, and to apply the detection procedure on real GNSS data with a window sliding over time. We first create 60,000 samples of synthetic GNSS position time series using SSEgenerator, which are then fed to SSEdetector for the training process. Half of the training samples contain only noise, with the remaining half containing an SSE of various sizes and depths along the subduction interface. Thanks to this procedure, SSEdetector can evaluate the probability of whether or not an SSE is occurring in the window, yet it does not allow it to determine its position on the fault. It is important to note that the detection is also linked to the GNSS network in Cascadia. On real data, the detection is applied to each time step and provides the probability of occurrence of a slow slip event over time. We first apply this strategy to synthetic data to evaluate the detection power of the method. Then, we apply SSEdetector on real GNSS time series and compare our results with the SSE catalog from Michel et

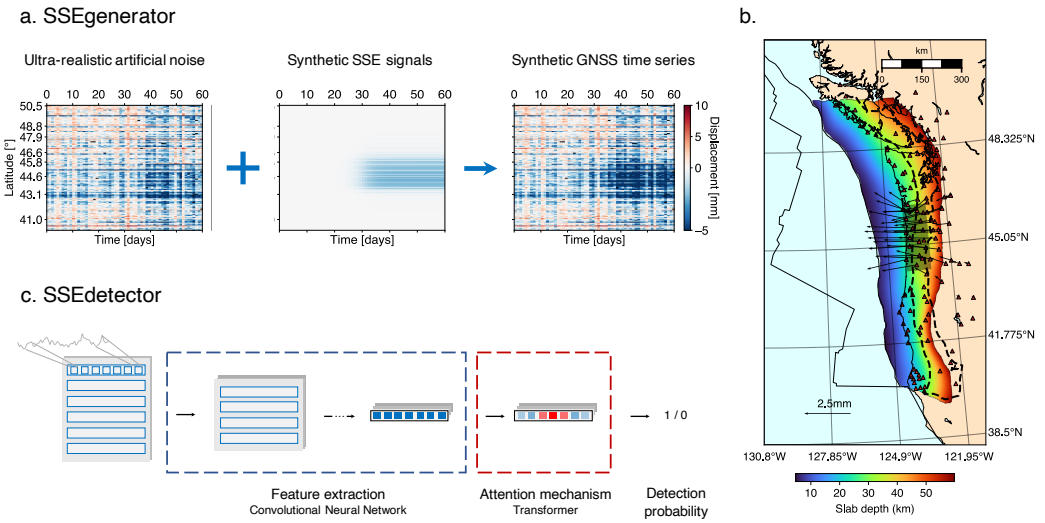


Fig. 4.1.: Schematic architecture of SSEgenerator and SSEdetector. (a) Overview of the synthetic data generation (SSEgenerator). In the matrix, each row represents the detrended GNSS position time series for a given station, color-coded by the value of the position. The 135 GNSS stations considered in this study are here shown sorted by latitude. The synthetic static displacement model (cf. (b) panel), due to a M_w 6.5 event, at each station is convolved to a sigmoid to model the SSE transient, and is added to the ultra-realistic artificial noise to build synthetic GNSS time series. (b) Location of the GNSS stations of MAGNET network used in this study (red triangles). An example of synthetic dislocation is represented by the black rectangle, with arrows showing the modeled static displacement field. The heatmap indicates the locations of the synthetic ruptures considered in this study, color-coded by the slab depth. The dashed black contour represents the tremor locations from the PNSN catalog. (c) High-level representation of the architecture of SSEdetector. The input GNSS time series are first convolved in the time domain. Then, the Transformer computes similarities between samples of each station, learning *self-attention* weights to discriminate between the relevant parts of the signals (here, slow slip transients) and the rest (*e.g.*, background noise) and a probability value is provided depending on whether slow deformation has been found in the data.

al. (2019), who used a different signal processing technique, and with the tremor rate.

4.2 Results

4.2.1 SSEgenerator: construction of the synthetic dataset

We choose the Cascadia subduction zone as the target region because (1) a link between slow deformation and tremor activity has been assessed (Rogers & Dragert,

2003b) and a high-quality tremor catalog is available (Wech, 2010); (2) a preliminary catalog of SSEs has recently been proposed during the period 2007-2017 with conventional methods (Michel, Gualandi, & Avouac, 2019). This proposed catalog will be used for comparison and baseline for our results, which are expected to provide a more comprehensive catalog that will better show the link between slow deformation and tremors.

To overcome the scarcity of catalogued SSEs, we train SSEdetector on synthetic data, consisting of simulated sets of geodetic time series for the full station network. Each group of signals (60 days and 135 stations) is considered a single input unit. In order to be able to detect SSEs in real raw time series, several characteristics need to be present in these synthetics. First, they must contain a wide range of realistic background signals at the level of the GNSS network, *i.e.* spatially and temporally-correlated realistic noise time series. On the other hand, while a subset of the samples (*negative* samples) will only consist of background noise, another subset must also include an SSE signal. Here, we use a training database containing half positive and half negative samples. For this, we modeled SSEs signals that are realistic enough compared to real transients of aseismic deformation. Finally, the synthetics should also carry realistic missing data recordings, as many GNSS stations have data gaps in practice.

First, we thus generate realistic synthetic time series, that reproduce the spatial and temporal correlated noise of the data acquired by the GNSS network, based on the method developed by Costantino et al. (Costantino et al., 2023b). This database of 60,000 synthetic time series is derived from real geodetic time series (details in section SSEgenerator: Generation of realistic noise time series). We select data in the periods 2007-2014 and 2018-2022 as sources for the noise generation, while we keep data in the period 2014-2017 as an independent test data set (details in section SSEgenerator: data selection).

In order to create the *positive* samples (time series containing an SSE), we model 30,000 dislocations (approximated as a point source) distributed along the Cascadia subduction interface (see Figure 4.1(b)) following the slab2 geometry (Hayes et al., 2018) (detailed procedure in section SSEgenerator: Modeling of synthetic slow slip events). The focal mechanism of the synthetic ruptures approximates a thrust, with rake angle following a uniform distribution (from 75 to 100°) and strike and dip defined by the geometry of the slab. The magnitude of the synthetic SSEs is drawn from a uniform probability distribution (from M_w 6 to 7). Their depths follow the slab geometry and are taken from 0 to 60 km, with further variability of ± 10 km. Such a variability on the depth allows us to better generalize over different slab

models and, most importantly, on their fitting to real GNSS data. We further assign each event a stress drop modeled from published scaling laws (H. Gao et al., 2012). We use the Okada dislocation model (Okada, 1985) to compute static displacements at each real GNSS station. We model synthetic SSE signals as sigmoidal-shaped transients, with duration following a uniform distribution (from 10 to 30 days). Eventually, we compute a database of 30,000 synthetic SSE transients, where the SSE signal is added to the positive samples (placed in the middle of the 60-day window).

The synthetic data set is thus made of 60,000 samples and labels, equally split into pure noise (labeled as 0) and signal (labeled as 1) with different nuances of signal-to-noise ratio, resulting both from different station noise levels and differences in magnitude and location, so that the deep learning method effectively learns to detect a variety of slow deformation transients from the background noise. The data set is further split into three independent training (60%), validation (20%) and test (20%) sets, with the latter being used after the training phase only.

4.2.2 SSEdetector: high-level architecture

SSEdetector is a deep neural network made of a CNN (LeCun et al., 2015) and a Transformer network (Vaswani et al., 2017) that are sequentially connected (detailed structure in section SSEdetector: Detailed architecture). We constructed the CNN to be a deep spatial-temporal encoder, that behaves as feature extractor. The structure of the encoder is a deep cascade of 1-dimensional temporal convolutional block sequences and spatial pooling layers. The depth of the feature extractor guarantees: (1) a high expressive power, *i.e.*, detailed low-level spatiotemporal features, (2) robustness to data gaps, since their propagation is kept limited to the first layers thanks to a cascade of pooling operators, and (3) limited overfit of the model on the station patterns, thanks to the spatial pooling operation. The decisive component of our architecture is the Transformer network, placed right after the deep CNN encoder. The role of the Transformer is to apply a temporal self-attention mechanism to the features computed by the CNN. As humans, we instinctively focus just on particular fragments of data when looking for any specific patterns. We wanted to replicate such a behavior in our methodology, leading to a network able to enhance crucial portions of the data and neglect the irrelevant ones. This is done by assigning a weight to the data, with those weights being learnt by finding significant connections between data samples and by relying on *a priori* knowledge of the labels, *i.e.*, whether there is a SSE or not. As a result, our Transformer has learnt (1) to precisely identify the timing of the aseismic deformation transients in the geodetic

time series and (2) to focus on it by assigning a weight close to zero to the rest of the time window. We further guide the process of finding slow deformation transients through a specific supervised-learning classification process. First, the disclosed outputs of the Transformer are averaged and passed through a sigmoid activation function. The output values are a detection probability lying in the (0,1) range and can be further interpreted as a confidence measure of the method. Second, we train SSEdetector by minimizing the binary cross-entropy loss between the target and the predicted labels (details in section Training details). The combination of the two strategies allows SSEdetector to be successfully applied in a real context because (1) we can run our detector on 1-day-shift windows of real data and collect an output value for each day used to build a temporal probability curve, (2) thanks to the Transformer neural network, such a curve will be smooth and the value of probability will gradually increase in time as SSEdetector identifies slow deformation in the geodetic data.

4.2.3 Application to the synthetic test set: detection threshold

We test SSEdetector against unseen synthetic samples and we analyze the results quantitatively. We build a synthetic test data set from GNSS data in the period 2018-2022 to limit the influence of data gaps (details in Method). We generate events down to magnitude 5.5 to better constrain the detection limit at low magnitudes. First, the probability p output by SSEdetector for each test sample must be classified (*i.e.*, either "noise" or "SSE"). We compute the ROC (receiving operating characteristic) curve as a function of the magnitude, with magnitudes higher than a given threshold, as shown in Figure 4.2(a). The curves represent the false positive rate (FPR, the probability of an actual negative being incorrectly classified as positive) as a function of the true positive rate (TPR, the probability that an actual positive will test positive). They are obtained by selecting the (positive) events whose magnitude is higher than a set of "limit" magnitudes with increments of 0.2. The negative sample set is the same for all the curves, so that each curve describes the increment of discriminative SSE power with respect to the noise Each point of the curve is computed with a different detection threshold p_θ, such that a sample is classified as SSE when $p > p_\theta$. Three threshold choices (0.4, 0.5, 0.6) are marked in Figure 4.2(a). As shown in the figure, the higher the magnitude, the higher the number of true positives. When the threshold is high (*e.g.*, 0.6) the model is more conservative: the false positive rate decreases, yet as well as the number of true positives. On the contrary, when the threshold is lower (*e.g.*, 0.4), more events can be correctly detected, at the cost of introducing more false positives. In this study, we choose

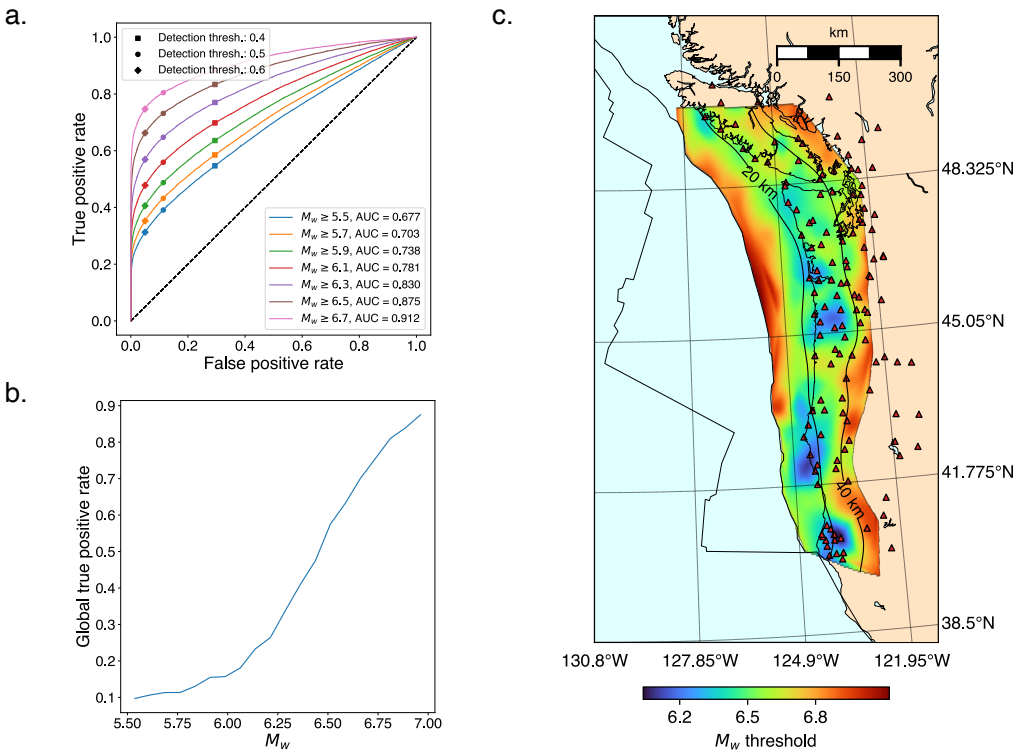


Fig. 4.2.: Performance of SSEdetector on synthetic data. (a) ROC (receiver operating characteristic) curve as a function of the magnitude greater than an incremental magnitude threshold (see lower legend), shown as different colors. The AUC (area under the curve) is shown in the legend for each curve (an AUC of 1 is associated with the perfect detector). For each curve, a detection threshold equal to 0.4, 0.5, and 0.6 is marked (see upper legend). (b) The blue curve represents the true positive rate (probability that an actual positive will test positive), computed on synthetic data, as a function of the magnitude. (c) Map showing the spatial distribution of the magnitude threshold for reliable detection, computed, for each spatial bin, as the minimum magnitude corresponding to a true positive rate value of 0.7.

$p_\theta = 0.5$ to have a trade-off between the ability to reveal potentially small SSEs (less false negatives, thus more true positives) and the introduction of false positives. This threshold will be used both for further analyses on synthetic data and for the detection of slow slip events in real data.

We obtain a further measure of the sensitivity of our model by computing the true positive rate as a function of the magnitude, as shown in Figure 4.2(b). On a global scale, the sensitivity is increasing with the signal-to-noise ratio (SNR), which also shows that it exists an SNR threshold limit for any SSE detection. This threshold is mainly linked to the magnitude of the event, rather than the moment rate. Thus, the ability of SSE detection is mostly influenced by the signal-to-noise ratio rather than the event duration (cf. Supplementary Figure 1). We compute the sensitivity as a function of the spatial coordinates of the SSE, by deriving a synthetic proxy as the magnitude threshold under which the TPR is smaller than 0.7 on a spatial neighborhood of approximately 50 km. We can see from Figure 4.2(c) that the detection power is related to the density of stations in the GNSS network, as well as to the distance between the rupture and the nearest station, and the rupture depth. When the density of GNSS stations is not high enough, our resolution power decreases as well as the reliability of the prediction. In those cases, we can only detect high-magnitude SSEs. This is also the case on the eastern side of the targeted region where the SSE sources are deeper because of the slab geometry (Figure 4.1(b)), even in locations where the density of stations is higher. In this case, the magnitude threshold increases because these events are more difficult to detect.

4.2.4 Continuous SSE detection in Cascadia from raw geodetic data during 2007-2022

4.2.4.1. Overall characteristics of the detected events

In order to evaluate how SSEdetector performs on real continuous data, we applied it to the raw GNSS time series in Cascadia for the period 2007-2022. SSEdetector scans the data with a 60-day sliding window (1-day stride), providing a probability of detection for the central day in each window. Figure 4.3(a) shows the probability of slow slip event detection (in blue) together with the tremor activity over the period 2007-2022 (in grey). We consider having a reliable detection when the probability value exceeds 0.5. We find 78 slow slip events over the period 2007-2022, with durations ranging from 2 to 79 days. We find 55 slow slip events in the period 2007-2017, to be compared with the 40 detections of the catalog of Michel et al.

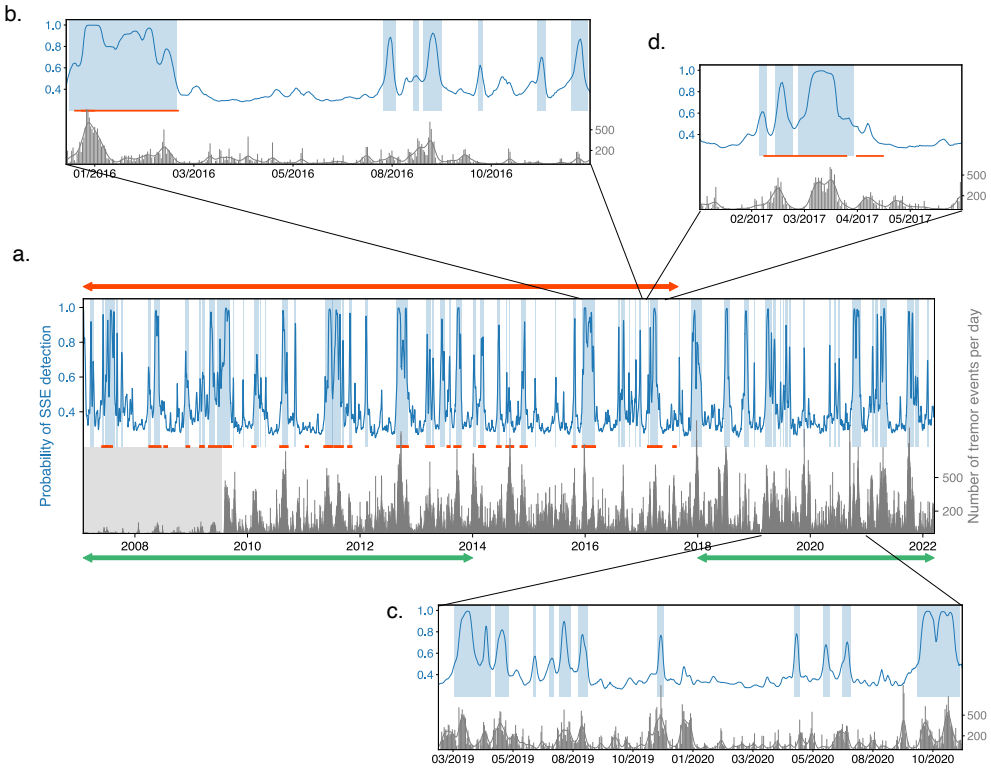


Fig. 4.3.: Overview of the performance of SSEdetector on real raw GNSS time series. The blue curves show the probability of detecting a slow slip event (output by SSEdetector) in 60-day sliding windows centered on a given date. Grey bars represent the number of tremors per day, smoothed (gaussian smoothing, $\sigma = 2$ days) in the grey curve. Red horizontal segments represent the known events catalogued by (Michel, Gualandi, & Avouac, 2019). The (a) panel shows the global performance of SSEdetector over 2007-2022. The red arrow indicates the time window analyzed by Michel et al., while the green arrows describe the two periods from which the synthetic training samples have been derived. The grey rectangle indicates the period which was not covered by the PNSN catalog. In this period, data from Ide, 2012 (Ide, 2012) has been used. The (b), (c) and (d) panels show zooms on 2016-2017, 2019-2021 and 2017 (January to July), respectively.

Tab. 4.1.: Comparison of the number detections from SSEdetector with respect to the catalog from Michel et al. (Michel, Gualandi, & Avouac, 2019). We distinguish detections in 2007-2017 (the same period analyzed by Michel et al.), and in 2017-2022. We further discriminate, in 2007-2017, between events in common with Michel et al., and new events in the same period.

Period		Method	
		Michel et al.	SSEdetector
2007-2017	Common with Michel et al.	40	35
	Not detected by Michel et al.	0	20
2017-2022		0	23

(Michel, Gualandi, & Avouac, 2019) (Table 4.1). We detect 35 of the 40 (87.5%) catalogued SSEs. Three of the missed SSEs have a magnitude smaller than 5.5, one of them has a magnitude of 5.86. The remaining one has a magnitude of 6.03. We show their location in Supplementary Figure 2, superimposed on the magnitude threshold map derived for SSEdetector (see Figure 4.2(b)). Given their location, the five missed events have magnitudes that are below the magnitude resolution limit (from 6 to 6.7, see Supplementary Figure 2). The remaining 20 events may be associated with new undetected SSEs. We also find 23 new events in the period 2017-2022, which was not covered by Michel et al. (Michel, Gualandi, & Avouac, 2019). We fixed the detection threshold to its default value of 0.5, *i.e.*, the model detects an event with a 50% confidence. Yet, this threshold can be modified to meet specific needs: if high-confidence detections are required, the threshold can be raised; conversely, it can be lowered to capture more events with lower confidence. Interestingly, the two SSEs from Michel et al. with magnitude 5.86 and 6.03, which were missed with a 0.5 probability threshold, can be retrieved when lowering it to 0.4, as shown in Supplementary Figure 18.

We also analyze the shape of the static displacement field in correspondence with the detected SSEs (cf. Supplementary Figure 3). We compute the static displacement field by taking the median displacement over three days and subtracting the displacement value at each station corresponding to the dates of the SSE. We find a good accordance with independent studies (Bletery & Nocquet, 2020; Itoh et al., 2022b; Michel, Gualandi, & Avouac, 2019). Moreover, many of the events found after 2018, as well as the new events detected in the period analyzed by Michel et al., have a displacement field suggesting that they are correct detections.

4.2.4.2. Analysis of the SSE durations

The shape of the probability curve gives insights into how SSEdetector reveals slow slip events from raw geodetic data. The probability curve in correspondence with an event has a bell shape: it grows until a maximum value, then it smoothly decreases when the model does not see any displacement associated with slow deformation in the data anymore. We use this property of the probability curve to extract a proxy on the detected SSE duration, based on the time span associated with the probability curve exceeding 0.5. We present the duration distribution in Figure 4.4 as well as a summary in Table 4.1. We detect most of the SSEs found by Michel et al., but we also find many more events, not only in the 2018-2022 period which was not investigated by Michel et al., but also within the 2007-2017 time window that they analyzed, suggesting that our method is more sensitive. We find potential

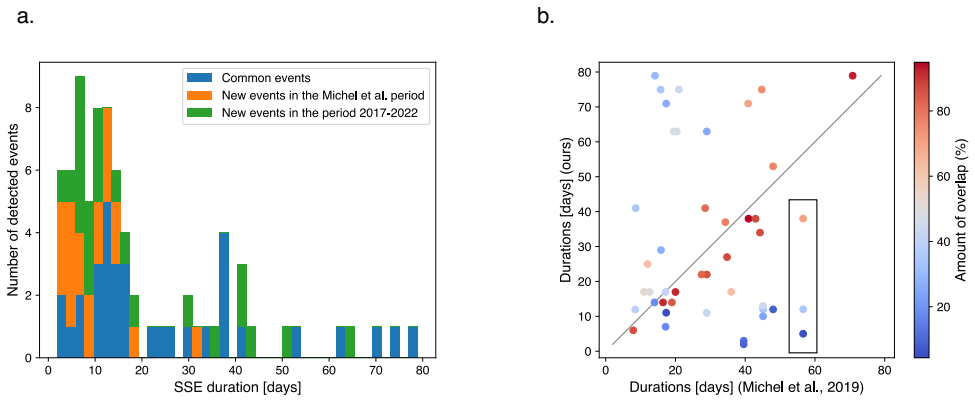


Fig. 4.4.: Distribution of the detected SSEs and comparison with the independent catalogue from Michel et al. (Michel, Gualandi, & Avouac, 2019). (a) Cumulative histogram of SSEdetector inferred durations. Blue bars represent the 35 catalogued events by Michel et al. (Michel, Gualandi, & Avouac, 2019) that have been successfully retrieved. Orange bars show the 20 additional events that have been discovered within the time window analyzed by Michel et al., while green ones represent the 23 events found in the time period 2017-2022, not covered by the catalog of Michel et al. (b) Event durations from Michel et al.'s catalogue with respect to the durations obtained by SSEdetector. Events are color-coded by the overlap percentage (details in section Overlap percentage calculation). The black rectangle represents an example of a large event found by Michel et al. (Michel, Gualandi, & Avouac, 2019), that is split into three sub-events by SSEdetector.

slow slip events at all scales of durations (from 2 to 79 days). Michel et al. hardly detect SSEs that last less than 15 days, probably due to temporal data smoothing (Michel, Gualandi, & Avouac, 2019), while we retrieve shorter events (less than 10 days) since we use raw time series, meaning that our method has a better temporal resolution. In Figure 4.4 (b), we show a comparison between the SSE durations of Michel et al.'s (Michel, Gualandi, & Avouac, 2019) catalogued events and ours. This plot is made by considering all the combinations between events in our catalog and in the Michel et al. one. Each horizontal alignment represents an event in our catalogue that is split into sub-events in the Michel et al. catalog, while vertical alignments show events in the Michel et al. catalog corresponding to sub-events in our catalog. We find that the durations are in good accordance for a large number of events, for which the overlap is often higher than 70%, both for small- and large-magnitude ones. We can also identify, from figure 4.4(b), that some events are separated in one method while identified as one single SSE in the other: this is the case for the 55 day-long event from Michel et al. (Michel, Gualandi, & Avouac, 2019), that was paired with 3 SSEdetector sub-events (see Figure 4.3(d) and the rectangle in Figure 4.4(b)). The majority of the points located off the identity line (the diagonal) are thus sub-events for which the grouping differs in the two catalogs. As more points are below the diagonal than above, we can see that SSEdetector tends to separate the detections more. We interpret this as a possible increase in the detection precision, yet a validation with an independent acquisition data set is needed, since the separation into sub-events strongly depends on the threshold applied to the detection probability to define a slow slip event (0.5 in this study). Also, a few points above the diagonal represent events that were split into sub-events by Michel et al., which our model tends to detect as a single event. This separation is also related to the choice of the threshold.

4.2.4.3. Validation against tremors

In order to have an independent validation, we compare our results with tremor activity from the Pacific Northwest Seismic Network (PNSN) catalog (Wech, 2010) between 2009-2022 and Ide's catalogue (Ide, 2012) catalog between 2006-2009, shown in grey in Figure 4.3. We show the location of the tremors in our catalogues with the dashed black contour in Figure 4.1(b). From a qualitative point of view, we can see that the detection probability curve seems to align well with the number of tremors per day, throughout the whole period. This is also true for the 20 possible new detected events that were not present in previous catalogs, for example during the period after 2017 (see Figure 4.3(c)), but also in 2016-2017, where we detect

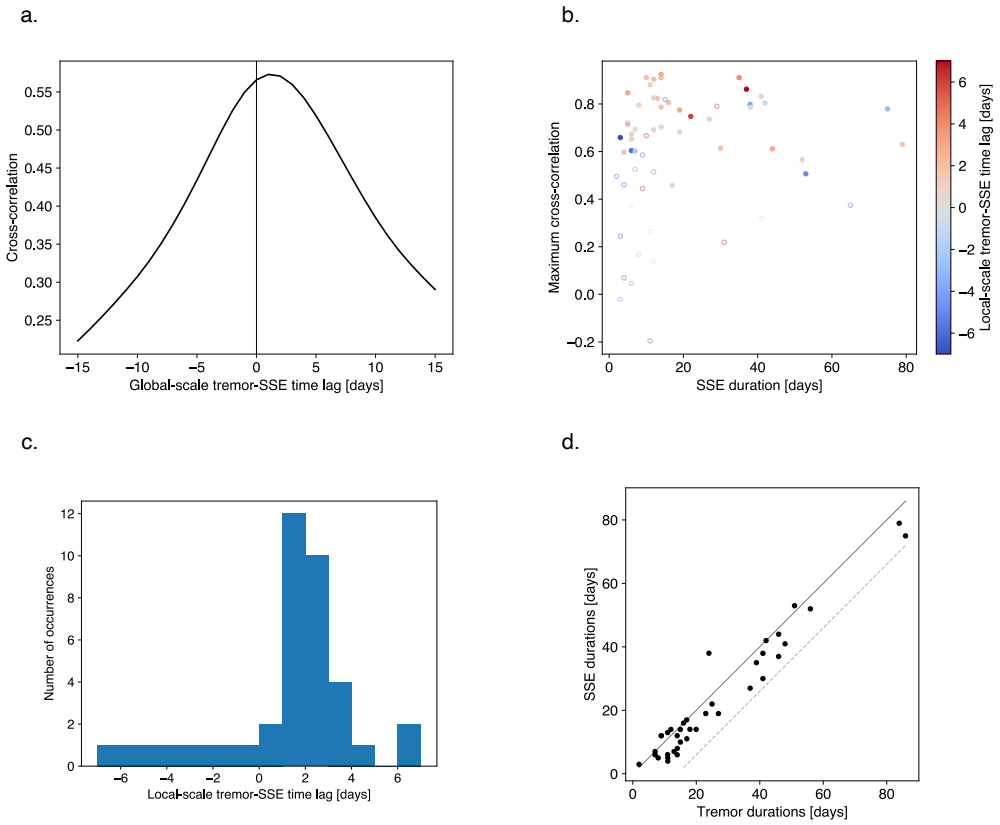


Fig. 4.5.: Validation of SSEdetector performance against tremor activity in 2010-2022. (a) Global-scale cross-correlation between the full-length SSEdetector output probability and the number of tremors per day, as a function of the time shift between the two curves. (b) Local-scale maximum value of cross-correlation for each SSE and tremor windows, centered on the SSE duration, as a function of the SSE duration, color-coded by the associated time lag between tremor and SSE (positive lag means deformation precedes tremor). Events having a zero-lag cross-correlation (correlation coefficient) lower than 0.4 are marked with an empty point. (c) Histogram of the time lags computed in the (b) panel. (d) SSE durations as a function of the tremor durations for the events in the (b) panel which have a correlation greater than 0.4. The solid black line represents the identity line, while the dashed grey line is the maximum tremor duration that can be attained for a given SSE duration, that is SSE duration + 14 days (see section "Computation of local- and global-scale correlations").

7 possible events that were not previously catalogued (see Figure 4.3(b)). The excellent similarity between tremors and our detections is quantitatively assessed by computing the cross-correlation between the probability curve and the number of tremors per day, the latter smoothed with a gaussian filter ($\sigma = 1.5$ days), as a function of the time shift between the two curves (Figure 4.5(a)). The interval 2007-2010 has been excluded from Figure 4.5(a) in order to consider the period covered by the PNSN catalog only. The maximum correlation value is around 0.58 and is obtained for a time shift between 1 and 2 days. This shows that, at a global scale, the probability peaks are coeval with the peaks of tremor activity.

We also make a further comparison at the local scale for each individual detected SSE. In Figure 4.5(b) we observe that most of the individual detected SSEs show a correlation larger than 0.4 with the coeval peak of tremor. SSE and tremor signals are offset by about 2 days on average (see Figure 4.5(c)). This result, obtained on windows of month-long scale, seems consistent with the decade-long correlation shown in Figure 4.5(a), suggesting that the found large-scale trend is also true at a smaller scale. This may suggest that the slow deformation, for which the detection probability is a proxy, precedes the tremor chatter by a few days, with potential implications on the nucleation of the slow rupture.

We compare the tremor peak duration (see details in section "Calculation of tremor durations") to the SSE duration in Figure 4.5(d) for all the events that have been also considered in Figure 4.5(b). The figure shows a correspondence between slow slip duration and coeval tremor activity duration: most of the events are associated with a peak of tremor activity of close duration. This is true also for large events, up to 80 days. This finding gives an insight that our deep learning-based method, blindly applied to raw geodetic time series, achieves reliable results. Yet, this result should be taken with caution, since it is strongly dependent on the choice of the window of observation (see sections "Calculation of tremor durations" and "Computation of local- and global-scale correlations" for further details).

4.2.5 Validity of the duration proxy against temporal smoothing: the March 2017 slow slip event

We test the first-order effectiveness of the duration proxy computation to assess any potential temporal smoothing effects. We focus on the M_w 6.7 slow slip event that occurred on March 2017. We rely both on the measured time series and on the kinematic model of the slip evolution by Itoh et al. (Itoh et al., 2022b). Figure 4.6 (a) shows the displacement field output by Itoh et al.'s model and its temporal evolution

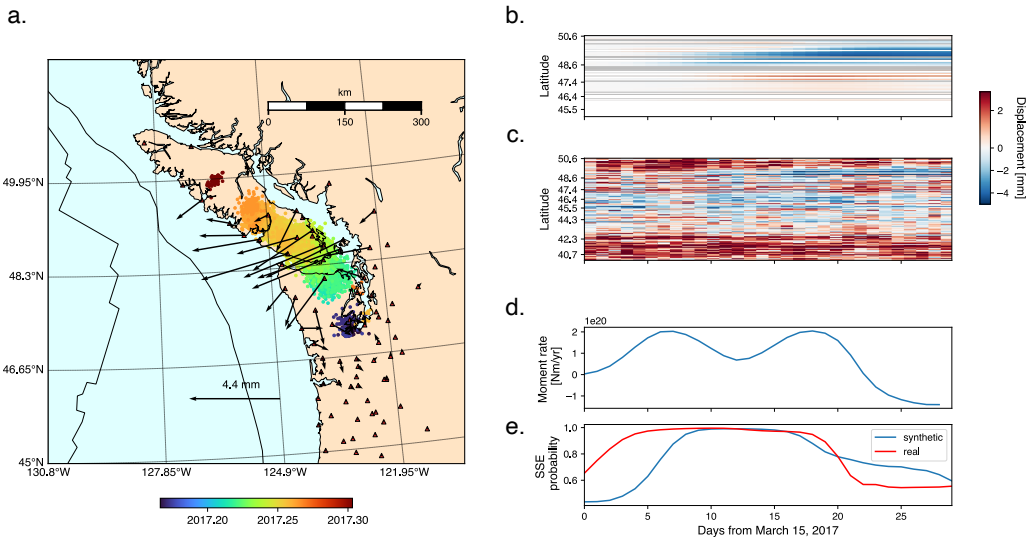


Fig. 4.6.: Temporal smoothing analysis on a synthetic model of the March 2017 slow slip event. (a) Model of the static displacement associated with the March 2017 slow slip event by Itoh et al. (Itoh et al., 2022b) (black arrows). The red triangles represent the 135 stations used in this study and the points show the tremor location from March 1 to April 30, 2017, color-coded by date of occurrence. (b) Matrix showing the temporal evolution of Itoh et al.'s model. Each row represents the detrended E-W GNSS position time series for a given station, color-coded by the position value. (c) Detrended E-W synthetic time series (sum of the Itoh et al.'s model and a realization of artificial noise output by SSEgenerator, color-coded by position value. (d) Moment rate function associated with the slip evolution of the model by Itoh et al. (e) The blue curve represents the daily probability output by SSEdetector from the synthetic time series shown in the (c) panel. The red curve shows the probability curve associated with the prediction of SSEdetector on the March 2017 slow slip event on real GNSS data (see also Figure 4.3(d)).

is shown in Figure 4.6 (b). We use SSEgenerator to build a random noise time series, to which we add the modeled displacement time series to build a synthetic time series reproducing the March 2017 SSE, as shown in Figure 4.6 (c). Figure 4.6 (e) shows the probability curve output by SSEdetector on the synthetic time series (in blue), to be also compared to the prediction associated with the same event in real GNSS data (in red, see also Figure 4.3 (d)). By the analysis of the probability curve, we estimate a duration of 28 days, comparable with the duration inferred by Itoh et al. (Itoh et al., 2022b) (30 days). A 3-day time shift is found between the probability curve and the slip evolution, represented by moment rate function in Figure 4.6 (d), which can be imputed to the specific realization of synthetic noise. When comparing the probability curve associated with the same event as computed from real data (red curve in 4.6 (e)), we find a better resemblance, consistent also with the catalog from Michel et al. (see Figure 4.3 (d)), which demonstrates that SSEdetector does not suffer from any first-order temporal smoothing issues.

4.2.6 Robustness of SSEdetector to variations in the source characteristics

We perform an extensive study to test how SSEdetector performs on SSEs that exhibit characteristics in the source parameters that differ from the synthetic data set used for the training. We train SSEdetector on synthetic dislocations with fixed aspect ratios and shapes. However, SSEs in Cascadia generally have aspect ratios between 3 and 13 and exhibit heterogeneous slip along strike with pulse-like propagation (Michel, Gualandi, & Avouac, 2019; Schmidt & Gao, 2010). We create a synthetic test sample containing three slow slip events that propagate in space and time, to simulate a real large SSE that lasts 60 days and propagates southwards, as shown in Supplementary Figure 19. The detection probability and the associated duration proxy suggest that the probability curve can still be used to retrieve SSEs that last longer than 30 days. Thanks to its multi-station approach, our method detects slices of slow slip events through time and space (station dimension), regardless of the training configuration. This can be seen as a stage in which some building blocks are provided to the model, that combines them during the testing phase to detect events having different shapes and sizes. For this reason, SSEdetector can be used to generalize, at first order, over more complex events in real data without the need to build a complex train data set, which should also account for realistic event propagation mechanisms, yet difficult to model.

We also test the generalization ability of SSEdetector over shorter (less than 10 days) events. We build a synthetic sample of a 3-day slow slip event. As shown in Supplementary Figure 20, SSEdetector is able to correctly detect it, with an inferred duration of 4 days, suggesting that it has learned what is the first-order temporal signature of the signals of interest, being able to detect them also at shorter scales.

SSEdetector is applied to real data by following a sliding-window strategy. In real data, two events could be found in the same window, most likely one in the northern and the other in the southern part. For this, we build two synthetic tests, the first test with two 20-day-long events that do not occur at the same time, and the second one with two 20-day-long synchronous events. Supplementary Figures 20 and 21 show the results for both cases, which suggest that SSEdetector can generalize well also when there are two events in the same window, also corroborating the results found on real data.

4.2.7 Sensitivity study

We analyze the sensitivity of SSEdetector with respect to the number of stations. We construct an alternative test selecting 352 GNSS stations (see Supplementary Figure 5), which is the number of stations used by Michel et al. (Michel, Gualandi, & Avouac, 2019). The 217 extra stations have larger percentages of missing data compared to the initial 135 stations (cf. Supplementary Figure 4). We train and test SSEdetector with 352 time series and we report the results in Supplementary Figures 6-7. We observe that the results are similar, with an excellent alignment with tremors and similar correlation and lag values, although with this setting the detection power slightly decreases, probably due to a larger number of missing data, suggesting that there is a tradeoff between the number of GNSS stations and the number of missing data.

We also test the ability of SSEdetector to identify SSEs in a sub-region only (even if it is trained with a large-scale network. For that, we test SSEdetector (trained on 135 stations), without re-training, on a subset of the GNSS network, situated in the northern part of Cascadia. To this end, we replace with zeros all the data associated with stations located at latitudes lower than 47 degrees (see Supplementary Figure 8). Similarly, we find that SSEdetector retrieves all the events which were found by Michel et al. (Michel, Gualandi, & Avouac, 2019) and the correlation with tremors that occur in this sub-region is still high, with a global-scale cross-correlation of 0.5 (cf. Supplementary Figures 9-10). This means that the model is robust against long periods of missing data and, thanks to the spatial pooling strategy, can generalize

over different settings of stations and obtain some information on the localization. We found similar results for SSEs in the southern part, with a lower correlation with tremors, probably because of the higher noise level in south Cascadia, as shown in Supplementary Figures 16-17.

Finally, we test SSEdetector against other possible deep learning models that could be used for detection. We report in Supplementary Figures 11-12 the results obtained by replacing the one-dimensional convolutional layers with two-dimensional convolutions on time series sorted by latitude (as shown in Figure 4.1(a)). This type of architecture was used in studies having similar multi-station time-series data (Licciardi et al., 2022). We observe that the results on real data are not satisfactory because of too high a rate of false detections and a lower temporal resolution than SSEdetector (in other words, short SSEs are not retrieved). This suggests that our specific model architecture, handling in different ways the time dimension and the station dimension, might be more suited to multi-station time-series data sets.

4.2.8 Discussion

In this study, we use a multi-station approach that proves efficient in detecting slow slip events in raw GNSS time series even in presence of SSE migrations (Bletery & Nocquet, 2020; Itoh et al., 2022b; Michel, Gualandi, & Avouac, 2019). Thanks to SSEdetector, we are able to detect 87 slow slip events with durations from 2 to 79 days, with an average limit magnitude of about 6.4 in north Cascadia and 6.2 in south Cascadia computed on the synthetic test set (see Figure 4.2(b)). The magnitude of the smallest detected SSE in common with Michel et al. is 5.42, with a corresponding duration of 8.5 days. One current limitation of this approach is that the location information is not directly inferred. In this direction, some efforts should be made in developing a method for characterizing slow slip events after the detection in order to have information on the location, but also on the magnitude, of the slow rupture.

We apply our methodology to the Cascadia subduction zone because it is the area where independent benchmarks exist and it is thus possible to validate a new method. However, the applicability of SSEgenerator and SSEdetector to other subduction zones is possible. The current approach is, however, region-specific. In fact, the characteristics of the targeted zone affect the structure of the synthetic data, thus a method trained on a specific region could have poor performance if tested on another one without retraining. This problem can be addressed by generating multiple data sets associated with different regions and combining them for the training. Also,

we focus on the Cascadia subduction zone, where not much regular seismicity occurs, making it a prototypical test zone when looking for slow earthquakes. When addressing other regions, such as Japan, for example, the influence of earthquakes or post-seismic relaxation signals could make the problem more complex. This extension goes beyond the scope of this study, yet we think that it will be essential to tackle this issue in order to use deep learning approaches for the detection of SSEs in any region.

4.3 Conclusions

We developed a powerful pipeline, consisting of a realistic synthetic GNSS time-series generation, SSEgenerator, and a deep-learning classification model, SSEdetector, aimed to detect slow slip events from a series of raw GNSS time series measured by a station network. We built a new catalog of slow slip events in the Cascadia subduction zone by means of SSEdetector. We found 78 slow slip events from 2007 to 2022, 35 of which are in good accordance with the existing catalog (Michel, Gualandi, & Avouac, 2019). The detected SSEs have durations that range between a few days to a few months. The detection probability curve correlates well with the occurrence of tremor episodes, even in time periods where we found new events. The duration of our SSEs, for the 35 known events, as well as for the 43 new detections, are found to be similar to the coeval tremor duration. The comparison between tremors and SSEs also shows that, both at a local and a global temporal scale, the slow deformation may precede the tremor chatter by a few days, with potential implications on the link between a slow slip that could drive the rupture of nearby small seismic asperities. This is the first successful attempt to detect SSEs from raw GNSS time series, and we hope that this preliminary study will lead to the detection of SSEs in other active regions of the world.

4.4 Methods

4.4.1 SSEgenerator: data selection

We consider the 550 stations in the Cascadia subduction zone, belonging to the MAGNET GNSS network, and we select data from 2007 to 2022. We train SSEdetector with synthetic data whose source was affected by different noise and data gap patterns. We divide the data into three periods: 2007-2014, 2014-2018, and

2018-2022. In order to create a more diverse training set, data in the period 2007-2014 and 2018-2022 has been chosen as a source for synthetic data generation. The period 2014-2018 was left aside and used as an independent validation set for performance assessment on real data. Nonetheless, since synthetic data is performed by applying a methodology based on randomization (SSEgenerator), a test on the whole sequence 2007-2022 is possible without overfitting.

For the two periods 2007-2014 and 2018-2022, we sort the GNSS stations by the total number of missing data points and we choose the 135 stations affected by fewer data gaps as the final subset for our study. We make sure that stations having too high a noise do not appear in this subset. We select 135 stations since it represents a good compromise between the presence of data and the longest data gap sequence in a 60-day window. However, we also train and test SSEdetector on 352 stations (the same number used in the study by Michel et al. (Michel, Gualandi, & Avouac, 2019)) to compare with a setting equivalent to the one from Michel et al. (Michel, Gualandi, & Avouac, 2019). We briefly discuss the results in the section "Sensitivity study".

4.4.2 SSEgenerator: Generation of realistic noise time series

Raw GNSS data is first detrended at each of the 135 stations, *i.e.* the linear trend is removed, where the slope and the intercept are computed, for each station, without taking into account the data gaps, *i.e.*, for each station the mean over time is calculated without considering the missing data points, and is removed from the series. Only the linear trend is removed from the data, without accounting for seasonal variations, common modes, or co- and post-seismic relaxation signals, which we want to keep to have a richer noise representation. We remove the linear trend for each station to avoid biases in our noise generation strategy, since the variability due to the linear trend would be captured as a principal component thus introducing spurious piece-wise linear signals in the realistic noise. We build a matrix containing all station time series $\mathbf{X} \in \mathbb{R}^{N_t \times N_s}$, where N_t is the temporal length of the input time series and N_s is the number of stations. In this study, we use 2 components (N-S and E-W) and we apply the following procedure for each component independently. Each column of $\mathbf{X}$ contains a detrended time series. We proceed as follows. The $\mathbf{X}$ matrix is then re-projected in another vector space through a Principal Component Analysis (PCA), as follows. First, the data is centered. The mean vector is computed $\boldsymbol{\mu} \in \mathbb{R}^{N_s}$, such that μ_i is the mean of the i-th time series. The centered matrix is considered $\tilde{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}$, and is decomposed through Singular Value Decomposition (SVD) to obtain the matrix of right singular vectors $\mathbf{V}$,

which is the rotation matrix containing the spatial variability of the original vector space. We further rotate the data by means of this spatial matrix to obtain spatially-uncorrelated time series $\hat{\mathbf{X}} = \tilde{\mathbf{X}}\mathbf{V}$. Then, we produce $\hat{\mathbf{X}}_R$, a randomized version of $\hat{\mathbf{X}}$, by applying the iteratively-refined amplitude-adjusted Fourier transform (AAFT) method (Schreiber & Schmitz, 2000), having globally the same power spectrum and amplitude distribution of the input data. The number of AAFT iterations has been set to 5 based on preliminary tests. The surrogate time series are then back-projected in the original vector space to obtain $\mathbf{X}_R = \hat{\mathbf{X}}_R\mathbf{V}^T + \boldsymbol{\mu}$. We further enrich the randomized time series $\mathbf{X}_R$ by imprinting the real pattern of missing data for 70 % of the synthetic data. We shuffle the data gaps before imprinting them to the data, assigning a station a data-gap sequence belonging to another station, such that SSEdetector can better generalize over unseen test data for the same station, which necessarily would have a different pattern of data gaps. We leave the remaining 30% of the data as it is. We prefer not to use any interpolation method to avoid to introduce new values in the data. Thus, we set all the missing data points to zero, which is a neutral value with respect to the trend of the data and the convolution operations performed by SSEdetector.

After this process, we generate sub-windows of noise time series as follows. Given the window length W_L, a uniformly distributed random variable is generated $s \sim \mathcal{U}(-\frac{W_L}{2}, \frac{W_L}{2})$ and the data is circularly shifted, in time, by the amount s. Then, $\lfloor \frac{N_t}{W_L} \rfloor$ contiguous (non-overlapping) windows are obtained. The circular shift is needed in order to prevent SSEdetector from learning a fixed temporal pattern of data gaps. Finally, by knowing the desired number N of noise windows to compute, the surrogate generation $\{\mathbf{X}_R\}_i$ can be repeated $\lceil \frac{N}{\lfloor N_t/W_L \rfloor} \rceil$ times. In our study, we generate $N = 60,000$ synthetic samples, by calling the surrogate data generation 1,429 times and extracting 42 non-overlapping noise windows from each randomized time series.

4.4.3 SSEgenerator: Modeling of synthetic slow slip events

We first generate synthetic displacements at all the 135 selected stations using Okada's dislocation model (Okada, 1985). We draw random locations (fault centroids), strike and dip angles using the slab2 model (Hayes et al., 2018) following the subduction geometry within the area of interest (see Figure 4.1(b)). We let the rake angle be a uniform random variable from 75 to 100 degrees, in order to have a variability around 90 degrees (thrust focal mechanism). For each (latitude, longitude) couple, we extract the corresponding depth from the slab and we add further variability, modeled as a uniformly distributed random variable from -10 to

10 km. We allow for this variability if the depth is at least 15 km, to prevent the ruptures from reaching the surface. We associate each rupture with a magnitude M_w, uniformly generated in the range (6, 7), and we compute the equivalent moment as $M_0 = 10^{1.5M_w+9.1}$. As for the SSE geometry, we rely on the circular crack approximation (Lay & Wallace, 1995) to compute the SSE radius as:

$$R = \left(\frac{7}{16}\frac{M_0}{\Delta\sigma}\right)^{1/3} \tag{4.1}$$

where $\Delta\sigma$ is the static stress drop. We compute the average slip on the fault as:

$$\bar{u} = \frac{16}{7\pi}\frac{\Delta\sigma}{\mu}R \tag{4.2}$$

where μ is the shear modulus. We assume $\mu = 30$ GPa. By imposing that the surface of the crack must equal a rectangular dislocation of length L and width W, we obtain $L = \sqrt{2\pi}R$. We assume that $W = L/2$. Finally, we model the stress drop as a lognormally-distributed random variable. We assume the average stress drop to be $\overline{\Delta\sigma} = 0.05$ MPa for the Cascadia subduction zone (H. Gao et al., 2012). We also assume that the coefficient of variation c_V, namely the ratio between the standard deviation and the mean, is equal to 10. Hence, we generate the static stress drop as $\Delta\sigma \sim$ Lognormal(μ_N, σ_N^2), where μ_N and σ_N are the mean and the standard deviation of the underlying normal distribution, respectively, that we derive as:

$$\sigma_N = \sqrt{\ln\left(c_V^2 + 1\right)} \tag{4.3}$$

and

$$\mu_N = \ln\left(\overline{\Delta\sigma}\right) - \sigma_N^2/2. \tag{4.4}$$

We thus obtain the (horizontal) synthetic displacement vector $\mathbf{D}_s = (D_s^{N-S}, D_s^{E-W})$ at each station s. We model the temporal evolution of slow slip events as a logistic function. Let D be the E-W displacement for simplicity. In this case, we model an SSE signal at a station s as:

$$d_s(t) = \frac{D}{1 + e^{-\beta(t-t_0)}} \tag{4.5}$$

where β is a parameter associated with the growth rate of the curve and t_0 is the time corresponding to the inflection point of the logistic function. We assume $t_0 = 30$ days, so that the signal is centered in the 60-day window. We derive the parameter β as a function of the slow slip event duration T. We can rewrite the duration as $T = t_{max} - t_{min}$, where t_{max} is the time corresponding to the post-SSE value of the signal (*i.e.*, D), while t_{min} is associated to the pre-SSE displacement (*i.e.*, 0). Since these values are only asymptotically reached, we introduce a threshold γ, such that t_{max} and t_{min} are associated with $d_s(D - \gamma D)$ and $d_s(\gamma D)$, respectively. We choose $\gamma = 0.01$. By rewriting the duration as $T = t_{max} - t_{min}$ and solving for β, we obtain:

$$\beta = \frac{2}{T} \ln \left(\frac{1}{\gamma} - 1 \right). \tag{4.6}$$

Finally, we generate slow slip events having uniform duration T between 10 and 30 days. We take half of the noise samples (30,000) and we create a positive sample (*i.e.*, time series containing a slow slip event) as $\mathbf{X}_R + \mathbf{d}(t)$, where $\mathbf{d}(t)$ is a matrix containing all the modeled time series $d_s(t)$ for each station. We let $\mathbf{X}_R$ contain missing data. Therefore, we do not add the signal $d_s(t)$ where data should not be present.

4.4.4 SSEdetector: Detailed architecture

SSEdetector is a deep neural network obtained by the combination of a convolutional and a Transformer neural network. The full architecture is shown in Supplementary Figure 13. The model takes input GNSS time series, which can be grouped as a matrix of shape (N_s, N_t, N_c), where N_s, N_t, N_c are the number of stations, window length and number of components, respectively. In this study, $N_s = 135, N_t = 60$ days and $N_c = 2$ (N-S, E-W). The basic unit of this CNN is a Convolutional Block. It is made of a sequence of a one-dimensional convolutional layer in the temporal (N_t) dimension, which computes N_f feature maps by employing a 1×5 kernel, followed by a Batch Normalization (Ioffe & Szegedy, 2015) and a ReLu activation function (Agarap, 2018). We will refer to this unit as ConvBlock(N_f) for the rest of the paragraph (see Supplementary Figure 13). We alternate convolutional operations in the temporal dimension with pooling operations in the station dimension (max-pooling with a kernel of 3) and we replicate this structure as long as the spatial (station) dimension is reduced to 1. To this end, we create a sequence of 3 ConvBlock($\cdot$) + max-pooling. As an example, the number of stations after the first pooling layer is reduced from

135 to 45. At each ConvBlock$(\cdot)$, we multiply by 4 the number of computed feature maps N_f. At the end of the CNN, the computed features have shape (N_t, N_f^{final}), with $N_f^{final} = 256$.

This feature matrix is given as input to a Transformer neural network. We first use a Positional Embedding to encode the temporal sequence. We do not impose any kind of pre-computed embedding, but we use a learnable matrix of shape (N_t, N_f^{final}). The learnt embeddings are added to the feature matrix (*i.e.*, the output of the CNN). The embedded inputs are then fed to a Transformer neural network (Vaswani et al., 2017), whose architecture is detailed in Supplementary Figure 14. Here, the global (additive) self-attention of the embedded CNN features is computed as:

$$\eta_{t_1,t_2} = W_a \tanh\left(W_1^T h_{t_1} + W_2^T h_{t_2} + b_h\right) + b_a \ , \tag{4.7}$$

$$a_{t_1,t_2} = \text{softmax}\left(\eta_{t_1,t_2}\right) = \frac{e^{\eta_{t_1,t_2}}}{\sum_{t_2} e^{\eta_{t_1,t_2}}} \ , \tag{4.8}$$

$$c_{t_1} = \sum_{t_2} a_{t_1,t_2} \cdot h_{t_2} \ , \tag{4.9}$$

where W represents a learnable weight matrix and b a bias vector. The matrices h_{t_1} and h_{t_2} are the hidden-state representations at time t_1 and t_2, respectively. The matrix a_{t_1,t_2} contains the attention scores for the time steps t_1 and t_2. Here, a context vector is computed as the weighted sum of the hidden-state representations by the attention scores. The context vector contains the importance at a given time step based on all the features in the window. The contextual information is then added to the Transformer inputs. Then, a position-wise Feed-Forward layer (with a dropout rate of 0.1) is employed to add further non-linearity. After the Transformer network, a Global Average Pooling in the temporal dimension (N_t) is employed to gather the transformed features and to output a vector summarizing the temporal information. A Dropout is then added as a form of regularization to reduce overfitting (Srivastava et al., 2014), with dropout rate $\delta = 0.2$. In the end, we use a fully-connected layer with one output, with a sigmoid activation function to express the probability of SSE detection.

4.4.5 Training details

We perform a mini-batch training (Bottou et al., 2018) (batch size of 128 samples) by minimizing the binary cross-entropy (BCE) loss between the target labels y and the predictions $\hat{y}$ (a probability estimate):

$$\text{BCE}(y, \hat{y}) = -y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y}). \tag{4.10}$$

The BCE loss is commonly used for binary classification problems (detection is a binary classification). We use the ADAM method for the optimization (Kingma & Ba, 2014) with a learning rate $\lambda = 10^{-3}$ which has been experimentally chosen. We schedule the learning rate such that it is reduced during training iterations and we stop the training when the validation loss did not improve for 50 consecutive epochs. We initialized the weights of SSEdetector with a uniform He initializer (K. He et al., 2015). We implemented the code of SSEdetector in Python using the Tensorflow and Keras libraries (Abadi et al., 2016; Chollet et al., 2015). We run the training on NVIDIA Tesla A100 Graphics Processing Units (GPUs). The training of SSEdetector takes less than 2 hours. The inference on the whole 15-year sequence (2007-2022) takes a few seconds.

4.4.6 Calculation of tremor durations

We compute the durations of tremor bursts using the notion of topographic prominence, explained in the following. We rely on the software implementation from the SciPy Python library (Virtanen et al., 2020). Given a peak in the curve, the topographic prominence is informally defined as the minimum elevation that needs to be descended to start reaching a higher peak. The procedure is graphically detailed in Supplementary Figure 15. We first search for peaks in the number of tremors per day by comparison with neighboring values. In order to avoid too many spurious local maxima, we smooth the number of tremors per day with a gaussian filter ($\sigma = 1.5$ days). For each detected SSE, we search for peaks of tremors in a window given by the SSE duration ± 3 days. For each peak of tremors that is found, the corresponding width is computed as follows. The topographic prominence is computed by placing a horizontal line at the peak height h (the value of the tremor curve corresponding to the peak). An interval is defined, corresponding to the points where the line crosses either the signal bounds or the signal at the slope towards a higher peak. In this interval, the minimum values of the signal on each side are computed, representing the bases of the peak. The topographic prominence p of

the peak is then defined as the height between the peak and its highest base value. Then, the local height of the peak is computed as $h_L = h - \alpha \cdot p$. We set $\alpha = 0.7$ in order to focus on the main tremor pulses, discarding further noise in the curve. From the local height, another horizontal line is considered and the peak width is computed as the intersection point of the line with either a slope, the vertical position of the bases or the signal bounds, on both sides. Finally, the total width of a tremor pulse in an SSE window is computed by considering the earliest starting point on the left side and the latest ending point on the right side. It must be noticed that, the derivation of the tremor duration depends on the window length. In fact, the inferred tremor duration can saturate to a maximum value equal to the length of the window. For this reason, we added in Figure 4.5(c) a dashed line corresponding to the window length (SSE duration + 14 days) (see section "Computation of local- and global-scale correlations").

4.4.7 Computation of local- and global-scale correlations

We compute the time-lagged cross-correlation between the SSE probability and the number of tremors per day (Fig. 4.5(a) and (b)). We smooth the number of tremors per day with a gaussian filter ($\sigma = 1.5$ days). We consider a lag between -7 and 7 days, with a 1-day stride.

In the case of Fig. 4.5(a), we compute the global correlation coefficient by considering the whole time sequence (2010-2022). As for Fig. 4.5(b), we make a local analysis. For each detected SSE, we first extract SSE and tremor slices from intervals centered on the SSE dates $[t_{SSE}^{start} - \Delta t, t_{SSE}^{end} + \Delta t]$, where $\Delta t = 30$ days. We first compute the cross-correlation between the two curves to filter out detected SSEs whose similarity with tremors is not statistically significant, *i.e.*, if their correlation coefficient is lower than 0.4. We build Fig. 4.5(b) after this process.

We compute Fig. 4.5(d) by comparing the SSE and tremor durations for all the events that had a cross-correlation higher than 0.4. For those, we infer the tremor duration, using the method explained in section "Calculation of tremor durations" on the daily tremor rate cut from an interval $[t_{SSE}^{start} - \Delta t', t_{SSE}^{end} + \Delta t']$, with $\Delta t' = 7$ days.

4.4.8 Overlap percentage calculation

In Figure 4.4 (b) we color-code the SSE durations by the overlap percentage between a pair of events, which we compute as the difference between the earliest end and the latest start, divided by the sum of the event lengths. Let E_1 and E_2 be two events with start and end dates given by (t_1^{start}, t_1^{end}) and (t_2^{start}, t_2^{end}) and with durations given by $D_1 = t_1^{end} - t_1^{start}$ and $D_2 = t_2^{end} - t_2^{start}$, respectively. We compute their overlap π as:

$$\pi = \frac{\max(0, \min(t_1^{end}, t_2^{end}) - \max(t_1^{start}, t_2^{start}))}{D_1 + D_2} \tag{4.11}$$

5 Characterization of slow slip events from GNSS data with deep learning

This chapter resumes the last part of the Ph.D. research, focused on slow slip event characterization. The first part (section 5.2) is based on a published IGARSS (*International Geoscience and Remote Sensing Symposium*) proceedings paper:

Costantino, G. and Giffard-Roisin, S. and Dalla Mura, M. and Socquet, A. (2023). *Characterization of slow slip events from GNSS data with deep learning*.

Section 5.3 is a recent and unpublished work at the date of the manuscript submission.

In this chapter, we discuss slow slip event characterization in GNSS data. In section 5.2, we follow the approach presented in chapter 3 to detect slow slip events contained in the middle of a sliding window. In section 5.3, we address the problem of SSE characterization by first denoising the GNSS time series. The latter approach gives the best results, with accurate performance on real GNSS data in the Cascadia subduction zone. Our method is capable of tracking and retrieving also small amounts of displacement, which correlate spatially and temporally with tremor propagation and can be used as the basis for SSE characterization. We manually select slow slip events to build a tentative SSE catalogue and we perform a static inversion of the source parameters associated with the displacement of each event, providing preliminary scaling laws for the detected events.

5.1 General overview

In this chapter, we discuss the problem of slow slip event characterization, that is the estimation of the SSE source parameters (at first order, magnitude estimation and localization). In chapter 3, we solve the problem of earthquake characterization in GNSS data by employing windows of GNSS time series centered on the co-seismic date. In section 5.2, we will discuss how slow slip event characterization can be made by following the same approach. We first address the problem with synthetic data. Then, we attempt to characterize them in real GNSS data, by relying on the events identified by SSEdetector.

The problem of SSE characterization can also be addressed from another perspective. Slow slip event (or earthquake) characterization is usually performed through an inversion of the source parameters. If we used ideal time series, *i.e.*, noiseless data, this inversion could be easily carried out. However, the presence of noise makes the problem way harder, introducing the necessity of pre-processing the data (*e.g.*, removing undesired signals such as common modes, seasonal signals, post-seismic relaxation signals, via trajectory models or independent-component-analysis-based methods) in order for the inversion to converge. Deep learning techniques can be used to generalize and make this technique more robust, as shown in the previous chapters, yet the noise still hinders the applicability of deep learning techniques as well. In the second part of this chapter we explore a complementary strategy: we aim at denoising the GNSS time series, such that any method can be used on the denoised time series afterwards, either "classical" or deep learning approaches. We will discuss GNSS time series denoising in section 5.3.

5.2 Characterization of slow slip events source parameters from raw GNSS time series with deep learning

This section is mostly based on our published IGARSS paper.

5.2.1 Introduction

The stress that accumulates on faults due to tectonic plate motion can be released seismically and aseismically. The aseismic part of this stress release occurs during

Slow Slip Events (SSEs), that last from days to years and do not radiate seismic waves. SSEs are monitored with dense GNSS networks that record the deformation induced at the surface of the Earth. Precise identification of slow slip events is key to better understanding the mechanics of active faults and to better describing the role of aseismic slip in the seismic cycle. Yet, the detection and characterization of SSEs of various sizes from existing GNSS networks is still a challenge, and extensive SSEs catalogs remain sparse and incomplete.

The difficulty in revealing low-magnitude events is related to their detection in the geodetic data, which contains large spatiotemporal perturbations and a low signal-to-noise ratio, making slow slip event detection hard to automatize. Traditionally employed methods either focus on specific events, identified visually with a high signal-to-noise ratio (e.g., (Radiguet et al., 2016)), use time series decomposition approaches such as Independent Component Analysis Inversion Methods (ICAIM) (e.g., (Michel, Gualandi, & Avouac, 2019b)); or, for small events, geodetic template matching (e.g., (Okada et al., 2022; Rousset et al., 2017b)). On the other hand, deep learning methods have been successfully applied in other geoscience applications such as earthquake detection and phase picking (Mousavi et al., 2020a; Zhu & Beroza, 2019), earthquake location and magnitude estimation from seismic data (Münchmeyer et al., 2021) and geodetic data (GNSS) (Costantino et al., 2023b), as well as application to slow earthquakes (Rouet-Leduc et al., 2019), volcano seismic signals (Lara et al., 2020; Malfante et al., 2018). Yet, no spatiotemporal methods have been developed to address the characterization of slow earthquakes using multi-station GNSS data. Thus, our work is mainly motivated by the necessity to explicitly account for spatial information, which can lead to better performance with respect to time-series-based methods (Costantino et al., 2023b).

Here we aim to develop and compare the performance of deep-learning-based methods for the characterization of SSEs in raw GNSS data with a multi-station approach. One way is to use 2D Convolutional Neural Networks (CNN), specifically developed for image-like data. With this method, two-dimensional convolutions (time and station dimension) can model both local- and large-scale spatial relationships between stations, but no explicit information coming from the spatial geometry of the GNSS network is enforced. To this end, spatiotemporal Graph Neural Network (STGNN) methods (Yu et al., 2017) can learn these relationships, with the potential of outperforming the 2D-CNN approach. However, developing such models is not trivial and requires setting up explicit joint processing of the information in the temporal and spatial domains. Classical graph-based approaches rely on message passing to propagate the information between neighbouring nodes. Time-varying features can be handled by using a Recurrent Neural Network (RNN) cell as an

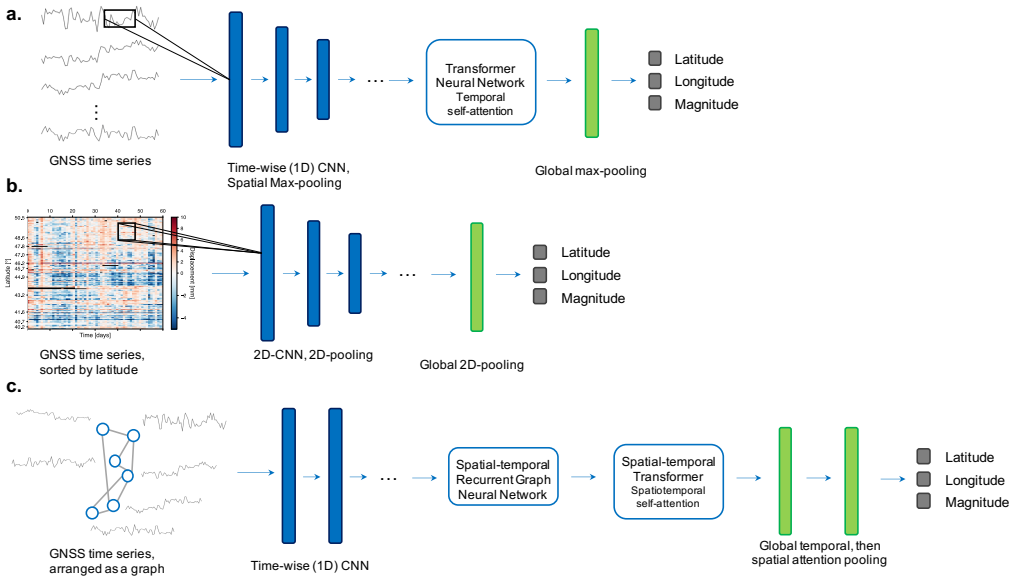


Fig. 5.1.: Schematic representation of the three developed deep learning models: (a) Time-series-attention-based Convolutional Neural Network for characterization (TSACNN), (b) matrix-based CNN (MCNN), (c) spatial-temporal attention-based graph-recurrent neural network (STAGRNN).

aggregation function, *e.g.*, long short-term memory (LSTM) or gated recurrent unit (GRU), but extending this mechanism to multivariate time series analysis is not straightforward, since GNSS measurements usually have more than one dimension (here, we use North-South and East-West components). Spatial-temporal approaches have thus been developed to jointly rely on multiple sensor measurements and multiple components. They can be classified in RNN-based and attention-based methods, and are capable of dealing with 3D data (here, [stations, time, directions]). The first family of approaches uses RNNs to extract temporal features (Bai et al., 2020), while the second focuses on attention mechanisms in the time and/or space dimension (L. Shi et al., 2019). Also, most of the spatial-temporal methods consider that the graph connectivity (adjacency matrix) is available beforehand. Most of the previous works compute the adjacency matrix based on distance metrics (*e.g.*, traffic forecasting) (Yu et al., 2017) or prior information on the nodes (*e.g.*, skeleton-based action recognition) (L. Shi et al., 2019). However, relying on pre-computed connectivity might not be the optimal choice, since additional edges could be learned from the data, and superfluous connections could also be removed. Moreover, graph connectivity is not always available, as in our case, and computing a satisfactory adjacency matrix is challenging since it would require connecting nodes within a homogeneous mesh as well as allowing for edges linking nodes that are far away depending on the data characteristics. Hence, we allow for the adjacency matrix

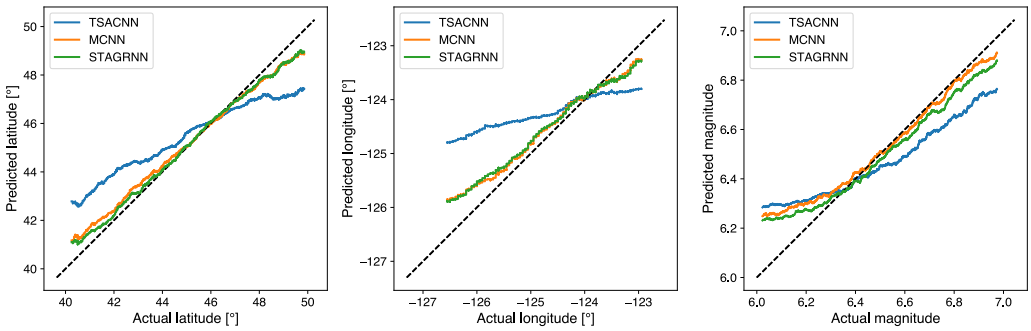


Fig. 5.2.: Comparison of actual and predicted values for latitude, longitude and magnitude. For each subplot, the curves represent the running average of the predicted variable over bins of 200 actual samples.

to be learned during the training, by following the same approach as (Bai et al., 2020).

5.2.2 Methods

We develop a deep learning method for SSE characterization, based on a graph-based Recurrent Neural Network with spatial-temporal attention, represented in Figure 5.1(c). We consider each GNSS station as a node in a graph and the edges between nodes as relationships between stations. We first extract time-dependent features by means of a 1D CNN made of three convolutional blocks (as for TSACNN) with filter sizes [3, 7, 11] and output feature maps [8, 16, 32]. The Adaptive Graph Convolutional Recurrent Network (AGCRN) (Bai et al., 2020) is used to learn the adjacency matrix (the graph's edges) as well as the temporal relationships in the GNSS data for each neighbouring node. A Transformer Neural Network is used to further extract spatial-temporal features with a stack of two self-attention modules (for both space and time embedding). Attention pooling is used first in the temporal and the spatial dimension and a linear layer is used for the output. We will refer to this model as spatial-temporal attention-based graph-recurrent neural network (STAGRNN).

We compare STAGRNN with two further deep-learning methods. The first one is based on time series, which we refer to as time-series-attention-based Convolutional Neural Network (TSACNN), as shown in Figure 5.1(a). This model has been readapted from a detection method previously developed by the authors (Costantino et al., 2023a) and is made of 1D convolutional layers (1x5 kernel) equipped with batch normalization and ReLu activation and interleaved with max-pooling operations (3x1 kernel) in the station dimension. We group three convolutional layers as

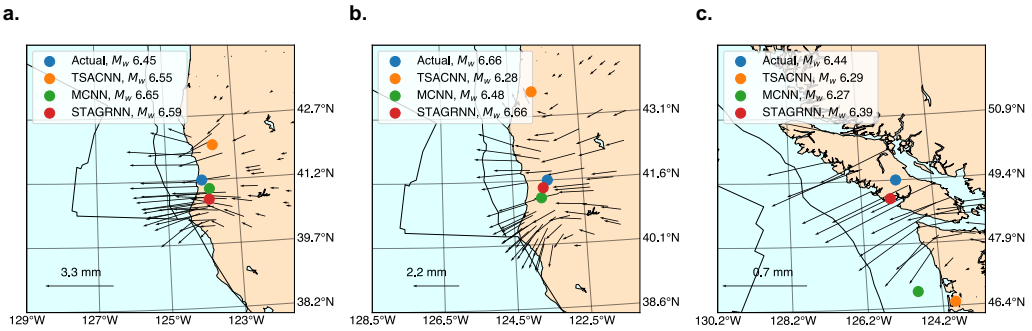


Fig. 5.3.: Example of the predictions of TSACNN, MCNN and STAGRNN. For each subfigure, arrows show the static displacement field associated with the tested SSE. Points represent the location of the tested and predicted SSEs (see legend). The actual and predicted magnitude for each model is shown in the legend of each subplot.

a convolutional block and we insert a max-pooling after each block. The number of feature maps computed for each block is [8, 32, 128, 256]. A max-reduce operation is performed at the end of the convolutional encoder to eliminate the station dimension. A Transformer, equipped with a time-wise learnable positional embedding is used to apply temporal attention. A 1D global average pooling is used afterward in the temporal dimension and a linear layer with three regression outputs (linear activation). The second method is a 2D-CNN working on GNSS data arranged as a matrix (stations $\times$ time), where each row of the matrix represents a GNSS time series for a given station, as shown in Figure 5.1(b). The stations are sorted by latitude. We use the MobileNetV2 model as the backbone of the CNN encoder. A 2D global average pooling is used after the 2D-CNN and a linear output layer is used as before. We refer to it as matrix-based CNN (MCNN).

5.2.3 Results

Tab. 5.1.: Position and magnitude error (mean $\pm$ standard deviation) on the synthetic test set.

Model	Position error (km)	Magnitude error
TSACNN	159.53 ± 144.36	0.15 ± 0.12
MCNN	78.78 ± 134.07	0.12 ± 0.07
STAGRNN	$\mathbf{76.07 \pm 113.49}$	$\mathbf{0.10 \pm 0.10}$

We train and test our models on synthetic data, because of the paucity of catalogued slow slip events. We choose the Cascadia subduction zone as the target region since it is a well-studied tectonic context and also because complementary data (such as

catalogues or tremor recordings) can be used as an independent validation. We build a synthetic database by generating randomly distributed SSEs along the Cascadia subduction following the approach already developed by the authors (Costantino et al., 2023a; Costantino et al., 2023b). The resulting training and evaluation sets are made of 60,000 synthetic GNSS time series obtained by adding synthetic slow slip time series (modeled as sigmoidal-shaped transients) to realistic noise time series computed as in Costantino et al., 2023a. We split the synthetic data set into 80% training set, 10% validation set and 10% test set. We train the four models by using the ADAM optimizer with a learning rate of 0.001 and mean squared error as the loss function.

A summary of the performance of the methods is reported in Table 5.1 and Figures 5.2 and 5.3. We compute the position error as the geodesic distance between the actual and predicted epicenter, and the magnitude error as the mean absolute error between the actual and predicted magnitude. TSACNN shows a fair performance in the magnitude estimation, suggesting that the magnitude information can be well recovered from time series. However, MCNN and STAGRNN seem to better constrain the spatial extent of the deformation, possibly improving the magnitude prediction. There are minor differences between MCNN and STAGRNN in terms of mean prediction: the STAGRNN method performs better than MCNN in latitude and low-magnitude estimation, but has lower precision at higher magnitudes. However, thanks to the graph-based approach, the STAGRNN method has the potential to be more flexible on real data, to better deal with data gaps and a variable set of stations. Also, it could be more easily transferred to different regions, which makes it more promising in the direction of testing on real data.

5.3 Denoising GNSS time series with deep learning: towards spatiotemporal slow slip characterization

5.3.1 Motivations

When dealing with GNSS time series analysis, scientists often pre-process the data in order to mitigate the noise contribution. When looking for SSEs, GNSS noise can be seen as a combination of several signals such as (1) common mode errors, due to perturbations of the GPS reference frame, likely associated with orbit and positioning errors (Wdowinski et al., 1997; Williams et al., 2004), (2) seasonal signals, mainly due to seasonal changes in atmospheric conditions, hydrological

effects, tides and snowpacks, (3) tectonic effects such as co-seismic signals, due to earthquakes, or post-seismic signals, associated with the ground relaxation following earthquake occurrence. Most studies use trajectory models (Bedford & Bevis, 2018; Marill et al., 2021) to subtract the contribution linked to seasonal variations or post-seismic relaxation in the GNSS time series. Yet, this can, in some cases, fail to constrain the noise. For example, seasonal variations can have different amplitudes in the whole time series and a signal shape that is not necessarily fully reproduced by a simple sum of sine and cosine functions. Moreover, the time series can be poorly fitted (or overfitted) by the trajectory model, leading to a biased estimate of the tectonic signals, notably of the displacement associated with slow slip events. This approach can thus introduce some spurious transient signals which could be erroneously modeled as slow slip events. Another solution would be to use independent-component-analysis-based methods (Michel, Gualandi, & Avouac, 2019a) to extract the seasonal variations. However, the extracted components might contain some signals of interest as well as some noise characteristics which could not be removed (e.g., specific harmonics or spatiotemporal patterns). Hence, the main motivation driving the approach used in this Ph.D. thesis is that we do not want to model any GNSS-constitutive signals, but use the GNSS time series as they are. Here, we want to investigate whether slow slip event characterization can be improved by first denoising GNSS time series as an intermediate step. The idea is that denoising can be seen as a more general procedure, which could isolate the aforementioned noise components and remove them more effectively. While recent works have shown promising results in denoising single GNSS time series in high-rate GNSS data (Thomas et al., 2023), we believe that jointly denoising all the GNSS time series would considerably improve the results as it would add a crucial spatial constrain. Deep learning models trained to solve this task can extract and leverage features that are general enough to approximate the joint influence of the single noise components.

Here, we rely on a part of the architecture of the spatial-temporal attention-based graph-recurrent neural network (STAGRNN) method (cf. section 5.2) developed to address the problem of SSE characterization and we develop SSEdenoiser, a graph-based attentive deep learning model trained to uncover slow deformation patterns in GNSS position time series by denoising them. We train SSEdenoiser on a refined version of the synthetic database introduced in chapter 4 (see sections 4.2.1, 4.4.2 and 4.4.3), output by SSEgeneratorV2, an improved version of SSEgenerator ((Costantino et al., 2023a)), targeted to generate more complex and realistic synthetic time series (see Figure 5.4).

5.3.2 SSEgeneratorV2: the training database

In chapter 4, we introduced SSEgenerator and described the main idea behind our synthetic spatiotemporal signal generation. However, albeit suited for first-order SSE detection, some limitations need to be solved to successfully address SSE denoising. First, we assume that the SSE does not necessarily occur in the middle of the temporal window considered, but that the SSE date is a uniform random variable drawn from 0 to 60 days (we use the same window length as in chapter 4). By allowing for the date to span the whole data window, we can expect the model to detect transients of slow deformation at any date. This way, when testing on real data, the SSE-driven displacement can be tracked at any time and not only when the event is positioned in the middle of the sliding window.

In the previous version of SSEgenerator, only one synthetic dislocation per sample was allowed. Although we proved in chapter 4 that our model can generalize well for the detection task, this would limit the ability of the deep learning model to denoise full-length time windows. In particular, we want to generalize over complex scenarios, including propagating slow slip events or slow slip events with a variable amount of slip. Here, we impose, for each window, a variable number of synthetic dislocations per window between 0 and 3, with independent source parameters[1]. The final database contains 25% of each setting (0 dislocations, 1 dislocation, 2 dislocations, 3 dislocations). The combination of variable number of dislocations and variable SSE initiation times makes the deep learning model suited to detect slow slip events that last longer than the training ones with higher precision with respect to SSEdetector.

Moreover, as for SSEdetector, we allow for the presence of *negative* samples (no synthetic dislocations in 25% of the samples). When addressing denoising, this is key for learning the noise structure at best, since the model is taught to output a zero-displacement, thus focusing on extracting an enhanced representation of the input noise. This can be used to better denoise the *positive* samples.

Finally, we also improve the synthetic noise generation. In SSEgenerator, the period 2014-2017 was not considered when computing the realistic noise, in order to leave it aside to test against potential overfitting. In section 4.2.4, we demonstrate that

[1] As we do in chapter 4, we generate synthetic dislocations (Okada, 1985) with randomly chosen locations along the Cascadia subduction zone. We uniformly generate the SSE zones within a band along the subduction between 20 and 40 km depth, with ± 10 km variability, in order not to rely on the specific slab model and thus to better generalize over real data. We generate dislocations with a fixed aspect ratio (0.5), with magnitudes uniformly drawn between 6 and 7 and stress drop modeled from published scaling laws (H. Gao et al., 2012). The SSE signals are assumed as sigmoids and can have a duration between 10 and 30 days.

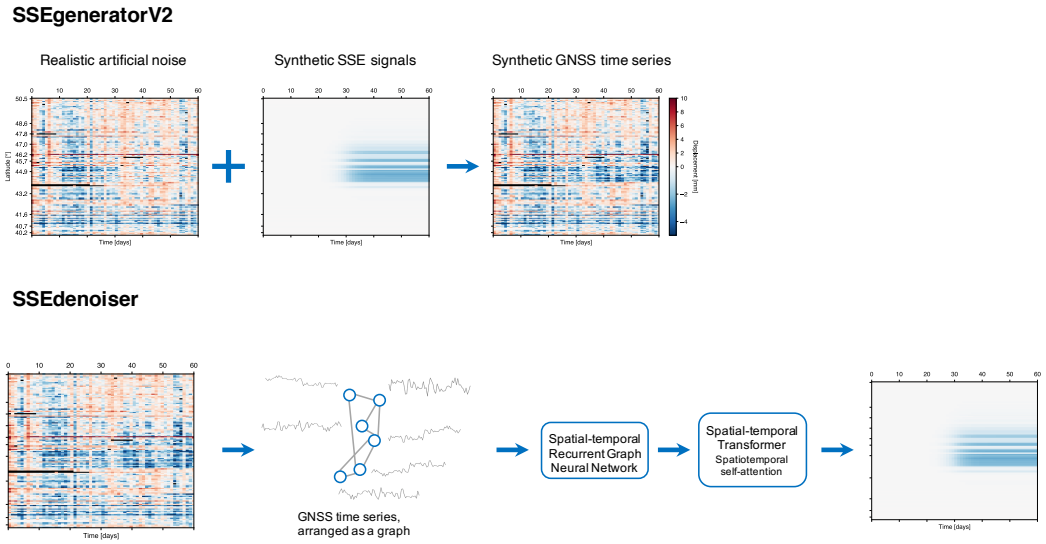


Fig. 5.4.: Overview of the architecture of SSEdenoiser.

this procedure does not produce any overfitting on real data, hence here we generate synthetic noise using time series from 2007 to 2022 in order to have longer time series, and thus a more robust PCA and spectrum estimate.

5.3.3 SSEdenoiser: architecture

The architecture of SSEdenoiser is mostly adapted from the STAGRNN model presented in section 5.2 and is shown in Figure 5.4. The two core components are the Adaptive Graph Convolutional Recurrent Network (AGCRN) (Bai et al., 2020) followed by a spatiotemporal Transformer neural network. We removed the CNN component prior to the AGCRN since it was associated with lower accuracy on the validation data.

5.3.4 Results on synthetic data: learned station connectivity

Our multi-station approach unlocks concrete access to spatial and temporal relationships between GNSS stations. During training, SSEdenoiser learns what stations (vertices of the graph) should be connected and how strong their connections should be: this information is synthesized in the adjacency matrix, which gives the edge strength of the graph network. In Figure 5.5, we show the learned adjacency matrix, with nodes sorted by latitudes (each pixel represents a relationship between two stations). The matrix is symmetric, meaning that the learned graph

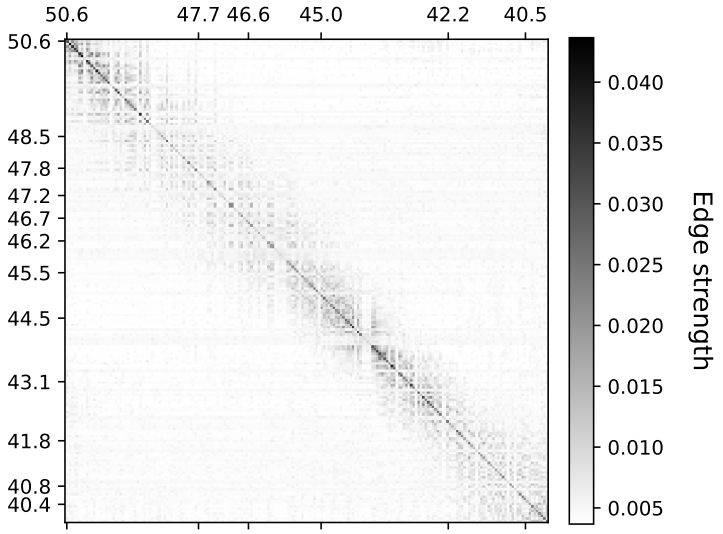


Fig. 5.5.: Graph adjacency matrix learned during training, obtained from the STAGRNN module. Nodes (stations) are sorted by latitude.

structure is undirected (see section 2.6.4.1). The number of edges in the graph is $N + \frac{N(N-1)}{2} = 20100$ including self-loops[2]. From Figure 5.5, we can see that SSEdenoiser has learned to connect nodes that are mostly spatially close to each other (near the diagonal). Yet it also allows for long-range connections, which are weaker (*e.g.*, they are assigned a lower edge strength). This suggests that the method could generally rely on information available within a neighborhood and then compare information coming from different sub-networks.

In Figure 5.7(a), we show a qualitative histogram of the edge strengths. We compute the edge strength histogram of all stations separately and we combine all the histograms together. This is not a proper edge strength histogram, but it gives information on potential patterns of the edge distribution. We see a partitioning of edge strengths into four main clusters. The first corresponds to edge strengths less than 0.008, which can be thought of as weak background connections that allow for a minimum amount of information flow. There is a total of 19026 weak connections.We find 1074 connections of strength higher than 0.008 (194 of which are self-loops). They can be grouped in three clusters centered around 0.01, 0.014, and > 0.016, corresponding to stronger connections.

In order to visualize the backbone of the graph structure, we further filter the adjacency matrix by selecting edge strength values in the interval (0.008, 0.0234). We found 878 connections (excluding self-connections on the diagonal) to which we will refer in the following as “strong connections”. We show this filtered adjacency

[2]This is equivalent to a clique, that is a graph where each node is connected to every node

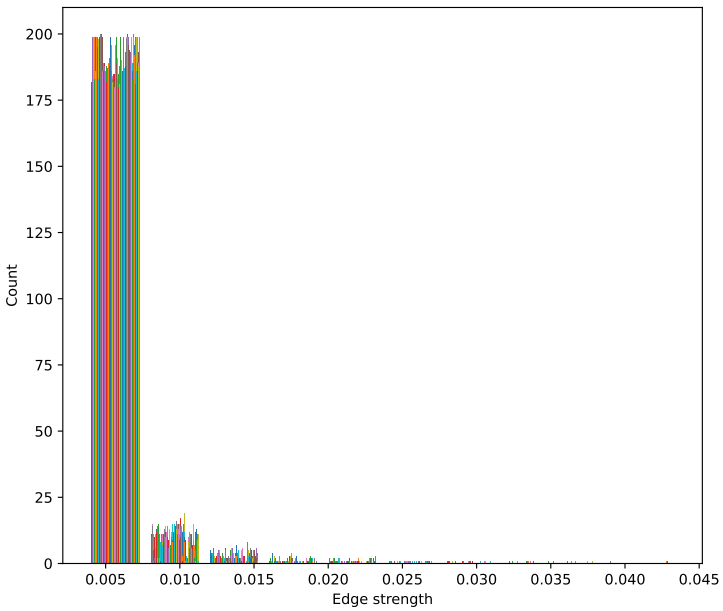


Fig. 5.6.: Qualitative histogram of edge strength. The histogram is obtained by grouping the histograms computed on each row separately, *e.g.*, each node with respect to its neighbouring connections.

matrix in Figure 5.7 (a), and we also represent their spatial distribution in terms of connections between GNSS stations in Figure 5.7 (b). We first see that the connections between stations are such that the azimuthal coverage is as high as possible between neighbouring stations. Also, the method has learned how to produce a mesh connecting all the stations that are located on top of the slow slip area used in the training phase (between 20 and 40 km depth). Stations located further inland (longitude $< 122°$) do not have a strong edge weight, suggesting that they are not informative for slow slip detection given that the synthetic slow slip events in the training database are located in a band between 20 and 40 km depth. Also, stronger edges connect stations that are more sparsely distributed in the SSE area. In fact, when the network is dense in SSE areas, there is no need for such a high weight because the information is already included in the signal coming from many nearby stations. This is interesting because the vertices locations (stations) are not given to the model: it has inferred their relevance from the data and learned to connect stations that are distant to each other when the density is not sufficient for effective information flow, possibly thanks to the spatiotemporal transformer module.

Figure 5.8 shows the GNSS stations used in this work, color-coded by the value of the adjacency matrix diagonal. This indicates the strength of the self-loop connections for each station, which can be thought of as a measure of the learned station's

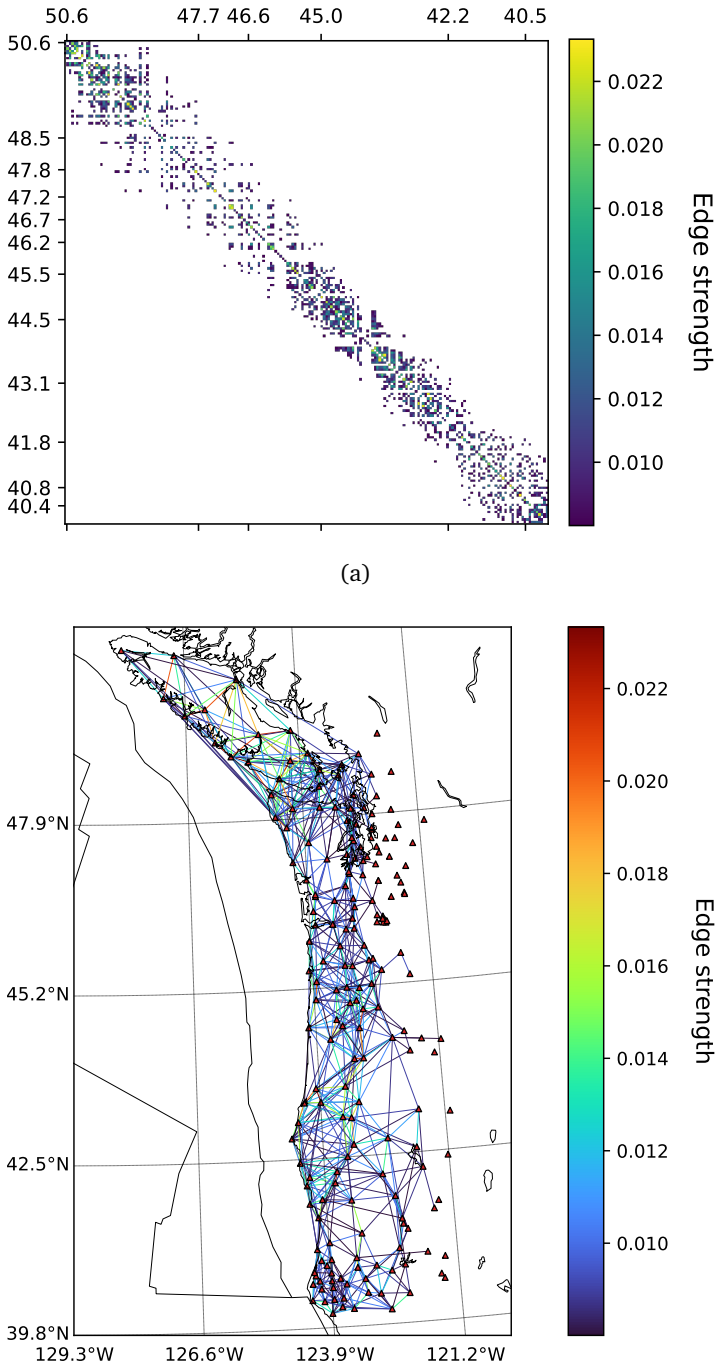


Fig. 5.7.: Strong connections learned by SSEdenoiser. (a) Graph adjacency matrix filtered by retaining node connections with edge strength between 0.008 and 0.0234. Nodes (stations) are sorted by latitude. (b) Map view of the graph obtained by selecting the strongest connections (cf. (a)).

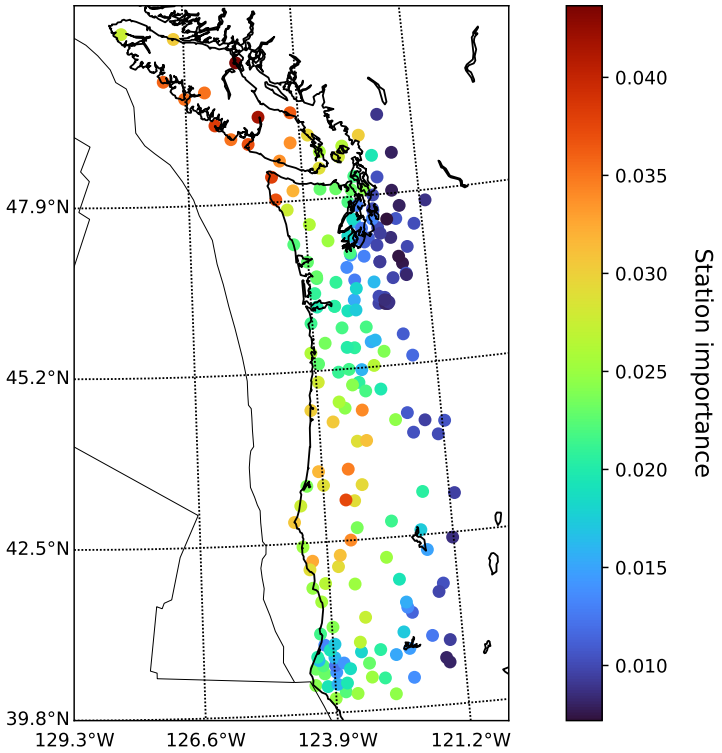


Fig. 5.8.: Importance (self-loop strength) of the GNSS stations learned from the data, contained in the diagonal of the adjacency matrix.

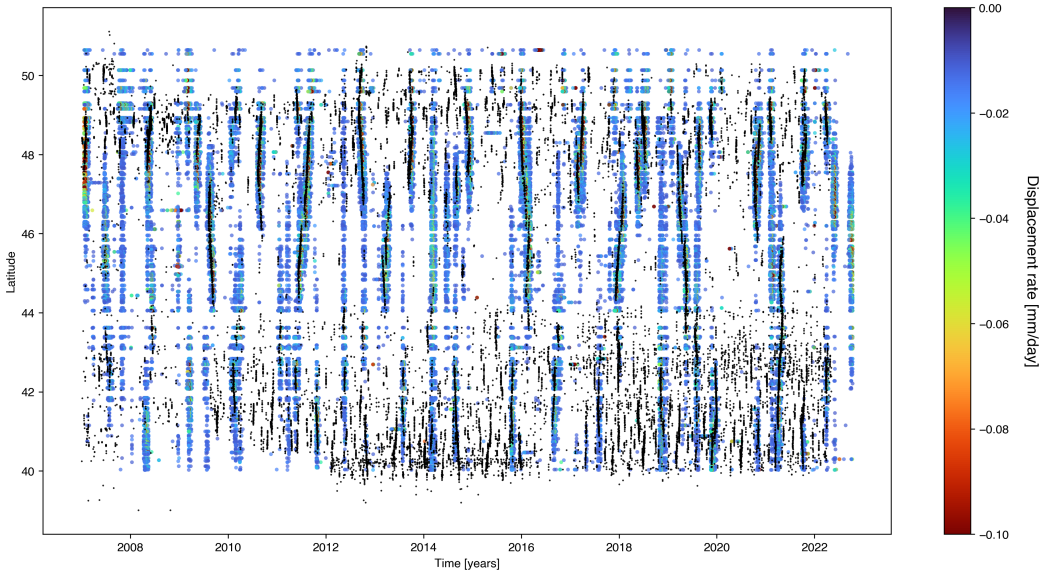
relevance (or importance). We can see that the importance of further inland stations is low compared to stations that are located in correspondence with the SSE area (band between 20 and 40 km depth). The highest values of importance are assigned to stations located in Vancouver Island and at latitudes between 42.5° and 45.2°, probably linked to a learned trade-off between slow slip occurrence and availability of GNSS stations. This is a proxy of the distance to the SSE source area combined with the local density of stations.

5.3.5 Denoising of real GNSS time series in Cascadia in 2007-2022

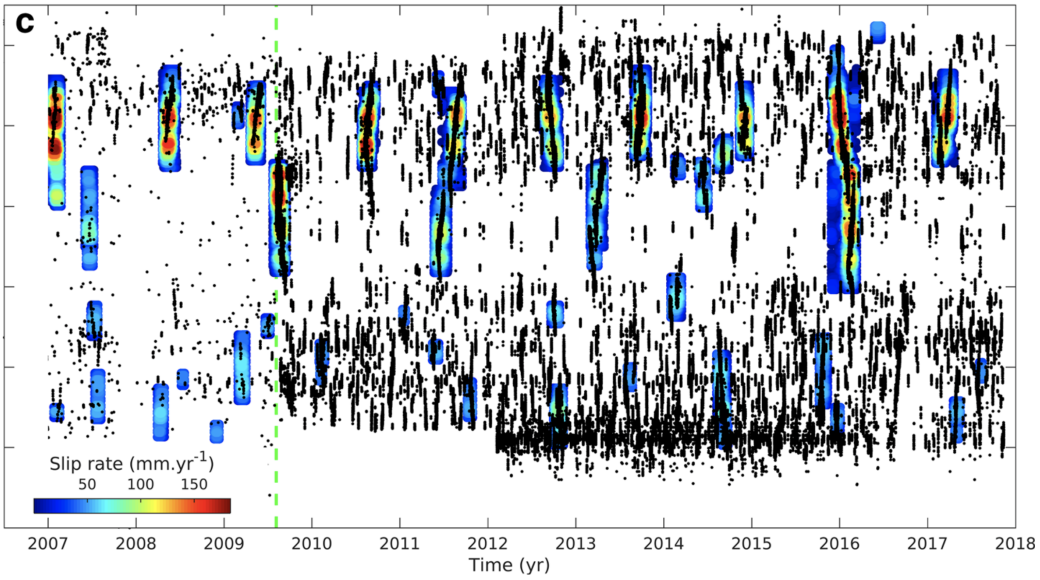
We test SSEdenoiser on real non-post-processed GNSS time series from 2007 to 2022. We take 60-day windows of data and apply SSEdenoiser on each of them, by sliding the windows with a stride of 1 day and collecting the resulting denoised GNSS time series[3]. We compute the temporal derivative of the data in each window to obtain the displacement rate and we keep the 20 days in the middle of the window to exclude potential border effects. We sum all these 20-day overlapping windows together in time (each day sample corresponds to the sum of all the windows that slide at that date). Then, we divide each sample by the number of windows that were summed together. With this procedure, we obtain the daily average displacement rate.

Figure 5.9 (a) shows the obtained denoised GNSS time series of daily (E-W) displacement rate (only displacement rates larger than 0.01 mm/day are shown) at all stations in matrix form, over the period 2007-2022. In Figure 5.9 (b) we show the results from Michel, Gualandi, and Avouac, 2019a to ease the comparison. We see that the retrieved displacement rate has a coherent spatiotemporal distribution. It occurs by bursts that are clustered in latitude and time. The largest slow slip events are associated with large displacement rates (see Figures 5.10, 5.11, 5.12, 5.13), such as the 2011.5 or the 2016 slow slip events. The retrieved surface displacements are a proxy of the slip on the subduction interface. This lets us compare our detections with Michel et al.'s. We find 36 potential events in 2007-2017, compared to the 40 Michel et al.'s events. In the same period, we find 17 additional events, 5 of which are small and might be associated with a lower detection threshold. In 2018-2022, not analyzed by Michel et al., we find 24 events (see also Figure 5.13). However, we manually select these events have been manually chosen, thus they have to be further analyzed and identified with more objective methods.

[3] In practice, non-overlapping sliding windows can be used (stride of 60 days). Here, we prefer relying on multiple denoising outputs (stride of 1 day) to provide a better estimate of the displacement by averaging the contributions coming from all the possible sliding windows.



(a)



(b)

Fig. 5.9.: (a) Denoised displacement rate (E-W component) as a function of time. The displacement rate computed from the output of SSEdenoiser is shown for each GNSS station, sorted as a function of the latitude. Tremor recordings are also shown. We use the catalogue from Ide, 2012 until August 5, 2009, and the tremor catalogue from PNSN afterwards. (b) Occurrence of slow slip events as a function of time, from Michel, Gualandi, and Avouac, 2019a, between 2007 and 2017.

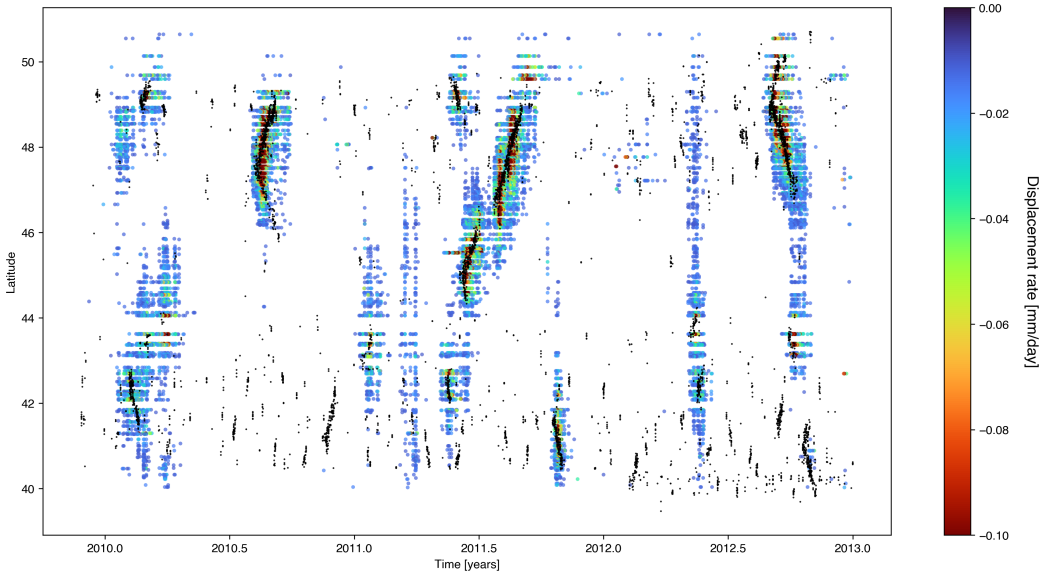


Fig. 5.10.: Zoom of Figure 5.9 (a) in 2010-2013. A clear correlation between tremor occurrence and a larger displacement rate is visible. Migrating events are also successfully retrieved.

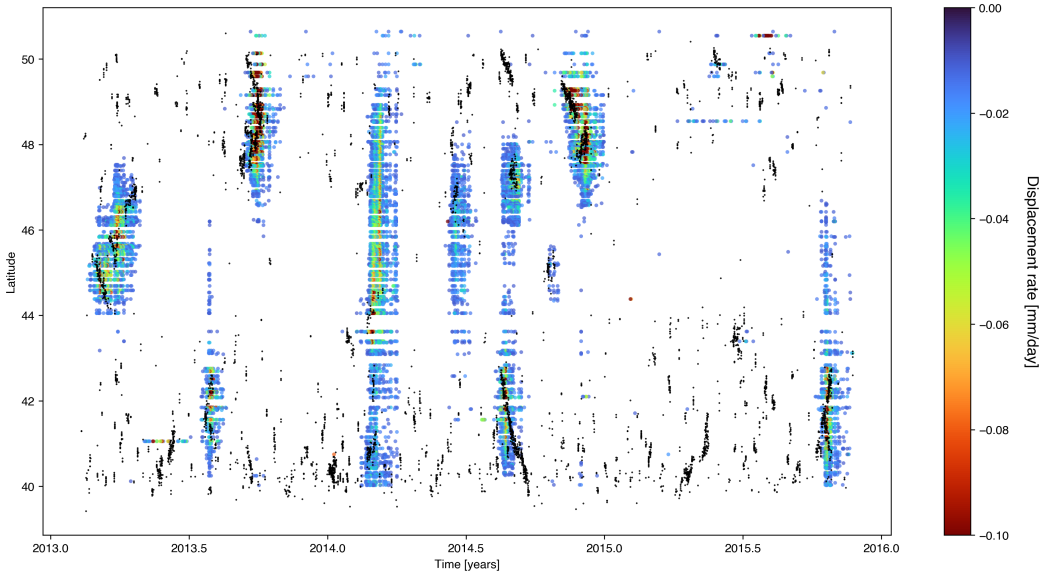


Fig. 5.11.: Zoom of Figure 5.9 (a) in 2013-2016. Again, a nice correlation with tremor occurrence is visible. In 2014 all stations are activated, probably due to a transient increased noise in the times series.

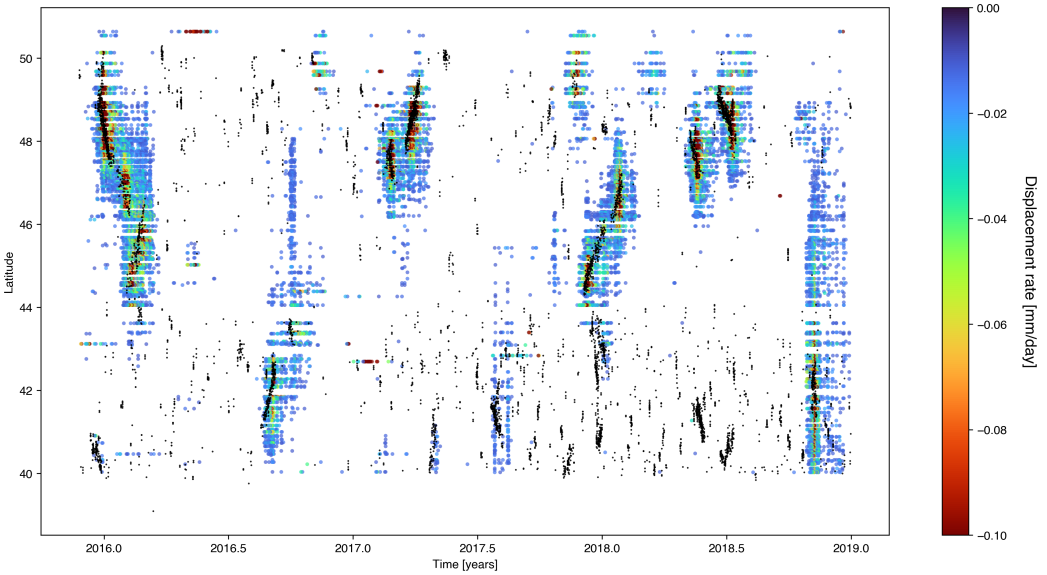


Fig. 5.12.: Zoom of Figure 5.9 (a) in 2016-2018. Again, a nice correlation with tremor occurrence is visible, migrating or complex events seem correctly retrieved. Some episodes of increased large-scale noise (unmodeled common mode?) generate coherent displacement rates over the whole network.

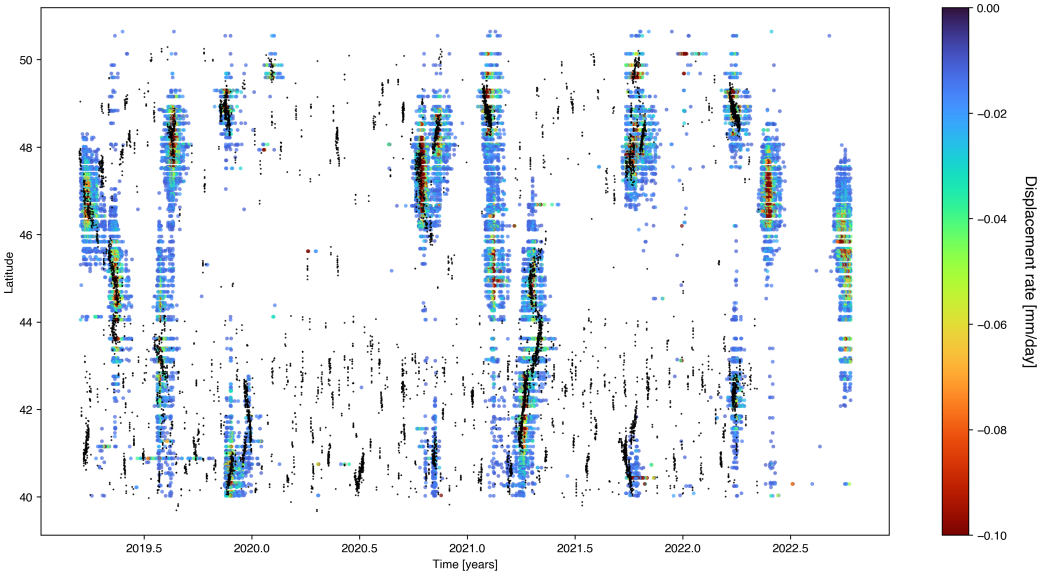


Fig. 5.13.: Zoom of Figure 5.9 (a) in 2019-2022.

Also, we found that SSEdenoiser is also able to constrain slow slip occurring in south Cascadia, which is more difficult than the northern area because of the higher noise level. We can see that the denoised displacement rates have a good correlation with the spatiotemporal distribution of tremors (shown in black in the figures). As in the case of SSEdetector (see chapter 4), SSEdenoiser is blindly trained on GNSS time series, without incorporating any information from tremors in the model: this means that our method achieves reliable results as tremors are expected to be correlated with SSEs and associated displacements. The displacement distribution follows the tremor propagation in space and time, both for large and smaller slow slip events, and for complex events as well, suggesting that SSEdenoiser has effectively learned what is the noise structure to retrieve concealed slow deformation at any scale. This also suggests that the retrieved sub-millimeter-scale displacement rate is not randomly distributed in space and time, but has a pattern similar to larger events, thus it might be associated with actual slow deformation which was undetectable so far. This needs to be further validated by comparing all the detections with the existing catalogs such as Michel, Gualandi, and Avouac, 2019b, but it is likely that our deep-learning method could detect slow slip events that were, so far, below the detection threshold. In fact, by learning hidden noise patterns, SSEdenoiser can effectively lower the detection threshold on geodetic data, providing new perspectives for the understanding of slow slip events.

We compute the cumulative displacement due to slow slip at each GNSS station, by calculating the integral in the time domain over the whole 2007-2022 period, as shown in Figure 5.14. We find that the cumulative surface displacement retrieved by SSEdenoiser is consistent with the cumulative slip at the interface found by independent studies (Bartlow, 2020; Michel, Gualandi, & Avouac, 2019a, 2019b).

In Figure 5.15, we focus on the large slow slip event that occurred from December to March 2016 to analyze the spatiotemporal surface displacement evolution compared with tremor episodes (visible at the beginning of Figure 5.12). We analyze snapshots of 9 days from December 17th to March 16th. We associate each subplot to a 9-day time span and, for each subplot, we show the tremors occurring in that time span as well as the 9-day cumulative displacement in the same time span. We find that SSEdenoiser can track the daily SSE propagation in space and time by precisely identifying the three major SSE propagations (one for each row), as also shown in Figure 5.12. The slow slip event started in the north and propagated southwards until meeting another rupture that started southwards and propagates northwards. SSEdenoiser is able to track the spatiotemporal slip evolution by retrieving sub-millimeter displacement in the raw GNSS time series. Since both

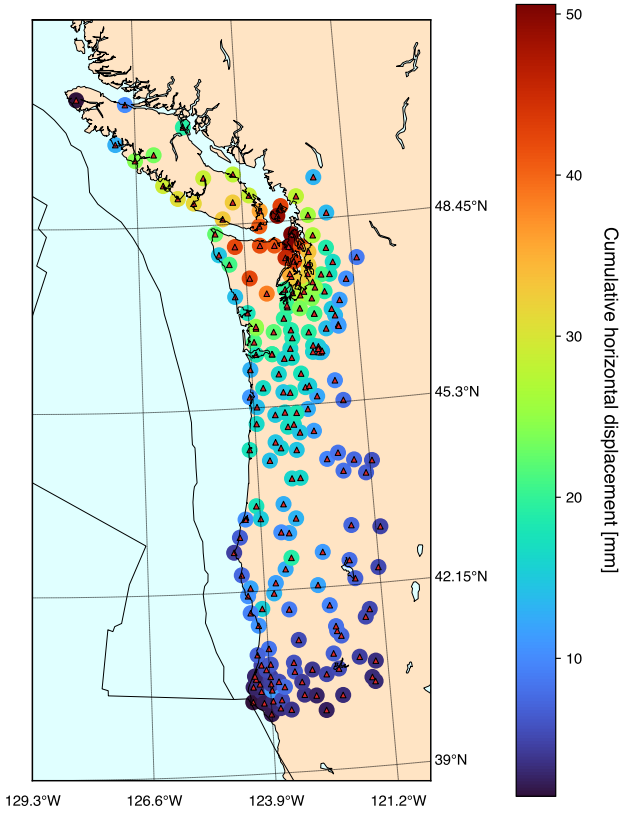


Fig. 5.14.: Cumulative horizontal displacement over 2007-2022 at the GNSS stations used in this study.

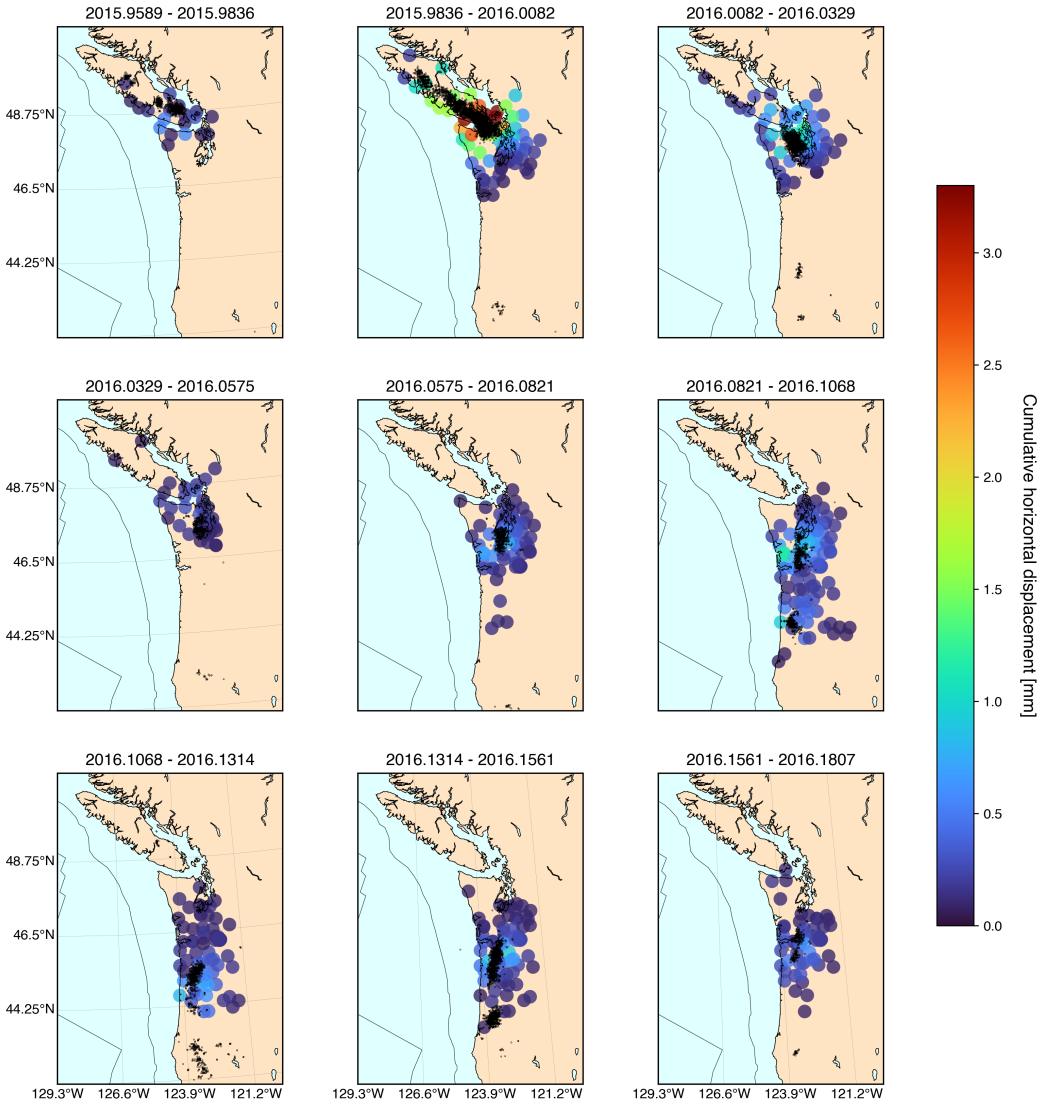


Fig. 5.15.: Spatiotemporal evolution of the displacement rate associated with the 2016 slow slip event. Each subplot shows a 9-day time period (indicated in the subplot title). For each snapshot, the cumulative horizontal displacement computed in the 9-day time period is shown at GNSS stations (only stations where a displacement larger than 0.08 mm is shown). Tremors are shown together with the surface displacement for each snapshot.

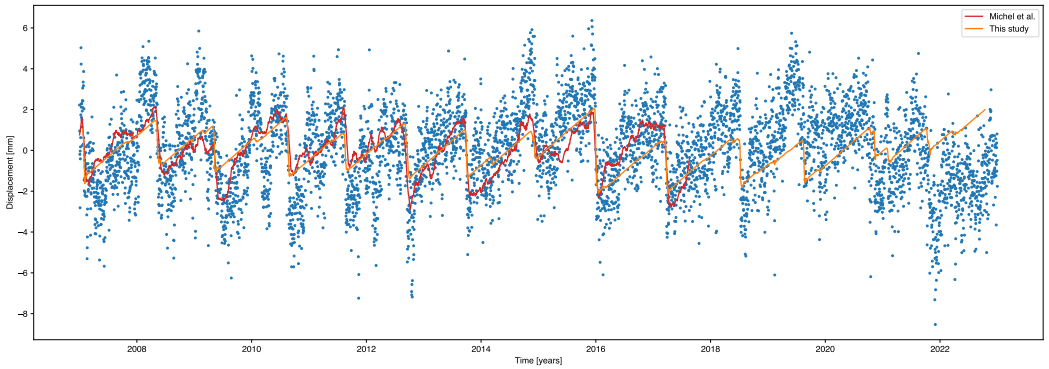


Fig. 5.16.: Example of denoised time series (ALBH station). Blue points represent the detrended GNSS time series for the ALBH stations in 2007-2022. The orange curve represents the denoised time series output by SSEdenoiser. The red curve shows the time series predicted by the model from Michel, Gualandi, and Avouac, 2019a, processed with a running average filter of 30 days.

latitude and longitude are well retrieved, it indicates that our spatiotemporal graph representation is particularly relevant to this problem.

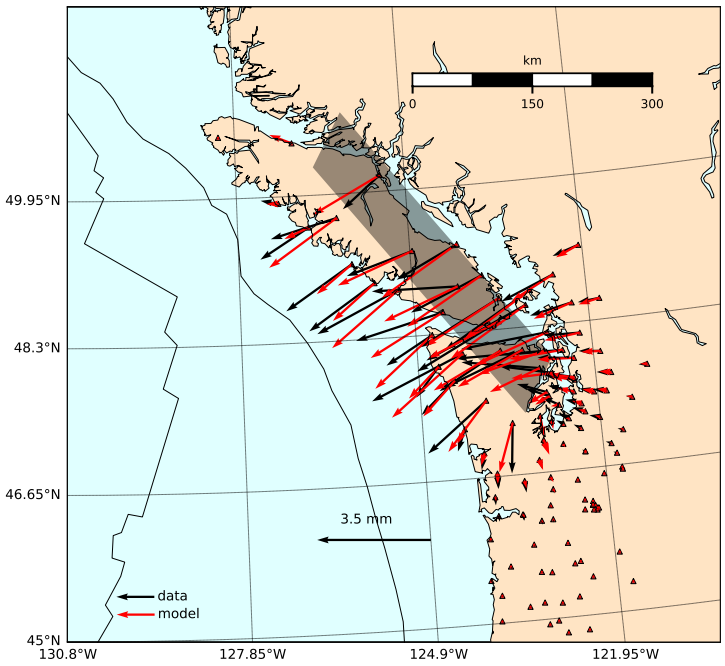
We further compare the detection power of SSEdenoising and its ability to disclose slow deformation below the GNSS detection threshold. We select the ALBH station (located on Vancouver Island) and we compute the integral of the displacement rate time series to obtain a time series showing the (detrended and denoised) displacement induced by slow slip events. We show it in Figure 5.16 (orange curve), while the blue dots represent the raw (detrended) data. We compare it with the time series predicted by the independent-component-analysis-based model from Michel, Gualandi, and Avouac, 2019a (red curve, only available between 2007 and 2017), for which we plot the running average computed on 30 days. The retrieved displacement does not seem to suffer from biases such as common modes or fortnightly signals (with an 8-day or 14-day period) (Abraha et al., 2017), which affect PPP (precise point positioning)-derived GNSS time series (used in this work). The displacement time series obtained by SSEdenoiser is, at first order, consistent with Michel et al.'s and accurately reproduces the data variability: we can see that all the curves align well both in terms of SSE duration and amplitude at first order. Also, it has the potential to better deal with several noise components which may affect Michel et al.'s model. Yet, this needs further validation. For the events that occurred in 2009.5, 2011.5, 2013, and 2017, the displacement is not well estimated with respect to Michel et al.'s model.We provide further analyses in the following sections. Also, the inter-SSE trend seems to be underestimated and a change in the trend is observed after 2019. This should be a point of attention. However, the trend of the denoised time series cannot be compared with the real trend, since

it is obtained by computing the cumulative sum over the displacement rate time series. Thus, the loading mechanisms of the real and denoised time series are not comparable.

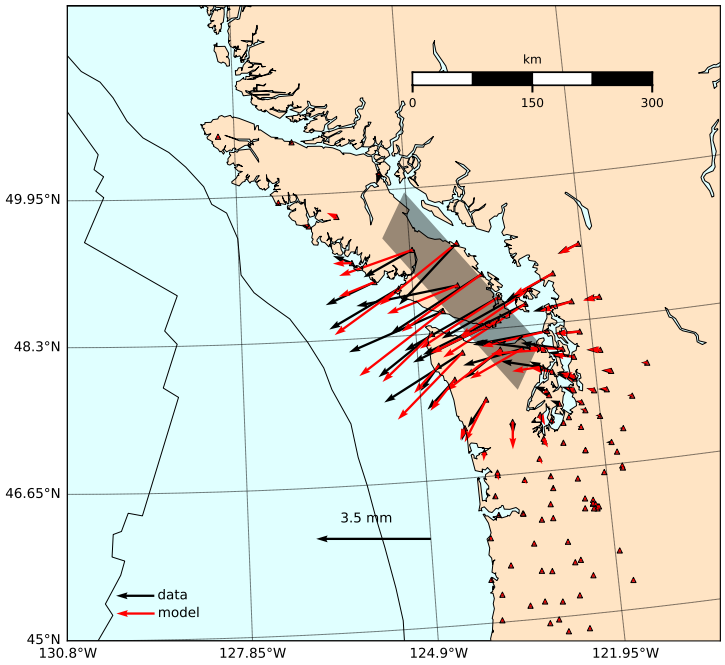
5.3.6 Static inversion of denoised displacements to retrieve simple source parameters for the 2013 and 2016 SSEs

Here we show an example of slow slip event characterization carried out from the denoised displacement output by SSEdenoiser. We use static inversion of the Okada dislocation model (Okada, 1985) to retrieve the source parameters of two well-known SSE: the 2013 (Bletery & Nocquet, 2020) and the 2017 (Itoh et al., 2022a) Cascadia slow slip events. This is a proof-of-concept and a first step towards a systematic characterization of all the potential slow slip events found by SSEdenoiser. We model the surface displacement as associated with a rectangular dislocation. We invert for the dislocation centroid position, depth, dislocation length and width, average slip, and focal mechanism. We force the depth, strike, and dip to follow the subduction interface, by using the values of the slab2 model at the location of the inverted dislocation centroid (Hayes et al., 2018). We show the static inversion results in Figure 5.17. For the 2013 event (Figure 5.17 (a)), we obtain the following parameters: centroid position 49.1°N 124.5°W, depth of 39 km, fault length and width of 489 km and 87 km, respectively, average slip on the interface 11.5 mm, and strike, dip, and rake angles of 311°, 10°, and 80°, respectively. For the 2017 event (Figure 5.17 (b)), we obtain a centroid position of 48.7°N 124°W, depth of 39 km, fault length and width of 304 km and 71 km, respectively, average slip on the interface of 14.5 mm, and strike, dip, and rake angles of 311°, 10°, and 80°. We remark, for both events, a good fit between the denoised data and the displacement modeled with the best parameters.

We compute the moment for both events as $M_0 = \mu\ A\ \bar{u}$, where μ is the shear modulus, assumed equal to 30 GPa, A is the fault area, computed as $L \cdot W$ and $\bar{u}$ is the average slip on the interface. Then, we compute the magnitude as $M_w = \frac{2}{3}\log_{10}(M_0 - 9.1)$ (Hanks & Kanamori, 1979). We obtain a magnitude estimate of 6.7 and 6.58 for the 2013 and 2017 events, respectively. Both magnitude estimates are coherent with the results obtained by Bletery and Nocquet, 2020 (M_w 6.6) and Itoh et al., 2022a (M_w 6.5). These results show that the denoised time series obtained from SSEdenoiser do not underestimate the static SSE slip value and moment, suggesting that the example provided in Figure 5.16 might reproduce the correct slow-slip displacement amplitude. However, further analyses have to be done to



(a)



(b)

Fig. 5.17.: Results of the static inversion from denoised GNSS data. Panel (a) and (b) show the results for the events that occurred in 2013 and 2017, respectively. For both events, the best-fitting dislocation on the subduction interface is plotted as well as the data (black arrows) and the Okada Okada, 1985 model predictions associated with the best parameters resulting from the inversion (red arrows).

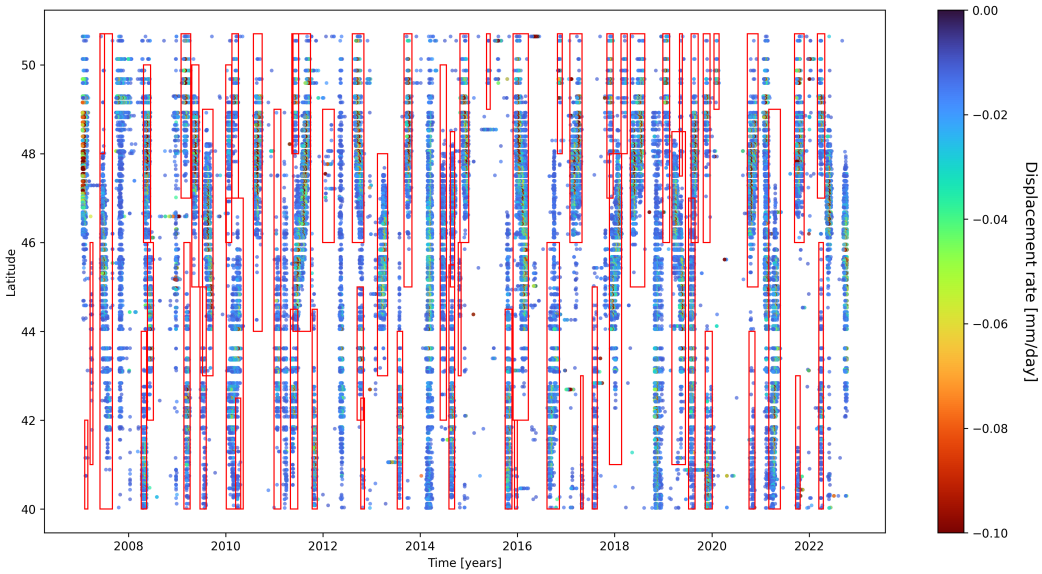


Fig. 5.18.: Manual selection of slow slip events from denoised GNSS time series. The denoised displacement rate output by SSEdenoiser is shown by colored points. Rectangles are associated with a manual picking of slow slip events based on spatial and temporal distribution of the surface displacement rate.

assess whether the denoised time series reproduce the correct displacement at all magnitude scales.

5.3.7 Characterization of all identified slow slip events in Cascadia and preliminary scaling laws

We manually select slow slip events based on their distribution in latitude and time (see Figure 5.9 (a)) to obtain a tentative catalogue of SSE dates, as we show in Figure 5.18. We select the SSE dates and zones based on the stations that recorded a displacement rate higher than 0.01 mm/day that is coherent with tremor occurrence. We exclude time periods where the displacement is contaminated by noise (predictions at all latitudes) even though the displacement rate distribution would suggest a slow slip event. For each event, we compute their duration as the difference between the final and initial date. We perform a static inversion for every slow slip event and we compute the moment M_0 and the fault area A, by following the same approach as in section 5.3.6 and discarding four events for which the inversion did not converge. Our final catalogue consists of 49 events.

In Figure 5.19 we show three tentative scaling laws to globally characterize SSEs in Cascadia. In Figure 5.19 (a), we show the logarithmic relationship between the

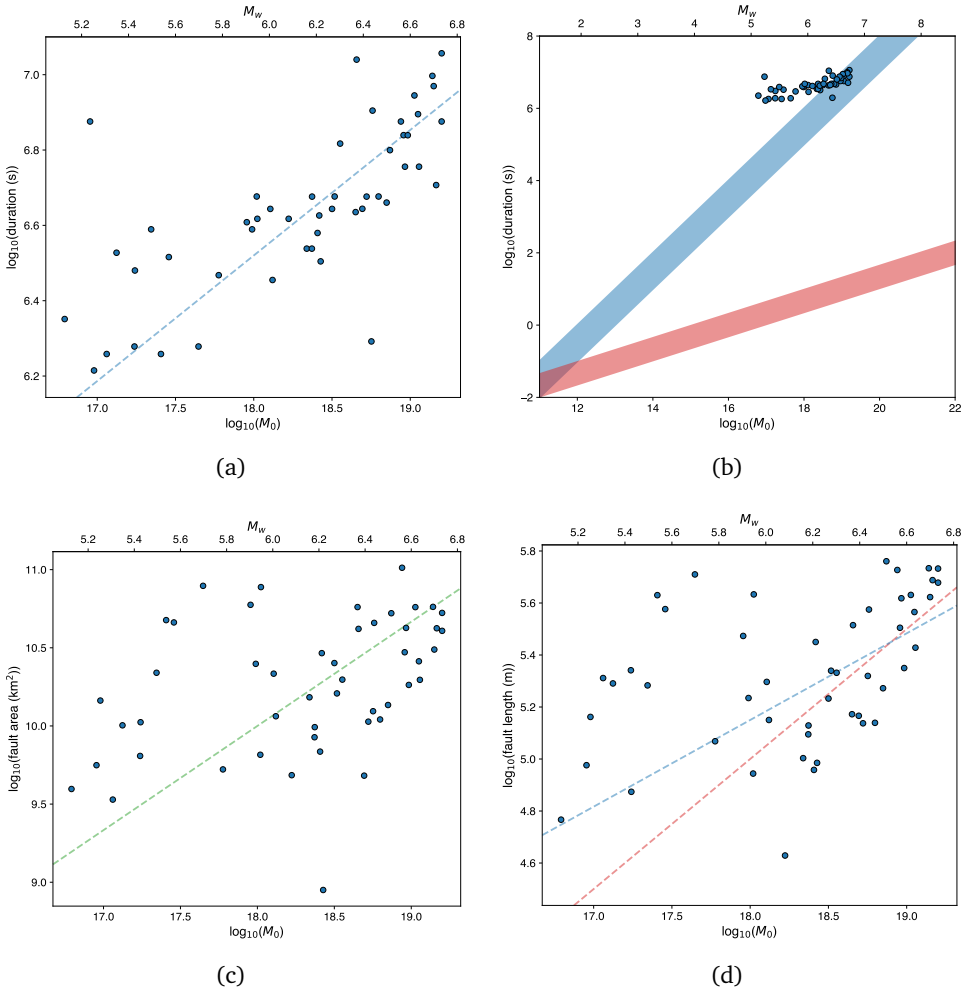


Fig. 5.19.: Tentative scaling laws for Cascadia slow slip events in 2007-2022. (a) Logarithmic relationship between moment and duration of slow slip events, compared with a cubic scaling law ($M_0 \sim T^3$) (blue dashed line). (b) Comparison with the scaling model proposed by Ide, Shelly, et al., 2007. (c) Logarithmic relationship between moment and fault slip area, compared with a 1 KPa stress drop isoline (green line). (d) Logarithmic relationship between moment and fault length L. The blue line corresponds to a cubic scaling law that Gomberg et al., 2016b associate with slow earthquakes ($M_0 \sim L^3$), while the red line is associated with a quadratic scaling law ($M_0 \sim L^2$) (closer to fast earthquake behaviour).

moment and the SSE duration (in seconds). We find that the trend of our detections generally follows a cubic scaling law ($M_0 \sim T^3$). This result is consistent with previous studies (Frank & Brodsky, 2019; Michel, Gualandi, & Avouac, 2019b), yet this is not in agreement with the originally proposed scaling laws (Gomberg et al., 2016b; Ide, Beroza, et al., 2007a; Peng & Gomberg, 2010). However, it must be noted that the manual selection of the events in our catalogue may be affected by non-objective duration biases. Thus, the cubic scaling law should not be overinterpreted, since low-magnitude events might have been associated with a duration higher than the corresponding tremor episode, which is often seen as a proxy of slow slip (e.g. Wech et al., 2009). In this case, our duration of small-magnitude events is overestimated and would rather imply a linear scaling law, as proposed by Ide, Beroza, et al., 2007a.

Our results show that the slow slip duration is comparable with the tremor duration for most of the events. Yet, smaller events seem to be associated with a higher duration. If the duration estimate is correct, it might suggest that spatial and temporal relationships between slow slip and tremor rather depend on the event magnitude, at first order, and on the rupture size. Also, the detected events in Figure 5.9 (a) seem to suggest that there is a difference in the relative timing between slow slip event initiation and tremor episode, with slow slip preceding tremor episodes. Yet, the causal relationship between SSE and tremor duration is still debated. On one hand, the increase in tremor activity could trigger the slow rupture. Conversely, slow slip may be triggering tremors. Providing evidence for the second argument with GNSS data is still challenging, given the difficulty in precisely and objectively resolving the slow slip duration because of the noise threshold. Our results seem to show that, our denoising technique has the potential to better resolve the slow slip initiation, providing new opportunities in the analysis of slow slip nucleation and the causal relationships with the associated seismological signals.

Figure 5.19 (c) shows the relationship between moment and fault area. We find that our detections follow a scaling law $M_0 \sim A^{3/2}$, as proposed by H. Gao et al., 2012; Gomberg et al., 2016b; Ide, Beroza, et al., 2007a; Schmidt and Gao, 2010. However, while they observe an alignment along a stress drop isoline of 10 KPa in Cascadia, we find an average stress drop of 1 KPa for events with $M_w > 6$ (green dashed line). This difference in stress drop might be associated with the fact that our inversion tends to generate dislocations with larger areas to fit the surface displacement. Moreover, we plot the relationship between moment and fault length in Figure 5.19 (d). Gomberg et al., 2016b propose $M_0 \sim L^3$ as a possible scaling law associated with slow slip events (blue dashed line). We find that our catalogue rather follows

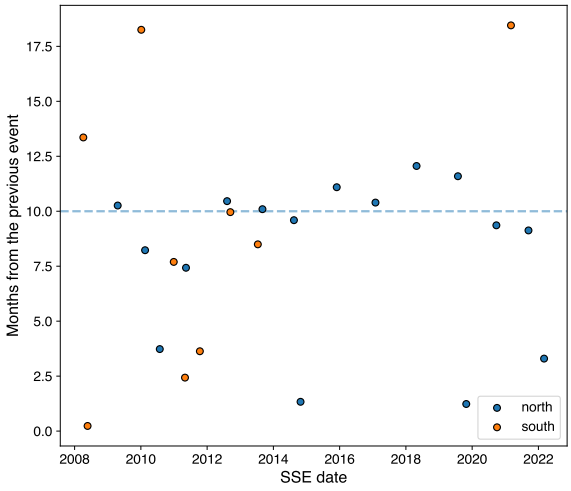


Fig. 5.20.: Interevent time distribution for the SSEs in our catalogue. Each point shows a slow slip event as a function of its date of occurrence and the temporal distance between the previous slow slip event. We filter out events having magnitudes less than 6. Blue points are associated with slow slip events in the northern part of Cascadia (latitudes $> 46°$) and orange points with events in the south (latitudes $< 44°$). The horizontal dashed line corresponds to an interevent time of 10 months, corresponding to previous findings (Aguiar et al., 2009; Brudzinski & Allen, 2007)

a $M_0 \sim L^2$ scaling law, usually associated with fast ruptures. This difference may suggest that our inversion tends to overestimate the fault length.

In Figure 5.20, we compute the interevent time between slow slip events in the north (blue points, latitudes $> 46°$) and in the south (orange, latitudes $< 44°$). We perform this analysis only on events with magnitudes higher than 6, to be more conservative, since the duration of smaller events is likely to be less well constrained. Each point represents an event as a function of the date of occurrence and the distance in time with respect to the previous event in the same latitude range. We find that slow slip events occur in North Cascadia with a recurrence interval of about 10 months (horizontal dashed line in Figure 5.20), as previously suggested by Aguiar et al., 2009; Brudzinski and Allen, 2007. The behavior in the north is at the first order compatible with a periodic behavior, while in the south the behavior is more chaotic. However, the scarcity of events, especially in 2014-2022 in the South, does not allow for further robust interpretations.

Our results show that our SSE catalogue can be resumed by tentative scaling laws. However, the event selection has been manually performed and the static inversion has not been properly tuned by taking into account specific characteristics for each event. As a consequence, our results are likely to be biased by this non-accurate parameter selection and do not have to be over-interpreted. However, the overall

features of our results suggest that, in spite of parameter tuning, deep learning can effectively denoise GNSS time series, leading to the construction of SSE catalogues that may help shed light on possible scaling relationships of slow slip events.

5.4 Conclusions

In section 5.2, we compare three deep-learning methodologies applied to slow slip characterization from synthetic GNSS data and we develop three associated methods: TSACNN, a 1D-CNN based on time series data, MCNN, a 2D-CNN working on images and STAGRNN, a spatial-temporal GNN approach. Our results show that encoding the spatial information is critical when dealing with multi-station GNSS approaches. As a result, the MCNN and STAGRNN show superior performance to TSACNN, with STAGRNN showing promising flexibility on real data, where graph-based techniques can easily exploit variable spatial configurations, thus being applicable to different tectonic contexts.

In section 5.3, we pose the problem of SSE characterization from the perspective of GNSS time series denoising. We build a synthetic database, based on an improved version of SSEgenerator (see section 4.2.1). We first analyze the characteristics of SSEdenoiser by looking at the learned adjacency matrix. We find that SSEdenoiser learns to connect stations based on the region where SSEs are located in the training samples and to the density of GNSS stations. When tested on real data, SSEdenoiser proves effective in isolating the displacement related to slow slip events, with remarkable spatial and temporal correlation with tremors, which are not given as input to the method. The obtained denoised time series are then validated by performing a static inversion of well-known events that occurred in 2013 and 2017: the estimated magnitudes are coherent with previous studies, suggesting that SSEdenoiser can effectively track and isolate the contribution due to slow slip events in raw GNSS time series. Yet, more robust analyses are needed to assess whether the precision of the method does not decrease with the magnitude.

We manually select slow slip events to build an SSE catalogue from 2007 to 2022. We invert for the SSE source parameters and we build tentative scaling laws. We find that our detections seem to follow a cubic M_0-T scaling law, and the moment-area observations follow 1 KPa stress drop isoline. Previous works proposed a linear M_0-T scaling law and an average stress drop of 10 KPa. Our slow slip events seem to share features that are peculiar to fast earthquakes, as also pointed out by some recent works (Dal Zilio et al., 2020; Frank & Brodsky, 2019; Michel, Gualandi, &

Avouac, 2019b). However, our results are still preliminary and the duration of our events has to be more objectively chosen. Also, the inversion parameters do not have been thoroughly tuned, which may explain the difference with respect to some previous studies. We also compute the inter-event time for our events. We find a recurrence time of 10 months in the north (latitudes $> 46°$) and no clear periodicity in the south (latitudes $< 44°$) for our events, as previously suggested by Aguiar et al., 2009; Brudzinski and Allen, 2007.

6 General conclusion and further perspectives

6.1 Overall conclusions

In this interdisciplinary thesis, at the intersection of geophysics and deep learning, we developed deep-learning approaches to leverage GNSS data for earthquake characterization, the identification and characterization of slow slip events and denoising of GNSS time series as an intermediate step towards slow slip event characterization. In this section, we will first show the difficulties in developing deep learning methods to work with GNSS data and we will then describe the contributions associated with this Ph.D. thesis, where we not only prove that advanced deep learning methods can successfully apply to GNSS data analysis and slow slip event detection, but also that we are able to extract scaling laws from the deep-learning characterized events (effectively trough denoising of GNSS time series), towards a better understanding of slow slip events.

6.1.1 Challenges in the use of deep learning with GNSS data

GNSS data is difficult to manage with deep learning. First, GNSS time series can be seen as a sampling of the spatial and temporal information. Each time series, recorded at a GNSS station, contains the position of that station along the time dimension. Typically, the spatial distribution of GNSS stations is not homogeneous. For this reason, the GNSS coverage does not allow for a complete mapping of the ground displacement, but several zones will remain uncovered, resulting in a sparse sampling. Since GNSS stations are not evenly distributed in space, “classic” deep learning methods, such as convolutional neural networks (CNNs), originally developed for image-like data, cannot be directly applied in our case, since the data is not formatted as a regular grid. For example, when dealing with seismic data, single-station CNN-based methods have proven effective in addressing most of the problems raised by the seismological community, such as P- and S-wave picking for earthquake detection. In our case, we demonstrate, in sections 3.3 and 3.4.2, that

CNN-based methods working on time series have some limitations that cannot be fully solved even by injecting the position of the GNSS stations, as a strategy to make the model aware of the spatial sparsity. For this, we proposed an alternative approach, consisting of creating images (and image time series) by interpolating the GNSS time series in the spatial domain. Thanks to this strategy, we show that the performance of deep learning models applied to earthquake characterization increases (see sections 3.3 and 3.4.2). However, the need for spatial information is task-dependent. When addressing slow slip event detection, in chapter 4, we show that temporal detection methods have less need for spatial consistency, and using spatial pooling has proven effective in isolating the stations recording the highest-amplitude signals. When addressing the slow slip event characterization and GNSS time series denoising in chapter 5, we show that more complex spatial encoding is needed when looking at the displacement associated with small slow slip events, which drives the need for graph-based approaches.

The second major difficulty in the use of GNSS data with deep learning lies in developing methods that have to deal with multivariate measurements. A typical GNSS data sample has the shape (stations, time steps, components). Most of the deep learning methods that are commonly employed separate the different data dimensions, under the hypothesis that each or some of the dimensions are independent. For example, in the case of SSEdetector (see chapter 4), we train a 1D-CNN model on the time steps and the N-S and E-W components and we apply spatial pooling on the station dimension. The CNN model learns shared weights among the station dimension, without fully accounting for the station information, even though the spatial pooling discards irrelevant stations. When dealing with slow slip event characterization, this method does not achieve a satisfactory performance. For this reason, we developed graph-neural-network (GNN)-based approaches (see chapter 5) to account for the full 3D information.

6.1.2 Difficulties in addressing slow slip analysis with deep learning

The first challenge in applying deep learning to SSE detection and characterization is the scarcity of catalogued events. This does not allow for proper training of deep learning models, because too few labeled events would be available. Moreover, the spatiotemporal complexity of the geodetic noise is such that most of the slow slip events are buried in the noise. Because of the high amplitude of the geodetic noise, there is a detection threshold, usually around M_w 6, hindering SSE detection. This means that almost all slow slip events with magnitude $M_w < 6$ are still systematically undetected. We address the first issue and try also to lower the detection threshold

by training our deep learning models on synthetic data. In chapter 3, 4 and 5 we present our physics-based method to build synthetic GNSS data, that has evolved, during the years, to generate a better and better synthetic data representation.

6.1.3 Overall contributions

In chapter 3, we first benchmark deep learning methods for earthquake characterization as a first step towards slow slip event detection and characterization. We associate each model with a different data configuration to evaluate the relevance of temporal, spatial, and spatiotemporal data analysis. We first show that time-series-based approaches without powerful spatial encoding lead to biased estimates of the earthquake position (see sections 6.1.1, 3.3 and 3.11). Image and image time series representations can lead to a better estimate of the earthquake source parameters down to magnitude 6, when associated with the proper deep learning approach. However, building image time series is computationally expensive, especially when longer time series (as in the case of slow slip events) are needed.

In chapter 4, we presented SSEgenerator and SSEdetector, our deep-learning-based model to generate a synthetic SSE database and to detect slow slip events in both synthetic and real data. In the development of SSEdetector, we found that spatial (station) pooling yields a good trade-off between temporal detection and the necessity for spatial encoding. With our method, we detect 78 potential slow slip events in Cascadia in 2007-2022: 87% of the events from Michel, Gualandi, and Avouac, 2019b are retrieved, each detection falling within a peak of tremor activity. Since the model is trained without this seismological counterpart, we use the correlation with tremor episodes to assess the reliability of the model.

In chapter 5, we first develop and compare temporal and spatiotemporal deep learning models to address slow slip event characterization, demonstrating that spatial-temporal graph neural networks have the potential to perform reliable slow slip characterization as well as providing flexibility in different tectonic contexts. In fact, graph neural network are capable of learning relationships between the GNSS stations as well as better taking into account their density and spatial distribution. However, such methods do not yield satisfactory results in the estimation of the slow slip souce parameters, since the detection threshold magnitude is still high.

In the second part of chapter 5, we address the problem of the denoising of GNSS time series. Our idea is that the major limitation preventing us to make systematic and effective slow slip identification is the too-high level of noise in the GNSS time

series. If this noise would be attenuated or, ideally, removed, SSE characterization would be carried out more effectively. With this objective in mind, we developed SSEdenoiser, a spatiotemporal attentive graph neural network, trained to recover the slow slip event signature in GNSS time data by learning how to separate it from the noise. Our proof of concept showed effectiveness in denoising GNSS data in Cascadia in 2007-2022, revealing slow slip events that prove remarkable alignment with the spatiotemporal tremor distribution as well as previously studied slow slip events.

In the last part of chapter 5, we further leverage our idea and we build a preliminary SSE catalogue, by manually selecting events from the denoised time series. We perform a static inversion of all the events to characterize their source parameters and to build preliminary scaling laws with our events. Our preliminary results showed that our detected events follow a cubic moment-duration scaling law and have an average stress drop of 1 KPa. The interevent time between SSEs with magnitude $M_w > 6$ is 10 months, consistent with previous studies.

6.1.4 Scientific potential

This work has shown that deep learning approaches can be applied to the detection and characterization of slow slip events in raw GNSS time series by (1) building realistic synthetic data to reproduce the variability of the geodetic noise and generate simulations of artificial slow slip events (or earthquakes) based on physical models and (2) by developing advanced deep learning methods trained to reveal the spatiotemporal signature of the events that we are looking for.

This approach has the potential to help geoscientists analyze geodetic data towards a better characterization of the underlying physical processes. By automatically extracting meaningful features from the data, deep learning has the added value of dealing with large data sets in an automatic manner, performing data assimilation to distill relevant information that can be made more accessible to experts. Moreover, our approach can be extended to other domains, especially in applications where it is necessary to analyze multivariate data jointly with physical models.

When discussing slow slip event characterization through denoising of the GNSS time series, we show that our deep learning model has learned an adjacency matrix from the data, *i.e.*, the optimal connections between GNSS stations in order to perform data denoising at best. We found that these connections arise from a learned trade-off between the local density of stations and the distance from the

area where slow slip events occur. This has the potential to ameliorate the GNSS geometry for a given task, by relying on optimal and essential connections and by discarding stations that revealed less useful during training, possibly removing noisy observations and optimizing the training procedure.

6.1.5 Limitations of the current work

Our approach used to detect and characterize slow slip events has some limitations, that we will briefly discuss here. First, SSEdetector does not allow for the localization of the detected events and it cannot distinguish two events occurring in the same window. We address these limitations in the development of the SSE characterization methods (in chapter 5), yet two events occurring in the same observation window cannot be detected separately. In fact, this is a major issue arising from the architecture of the deep learning models that we develop for the characterization. In fact, these models are trained to provide an estimate of the location and magnitude of a slow slip event occurring in the middle of the observation window, and allowing for more complex outputs (*e.g.*, a variable number of outputs per window) is hard to accomplish. This could be made with sequence-to-sequence models (*e.g.*, RNNs, Transformers), yet it is not guaranteed that the performance in the characterization would increase.

We address this last issue with SSEdenoiser. We take advantage of the GNN framework to make the model recognize multiple events in the same window (in our case, up to three). Since our output becomes a (denoised) displacement field, recognizing multiple events becomes more straightforward, because this information would be encoded in the displacement. SSEdenoiser proves effective in solving the task of denoising GNSS time series up to sub-millimeter precision and we make use of these denoised time series to perform slow slip characterization through static inversion of the selected events. However, it should be noted that our results are still preliminary and our duration estimate is still rough (manually selected) and will need to be computed with more objective methods. Also, we use tremors to help in the event selection process, which may also bias our analysis.

6.2 Further perspectives

Our work opens several perspectives in the direction of the development of more complex deep-learning methods as well as new research trails that can be pursued.

First, the denoising of GNSS time series showed great potential as an aid for slow slip detection and characterization. In fact, simultaneously event detection and characterization is not an easy task to accomplish with deep learning, because classification (detection) and regression (characterization) tasks can hardly be mixed. SSEdenoiser solves this technical issue since the two tasks can be performed after the denoising step. For the detection, clustering methods could be used, such as ST-DBSCAN (Birant & Kut, 2007), to perform more objective event selection than manual selection. Slow slip event characterization could be still performed by classic static inversion methods. Alternatively, deep learning methods could also be used for this task, which would be simpler than the ones developed here, since the geodetic noise would not be present anymore. For this, either spatiotemporal deep learning methods could be used, working with denoised time series as input, or we can think of even simpler methods, dealing with the static displacement information only.

Once a robust method for event selection and characterization has been set up, our method can be applied to other subduction zones. In the direction of systematic SSE detection and characterization, the application of our method to other regions is one of the very next steps. However, this is not just a matter of generating the correct synthetic data targeted to the specific region: it can require further analyses. For example, the application of our method to the Japanese subduction zone would not probably work because of the intense seismic activity that would affect the time series. In this case, earthquakes and the associated post-seismic relaxation signals would prevent SSEgenerator from generating the correct synthetic noise time series. For this, more work is needed to refine the way in which synthetic data is built. Moreover, other regions, like the Mexican or the Chilean subduction zones are not as well instrumented as Japan. Here, an improved method to deal with long periods of missing data could be required as well as dealing with possible gaps in the GNSS coverage, as in the case of the Mexican subduction. Our GNN-based method could learn how to deal with a very sparse GNSS coverage, yet there is no guarantee that the method would obtain satisfactory results without structural modifications.

Finally, the possibility of facing a very sparse and heterogeneous GNSS network can also be seen as the chance to integrate multiple data sets to complement the missing information and to improve the detection and characterization. However, it is not trivial to use different data types at once, *e.g.*, GNSS, seismic data, InSAR, strainmeters, tiltmeters, pressure gouges, etc. This poses some issues from the technical point of view, because the development of hybrid deep learning methods is challenging, but also from the viewpoint of data preparation, since each type of data has its own difficulties and it is not obvious to extract information from all of them (*e.g.*, extracting slow slip signals from strainmeter data). However, this direction

is quite promising and, albeit extremely challenging, could result in a paradigm shift towards the use of heterogeneous data sets. In fact, the detection threshold in GNSS data for slow slip event detection can be significantly lowered by integrating, for example, strainmeter and pressure data, making small slow slip events more accessible.

List of Figures

List of Tables

List of Listings

A

Supporting Information for “Seismic source characterization from GNSS data using deep learning”

Supplementary figures for chapter 3.

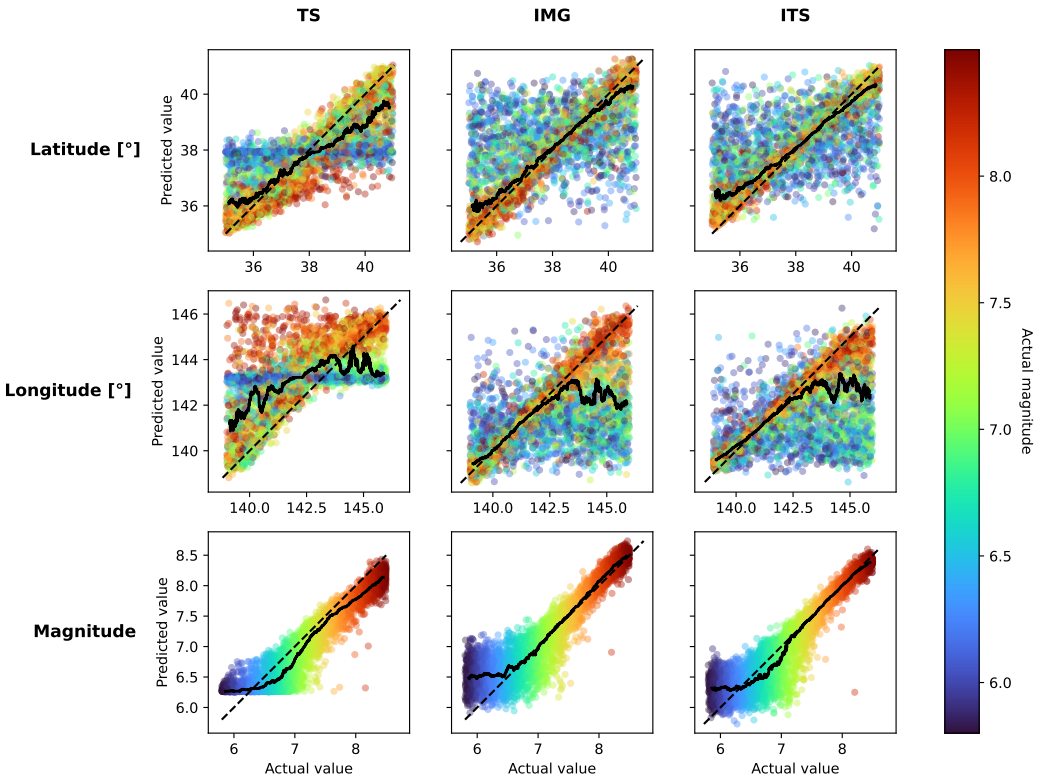


Fig. A.1.: Comparison of the performance of the deep learning models on a synthetic data set with post-seismic signals following the co-seismic offset. TS, IMG and ITS models are shown on columns respectively. For each row, latitude, longitude and magnitude predictions are reported, respectively. Each 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.

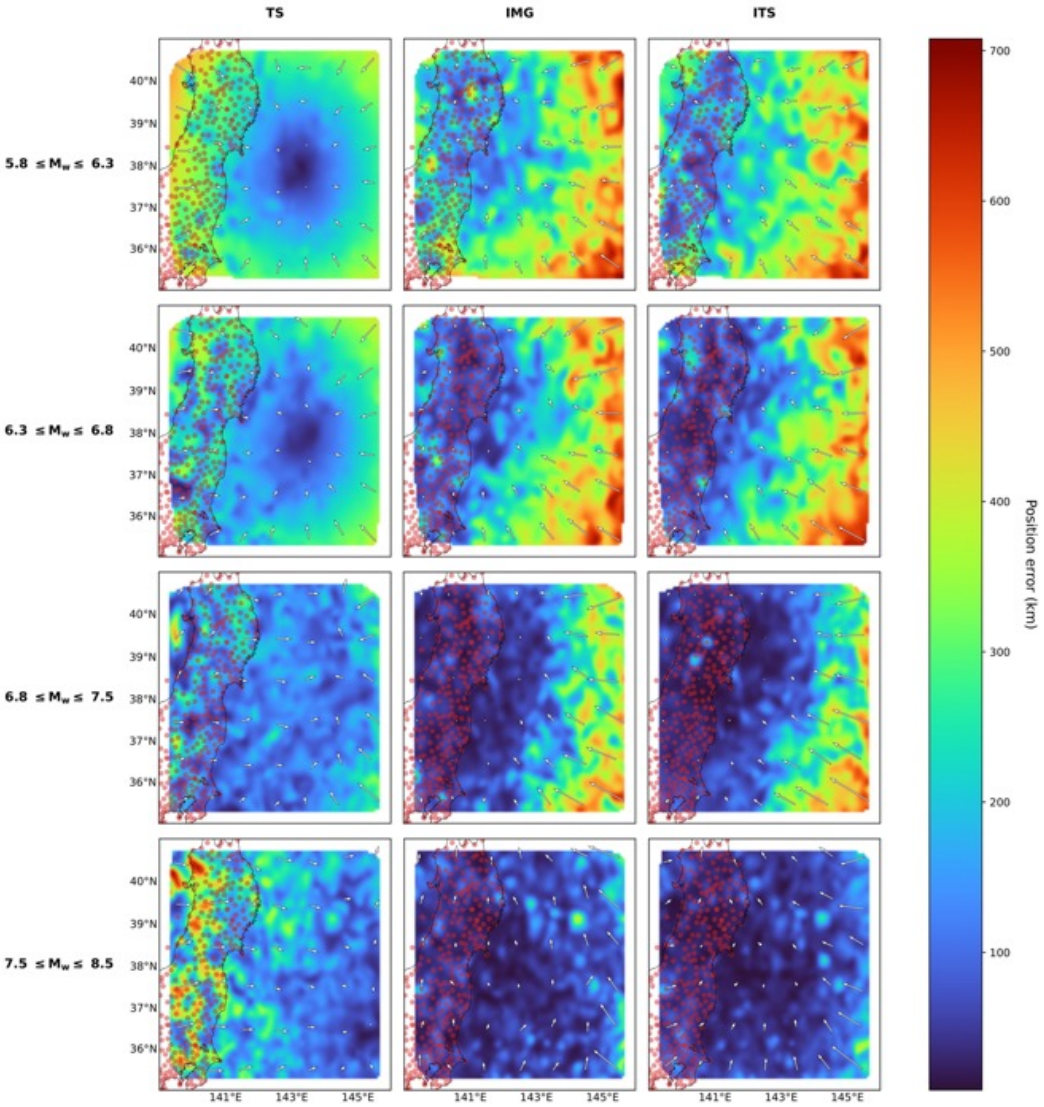


Fig. A.2.: Comparison of the location error of the deep learning models on a synthetic data set with post-seismic signals following the co-seismic offset. Each subplot shows the location error associated with the test samples, interpolated on a grid whose corresponding spatial coordinates are indicated along the axes. Magenta data points represent the position of 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.

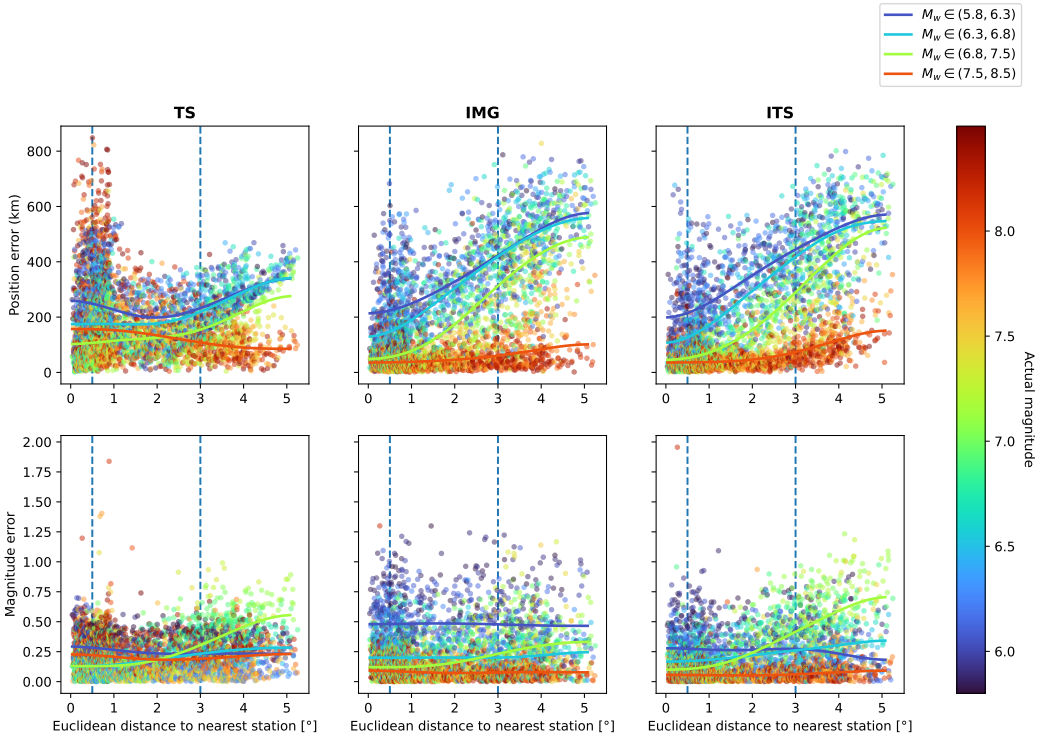


Fig. A.3.: Comparison of errors as a function of the distance to the nearest GNSS station on a synthetic data set with post-seismic signals following the co-seismic offset. The deep learning models are shown in columns, while the rows indicate position and magnitude errors, respectively. Each scatter 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.

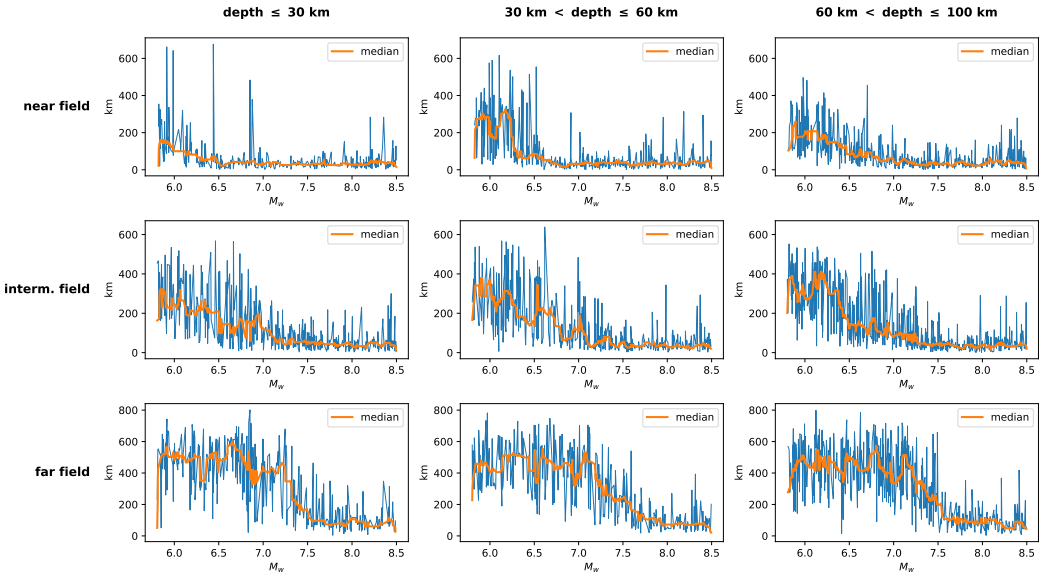


Fig. A.4.: Position error, computed for each test sample on a synthetic data set with post-seismic signals following the co-seismic offset, as a function of the magnitude (x-axis), the depth range (columns) and the distance range (cf. Figure 3.7) with respect to the GNSS network (rows) for ITS. The orange solid line represents the result of a median smoothing by employing a kernel size of 15 points.

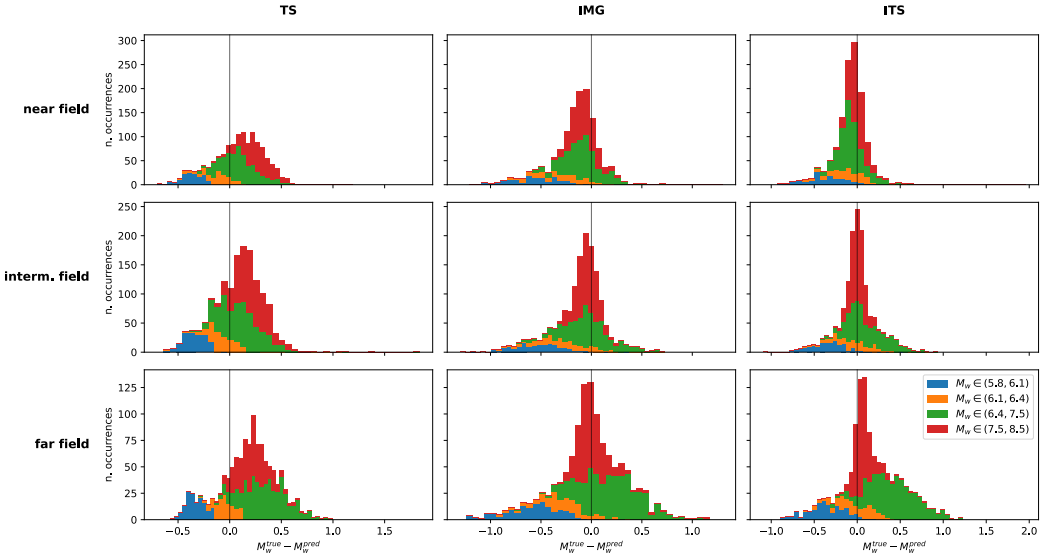


Fig. A.5.: Stacked histograms of the magnitude difference $M_w^{true} - M_w^{pred}$ for the three models (columns) on a synthetic data set with post-seismic signals following the co-seismic offset, as a function of the distance range (cf. Figure 3.7) with respect to the GNSS network (rows) and for different magnitude ranges (color). Vertical lines individuate the zero.

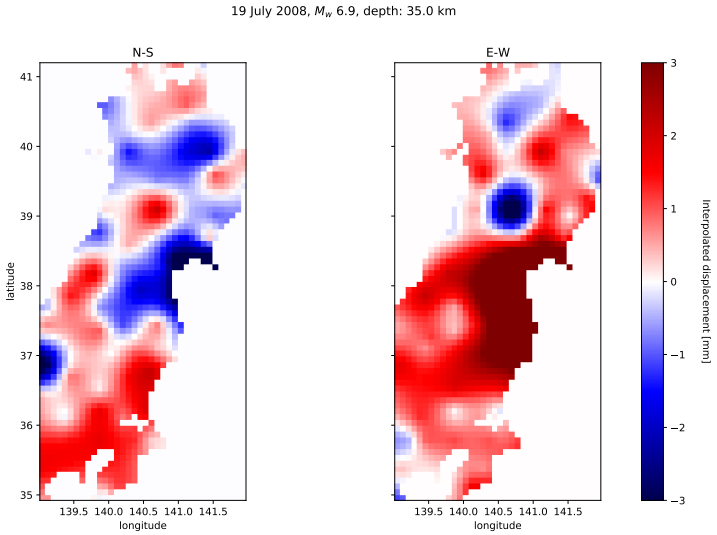


Fig. A.6.: Differential images associated with the 19 July 2008 for the **ISTerre/DD** data set. The deformation value has been saturated over ± 3 mm.

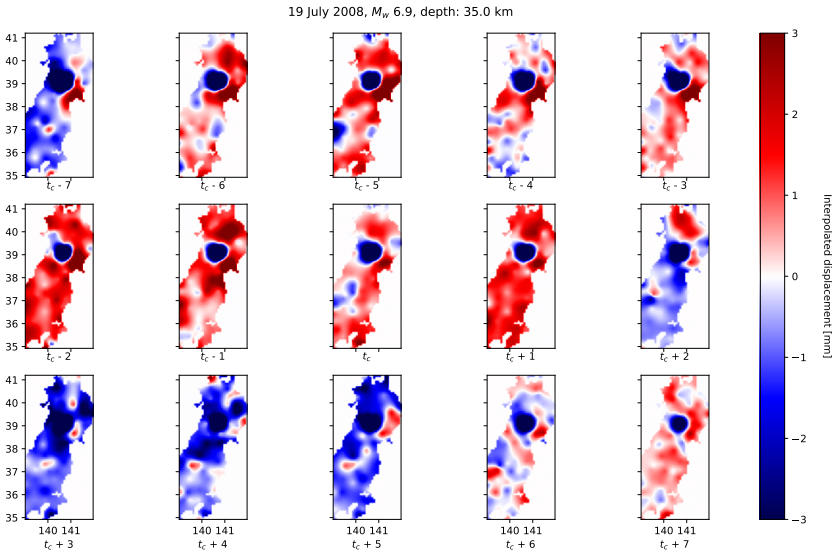


Fig. A.7.: Image time series (N-S component) associated with the 19 July 2008 for the **ISTerre/DD** data set. The deformation value has been saturated over ± 3 mm. Each frame is associated with the day written below (e.g., $t_c - 2$, where t_c is the time associated with the coseismic offset).

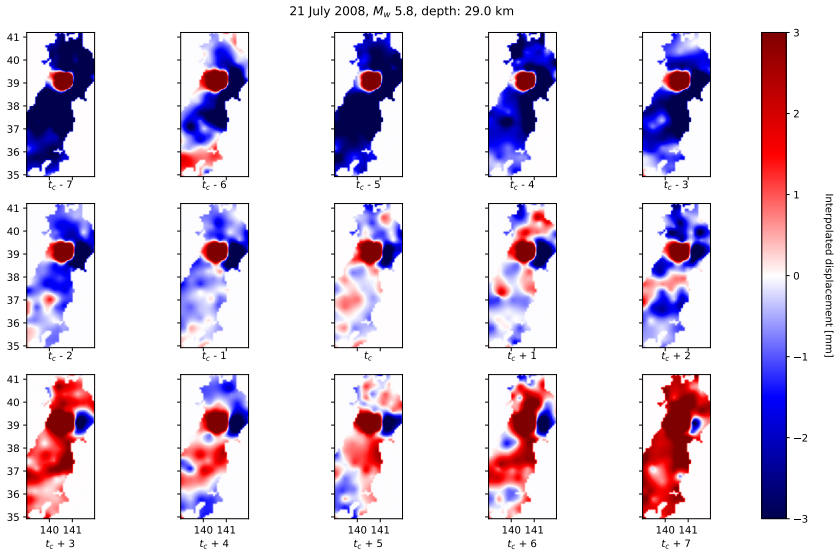


Fig. A.8.: Image time series (E-W component) associated with the 19 July 2008 for the **ISTerre/DD** data set. The deformation value has been saturated over ± 3 mm. Each frame is associated with the day written below (e.g., $t_c - 2$, where t_c is the time associated with the coseismic offset).

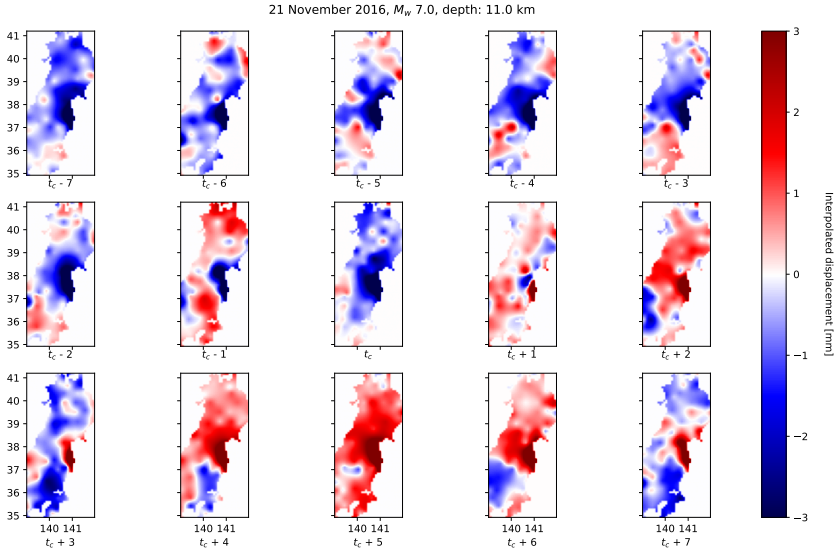


Fig. A.9.: Image time series (N-S component) associated with the 21 November 2016 for the **ISTerre/DD** data set. The deformation value has been saturated over ± 3 mm. Each frame is associated with the day written below (e.g., $t_c - 2$, where t_c is the time associated with the coseismic offset).

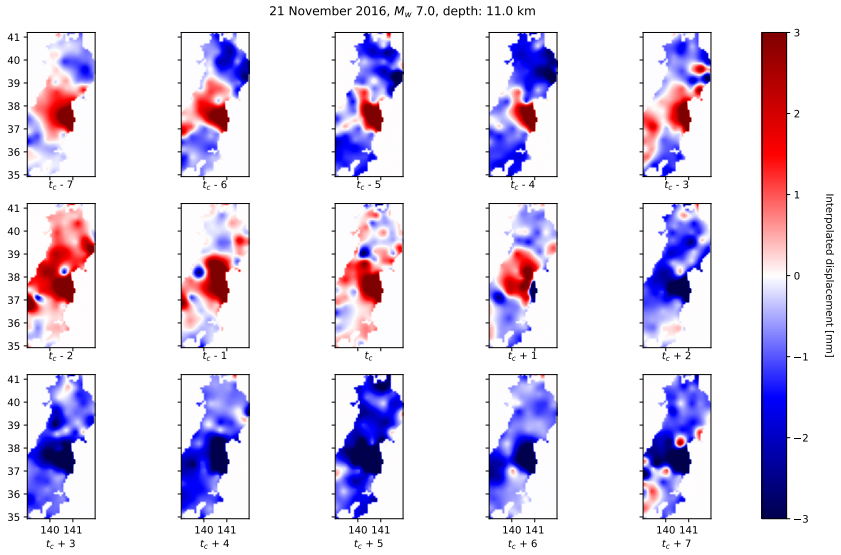


Fig. A.10.: Image time series (E-W component) associated with the 21 November 2016 for the **ISTerre/DD** data set. The deformation value has been saturated over ± 3 mm. Each frame is associated with the day written below (e.g., $t_c - 2$, where t_c is the time associated with the coseismic offset).

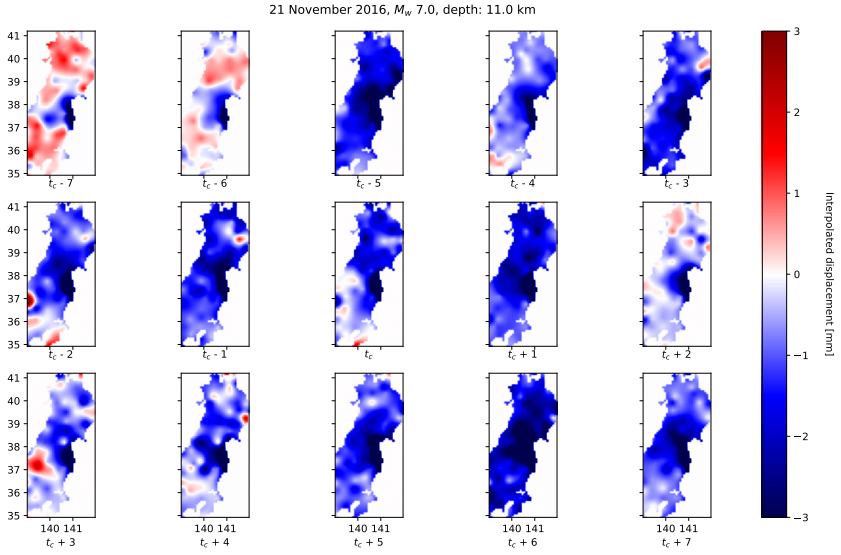


Fig. A.11.: Image time series (N-S component) associated with the 21 November 2016 for the **NGL/PPP** data set. The deformation value has been saturated over ± 3 mm. Each frame is associated with the day written below (e.g., $t_c - 2$, where t_c is the time associated with the coseismic offset).

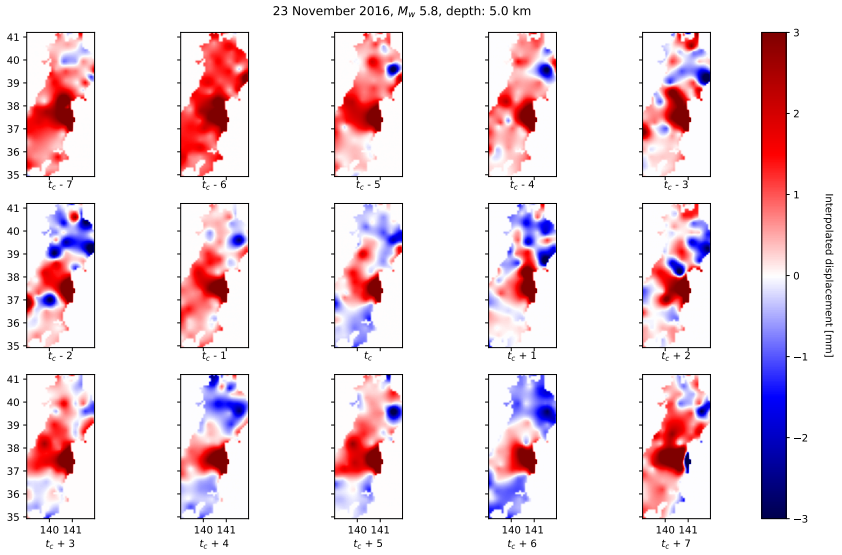


Fig. A.12.: Image time series (E-W component) associated with the 21 November 2016 for the **NGL/PPP** data set. The deformation value has been saturated over ± 3 mm. Each frame is associated with the day written below (e.g., $t_c - 2$, where t_c is the time associated with the coseismic offset).

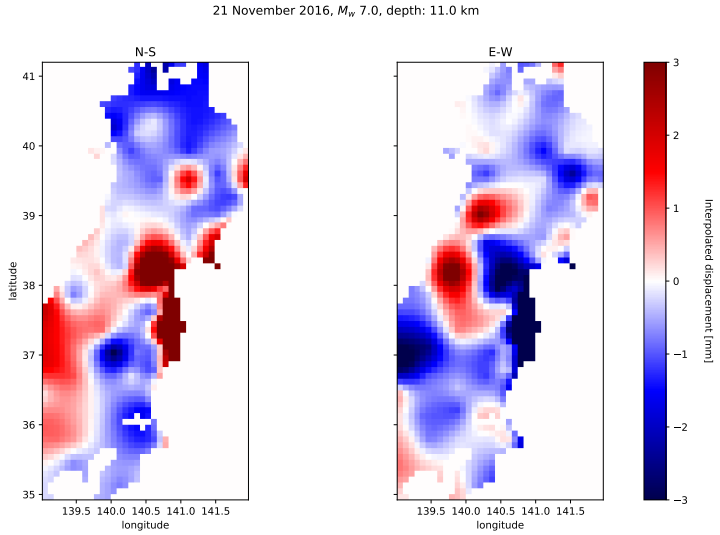


Fig. A.13.: Differential images associated with the 21 November 2016 for the **ISTerre/DD** data set. The deformation value has been saturated over ± 3 mm.

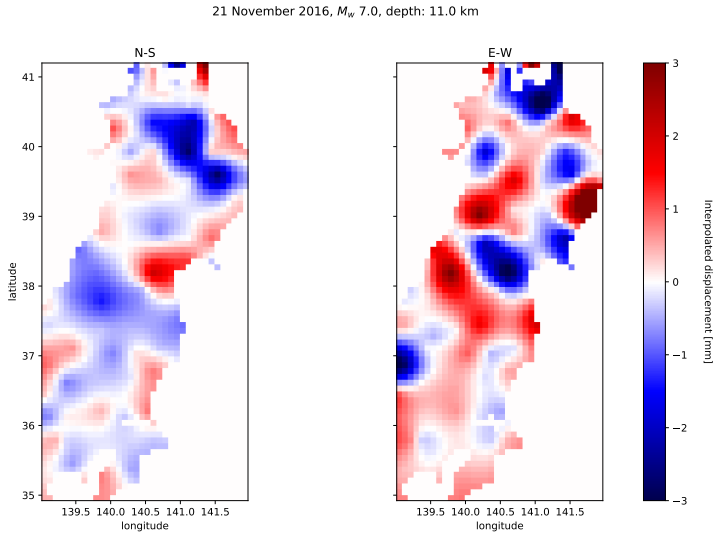


Fig. A.14.: Differential images associated with the 21 November 2016 for the **NGL/PPP** data set. The deformation value has been saturated over ± 3 mm.

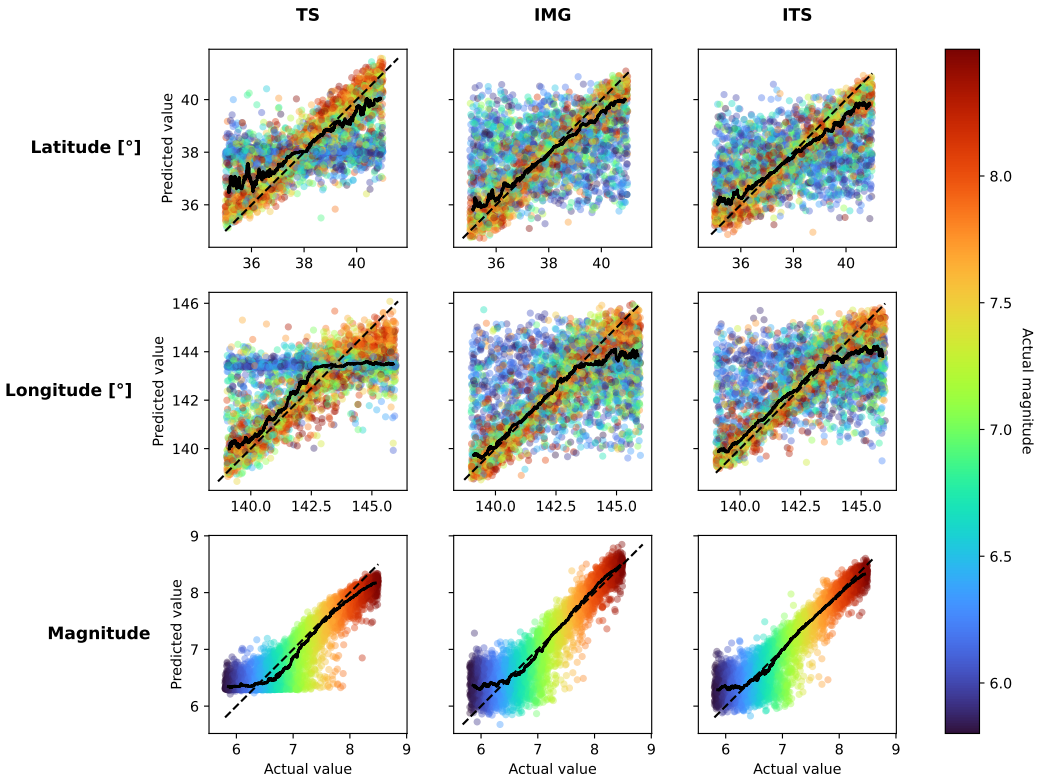


Fig. A.15.: Comparison of the performance of the deep learning models trained on a synthetic data set with all focal mechanisms. TS, IMG and ITS models are shown on columns respectively. For each row, latitude, longitude and magnitude predictions are reported, respectively. Each point of the scatter 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.

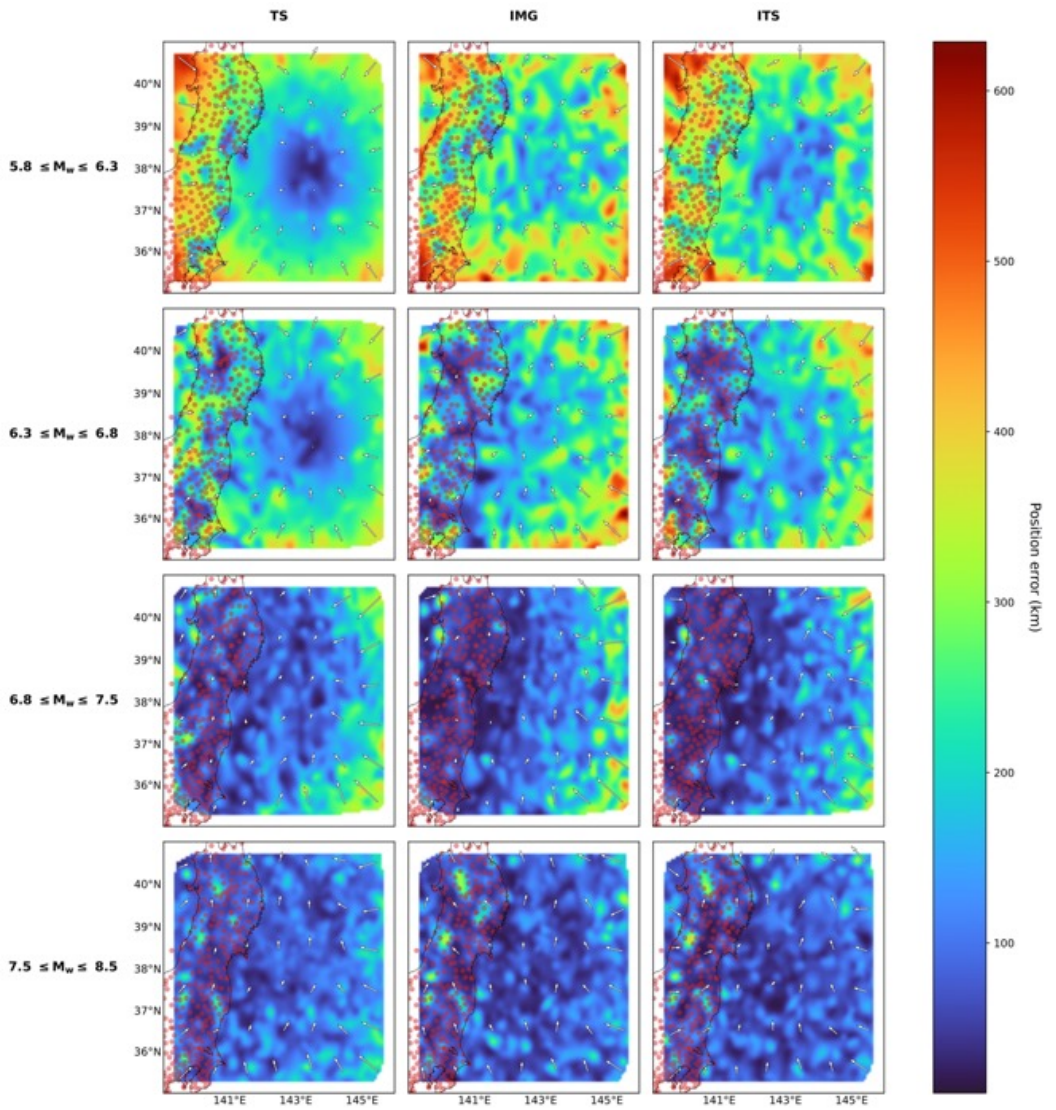


Fig. A.16.: Comparison of the location error of the deep learning models trained on a synthetic data set with all focal mechanisms. Each subplot shows the location error associated with the test samples, interpolated on a grid whose corresponding spatial coordinates are indicated along the axes. Magenta data points represent the position of 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.

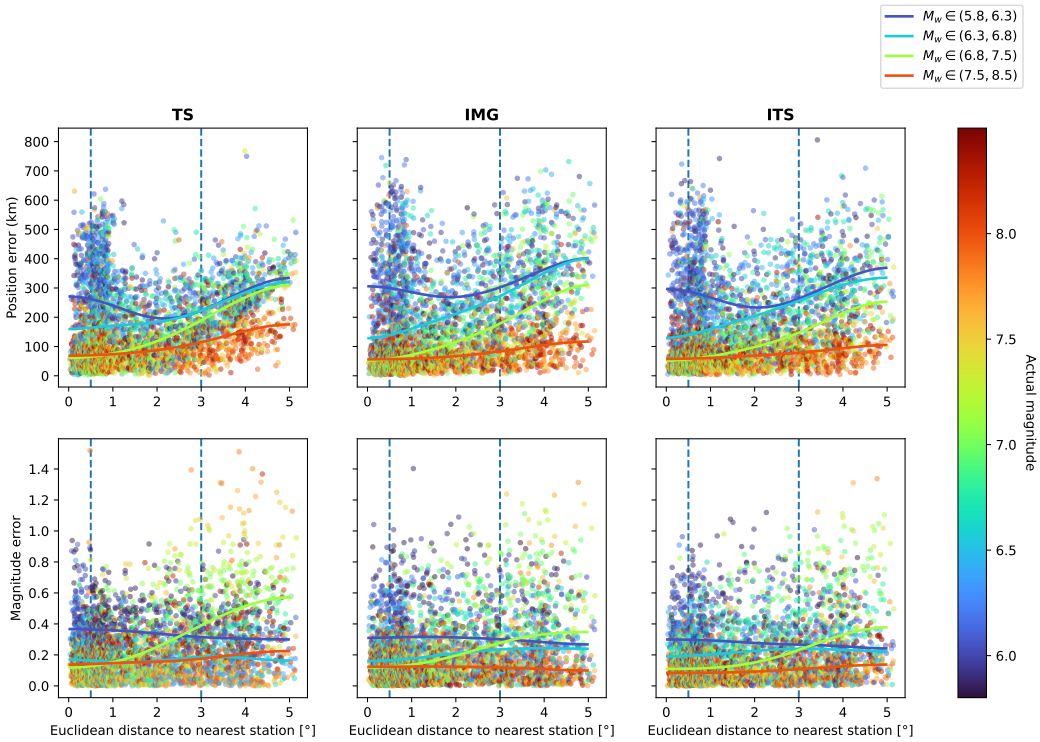


Fig. A.17.: Comparison of errors as a function of the distance to the nearest GNSS station of the models trained on a synthetic data set with all focal mechanisms. The deep learning models are shown in columns, while the rows indicate position and magnitude errors, respectively. Each scatter plot depicts 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.

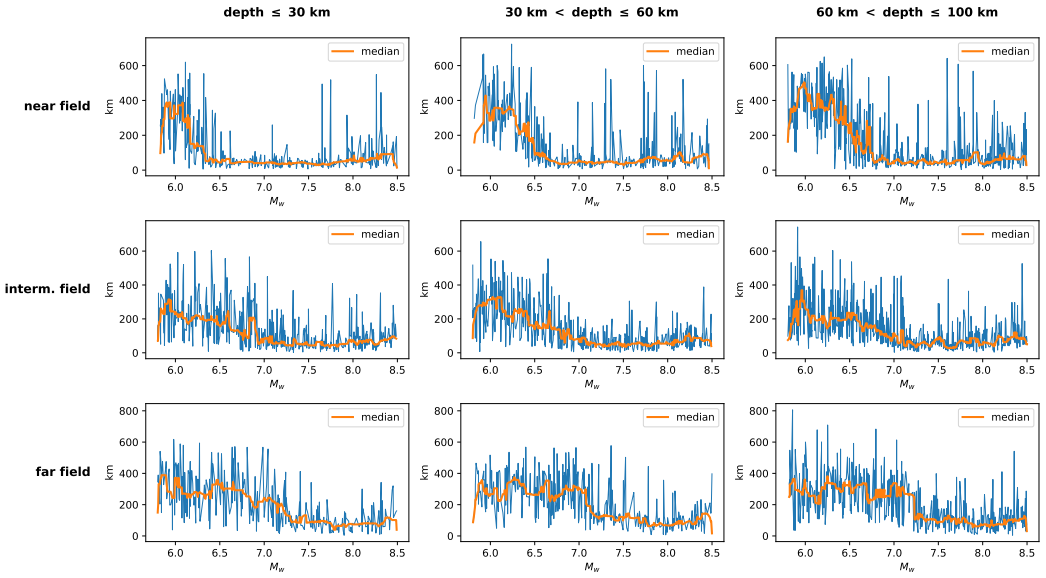


Fig. A.18.: Position error, computed for each test sample, as a function of the magnitude (x-axis), the depth range (columns) and the distance range (cf. Figure 3.7) with respect to the GNSS network (rows) for ITS, trained on a synthetic data set with all focal mechanisms. The orange solid line represents the result of a median smoothing by employing a kernel size of 15 points.

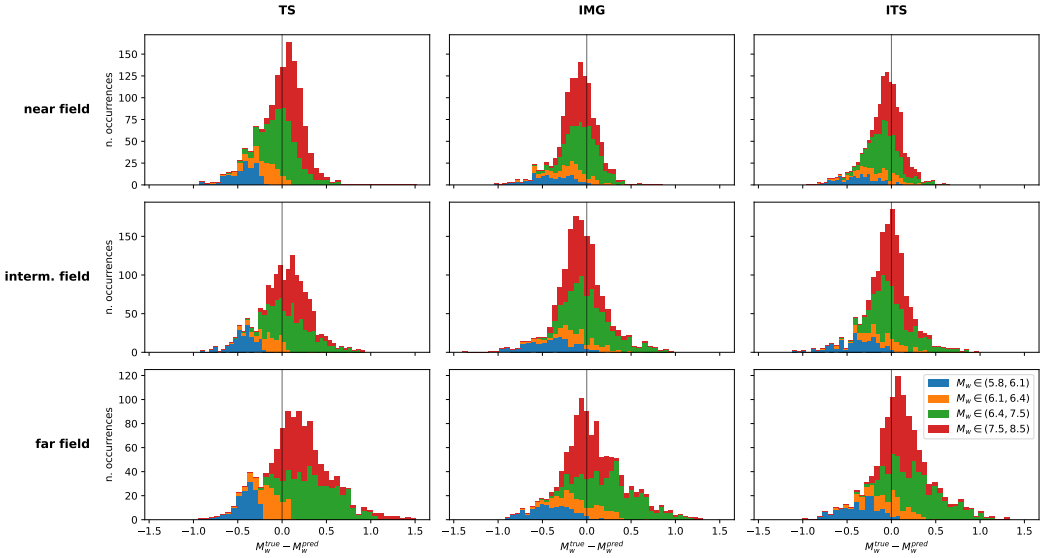


Fig. A.19.: Stacked histograms of the magnitude difference $M_w^{true} - M_w^{pred}$ for the three models (columns) trained on a synthetic data set with all focal mechanisms, as a function of the distance range (cf. Figure 3.7) with respect to the GNSS network (rows) and for different magnitude ranges (color). Vertical lines individuate the zero.

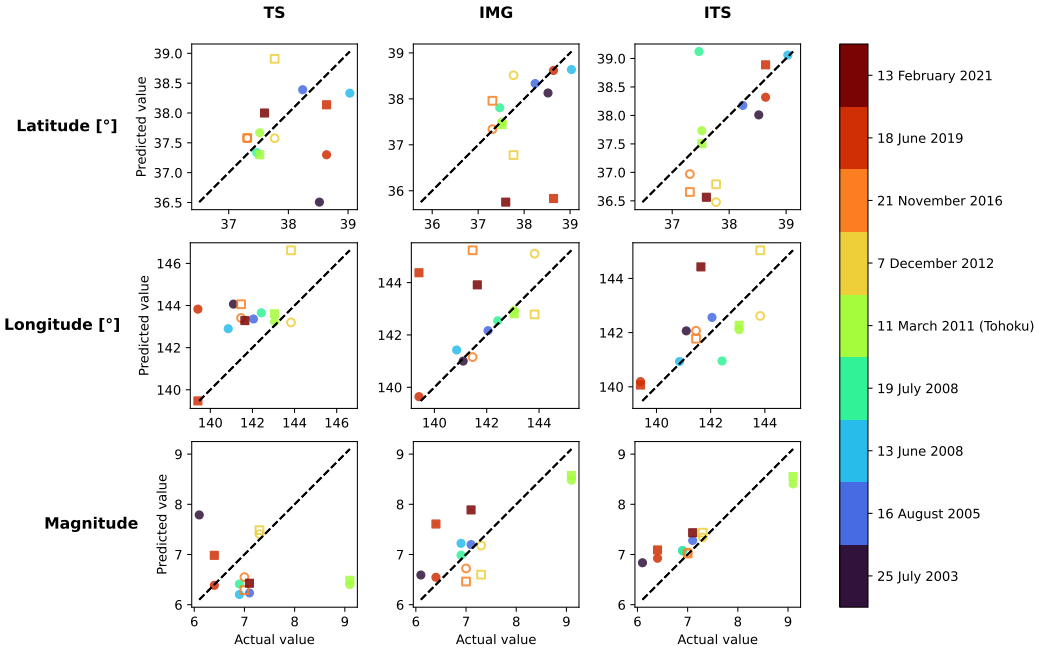


Fig. A.20.: Actual *vs* predicted plot on real data from **ISTerre/DD** and **NGL/PPP** data sets for the models trained on a synthetic data set with all focal mechanisms. Each subplot shows the real vs predicted comparison for the estimated parameters (fault centroid latitude, fault centroid longitude 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** data sets, 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.

B

Supplementary Information for “Slow slip detection with deep learning in multi-station raw geodetic time series validated against tremors in Cascadia”

Supplementary figures for chapter 4.

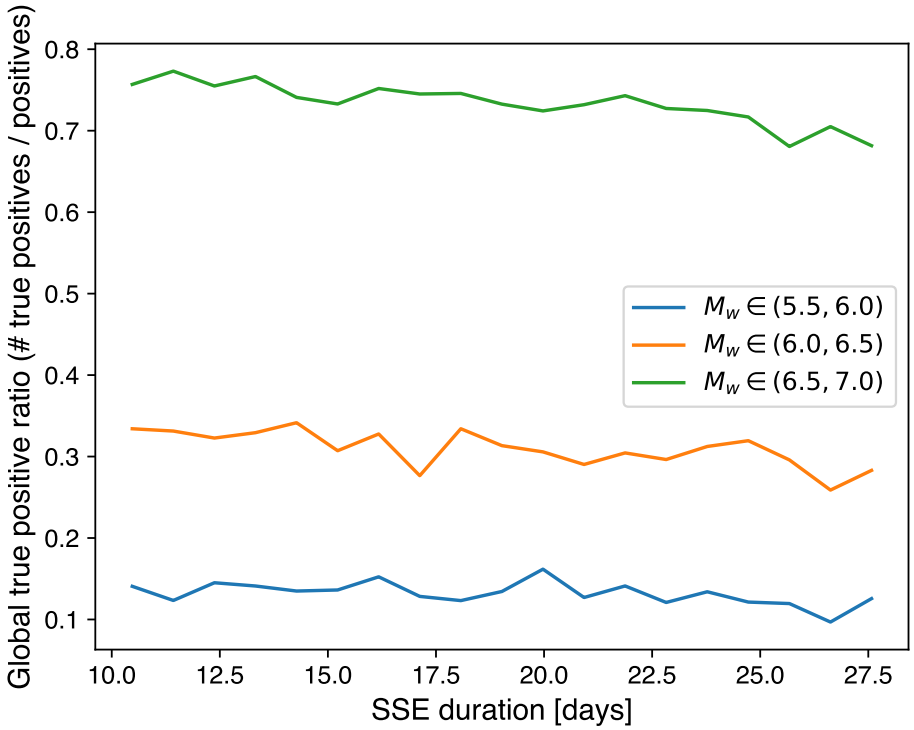


Fig. B.1.: (Complementary) performance of SSEdetector on synthetic data. The curves represent the true positive rate as a function of the SSE duration for magnitude ranges, as shown in the legend.

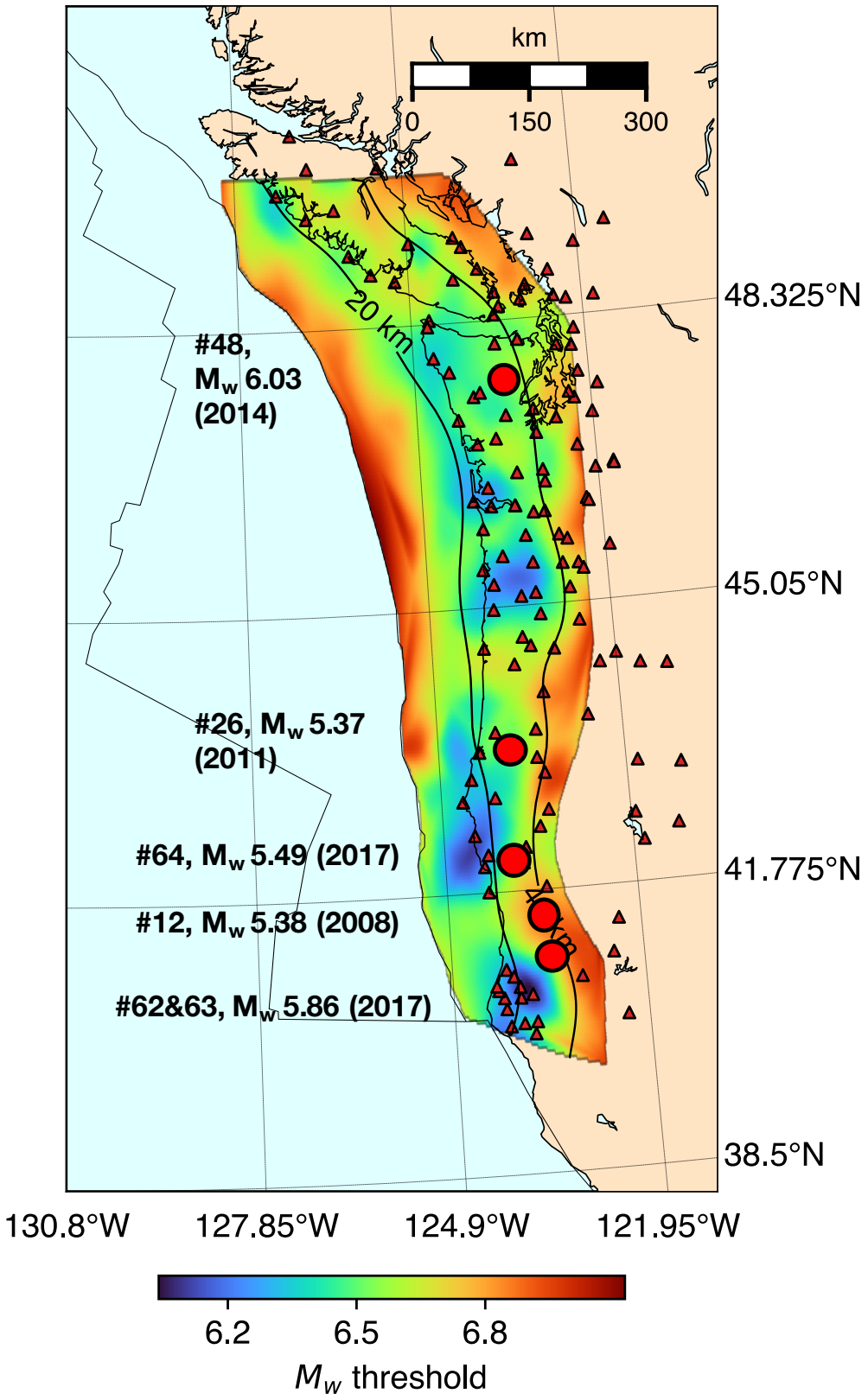


Fig. B.2.: Detail of the 5 missed events in the Michel et al. catalogue. The map shows each missed event from the Michel et al. catalogue, detailed with the occurrence date, location and magnitude estimated by Michel et al. The points are superimposed on the magnitude threshold map (see. Figure 2(b)).

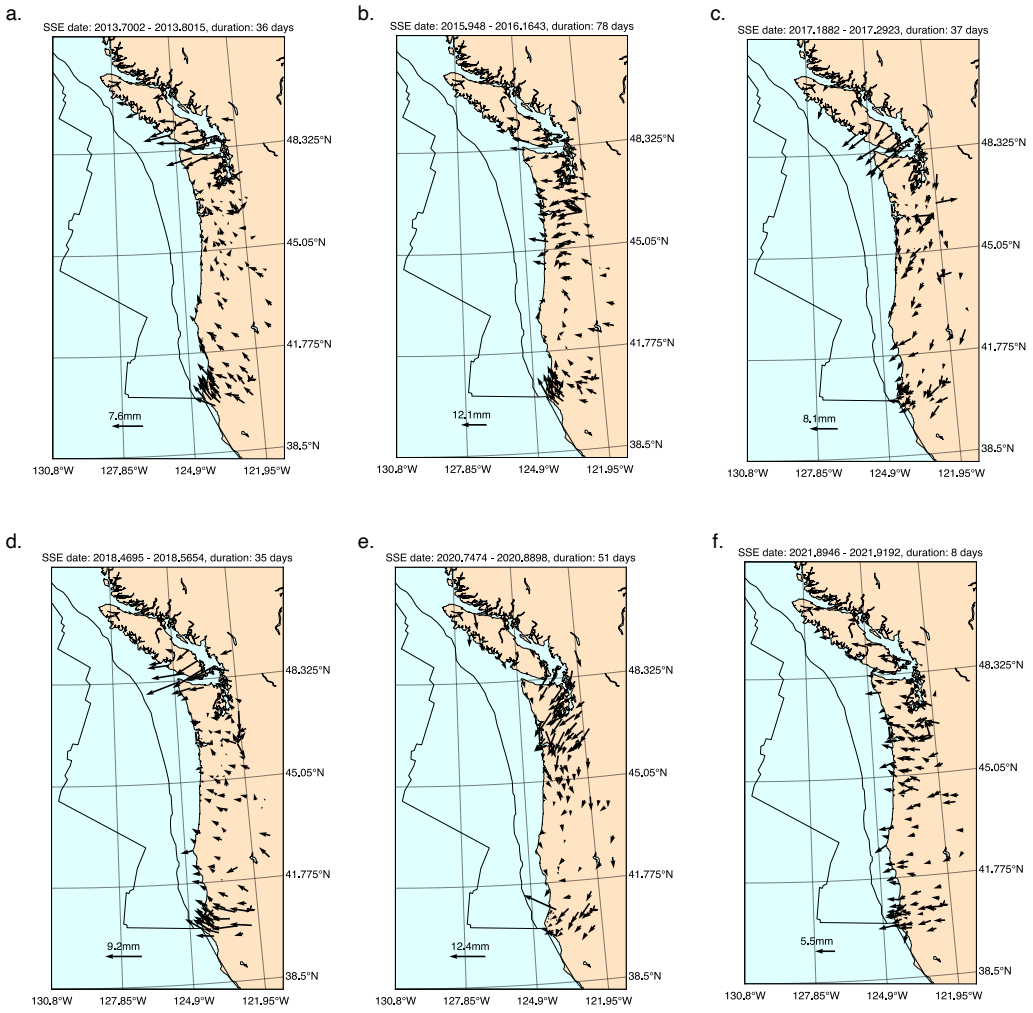


Fig. B.3.: Displacement field associated with some of the detected SSEs. (a) Estimated SSE period: from 13 September to 20 October 2013. This event has been studied by Bletery & Nocquet, 2020 and is also present in the Michel et al. catalog (#41). (b) Estimated SSE period: from 13 December 2015 to 1 March 2016. This event is also present in the Michel et al. catalog (#54 & 55). (c) Estimated SSE period: from 10 March to 17 April 2017. This event is has been studied by Itoh et al., 2022 and is also present in the Michel et al. catalog (#59). The following events could likely be new detections: (d) from 21 June to 26 July 2018, (e) from 30 September to 21 November 2020, (f) from 23 November to 2 December 2021.

Sensitivity study: train and test with 352 stations

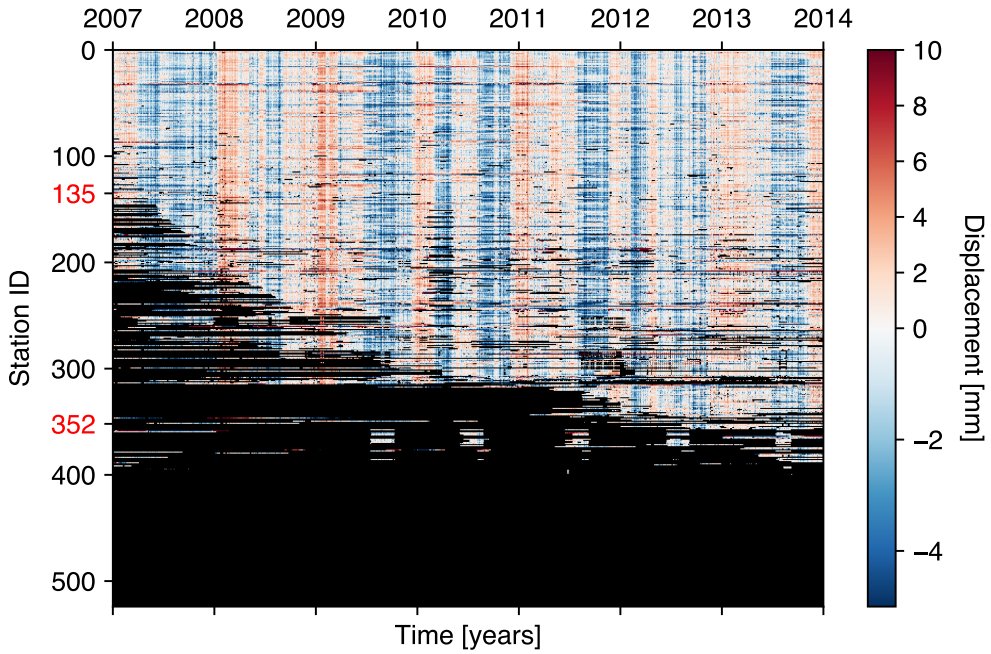


Fig. B.4.: GNSS data in (2007, 2014), sorted by the number of data gaps. Each row of the matrix corresponds to a GNSS station and is color-coded by the position value on the detrended time series. Two of the y-axis labels have been marked in red to show the range of stations used in the 135- and 352-station subsets. Black entries in the matrix represent data gaps.

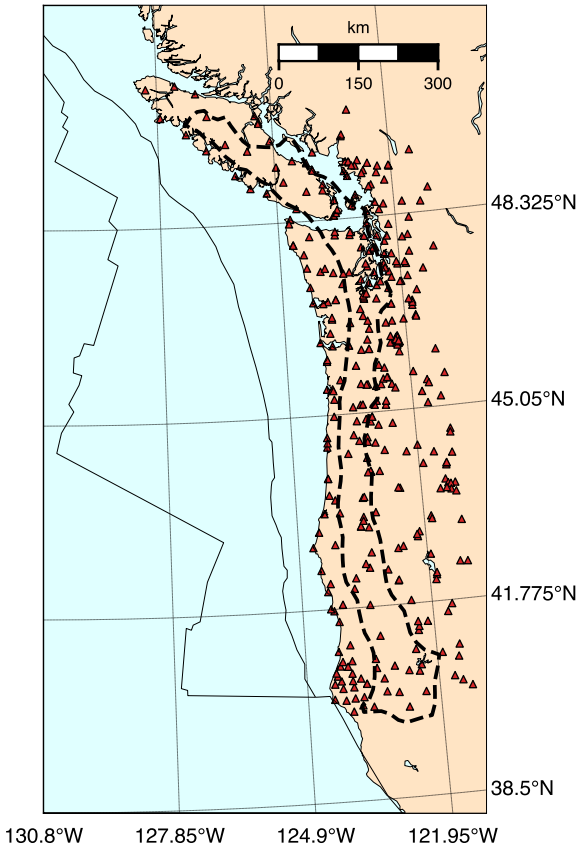


Fig. B.5.: **Location of the 352 best stations used in the sensitivity study.** Triangles show the location of the GNSS stations. The dashed black contour represents the tremor locations from the PNSN catalog.

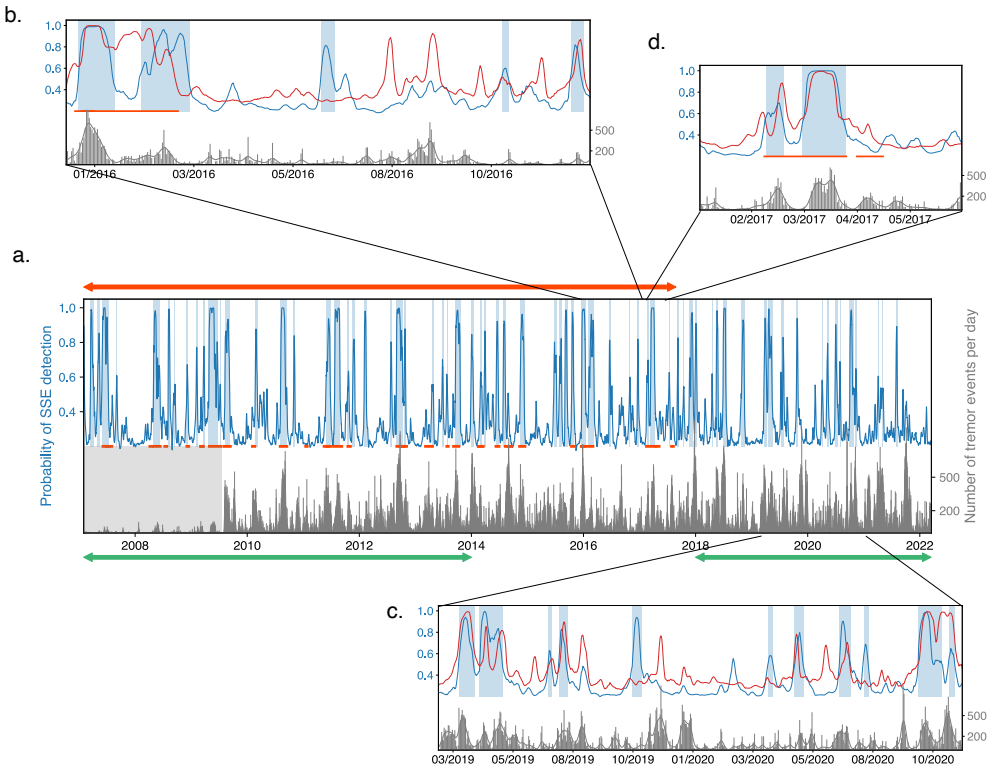


Fig. B.6.: Performance of SSEdetector on real raw GNSS time series, consisting of 352 GNSS stations (cf. Methods). For a detailed explanation of the figure, refer to Figure 3 in the main text. The red curve in the (b), (c) and (d) panels shows the probability curve of (the preferred) SSEdetector (see Figure 3), with the blue curve associated with SSEdetector trained on 352 stations.

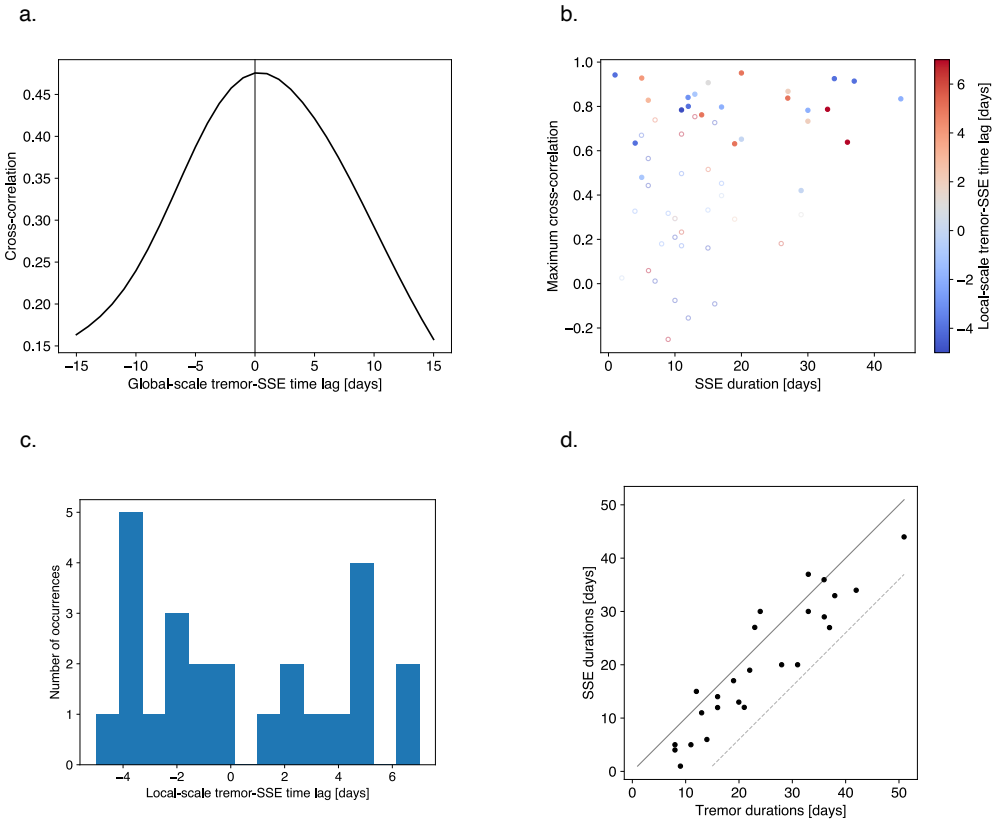


Fig. B.7.: Validation of SSEdetector performance against tremor activity in 2010-2022, from 352 GNSS stations. For a detailed explanation of the figure, refer to Figure 5 in the main text. The maximum global correlation value is 0.47 (compared to 0.58 obtained for the set of 135 stations), with a global lag value of 1 day (compared to 2 days for the 135-station subset). At a local scale, the lag distribution seems to be bimodal, with a lag of around 4 days (deformation before tremors) and 3 days (tremors before deformation), compared to the 2-day time lag of SSEdetector (see Figure 5c).

Sensitivity study: test with a subset of GNSS stations in the northern part of Cascadia

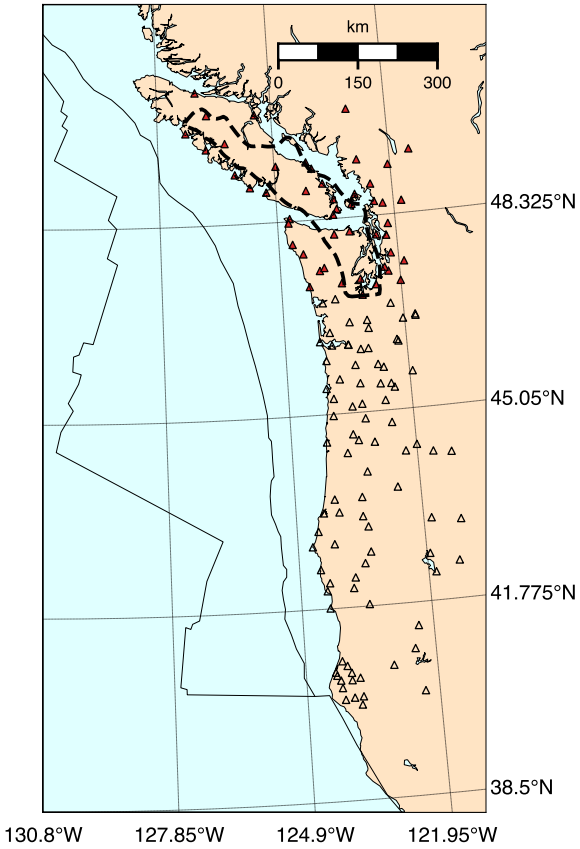


Fig. B.8.: Location of the subset of GNSS stations in North Cascadia. Red triangles show the location of the GNSS stations that have been used in the sensitivity study, while empty triangles show the stations that have been removed. The dashed black contour represents the tremor locations from the PNSN catalog corresponding to latitudes higher than 47 degrees that have been considered in the analysis.

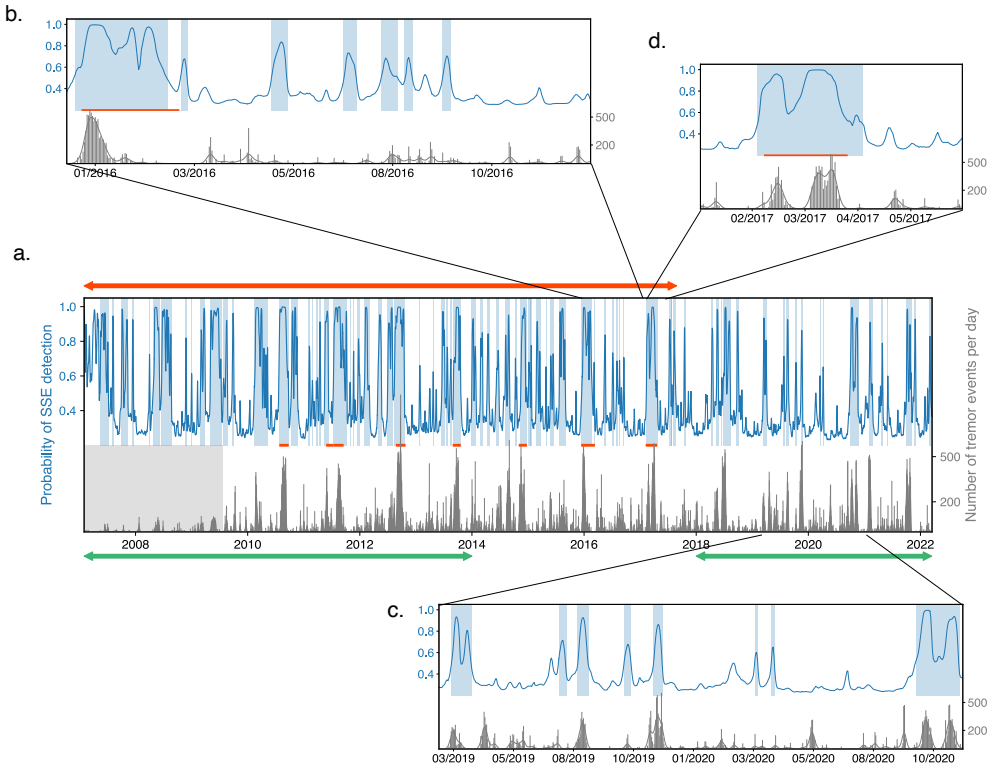


Fig. B.9.: Performance of SSEdetector on real raw GNSS time series, on a subset of stations in north Cascadia (cf. Methods). For a detailed explanation of the figure, refer to Figure 3 in the main text.

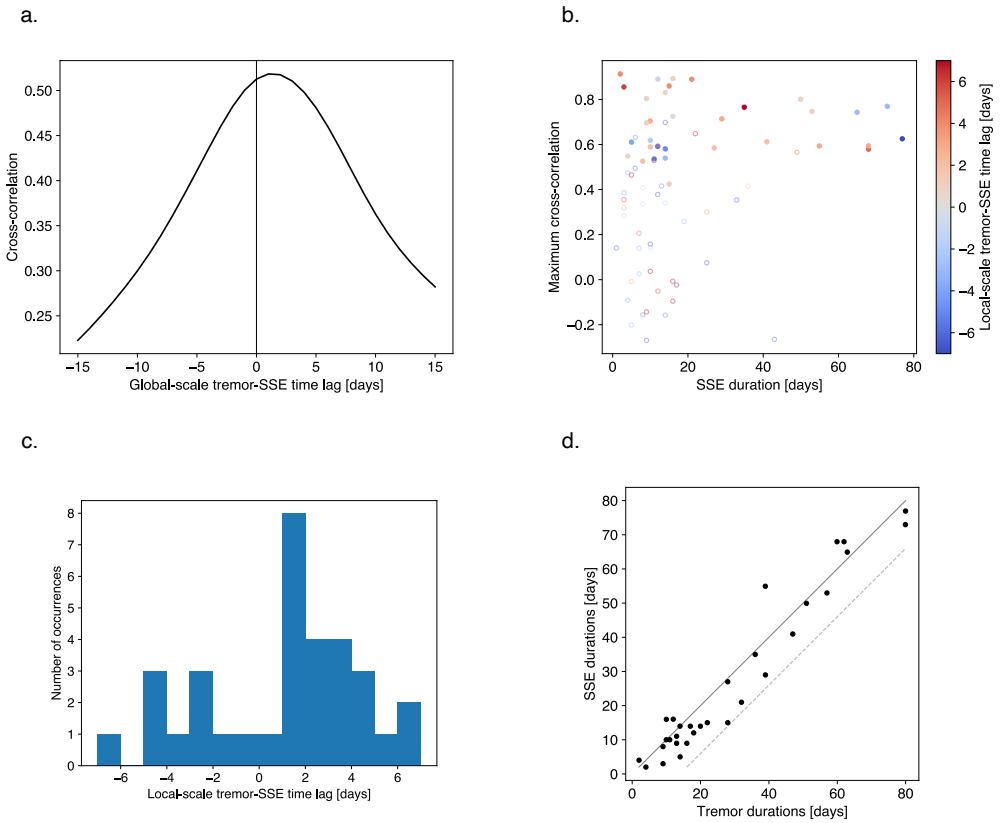


Fig. B.10.: Validation of SSEdetector performance against tremor activity in 2010-2022, from a subset of GNSS stations in north Cascadia. For a detailed explanation of the figure, refer to Figure 5 in the main text. The maximum global correlation value is 0.52 (compared to 0.58 obtained for the whole set of 135 stations), with the same global lag value of 2 days. At a local scale, we found an average lag of 2 days (deformation precedes tremors), although it is less clear-cut than in Figure 5.

Sensitivity study: train and test with a 2D-CNN

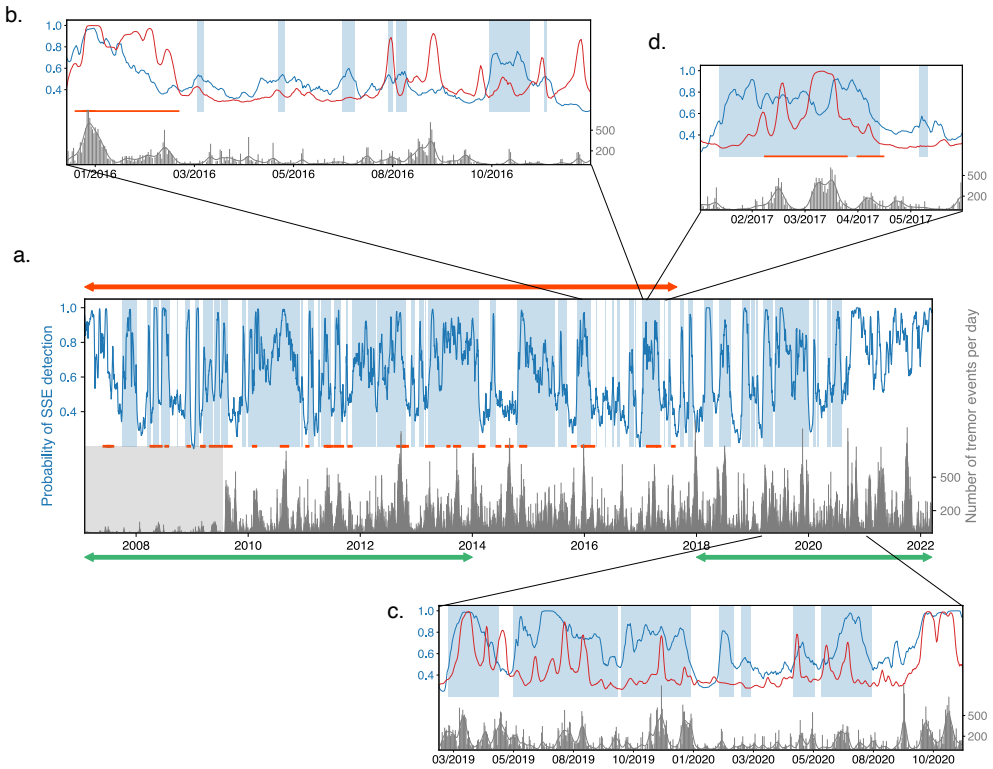


Fig. B.11.: Performance of a 2D-CNN on real raw GNSS time series (cf. section "Sensitivity study"). For a detailed explanation of the figure, refer to Figure 3 in the main text. The red curve in the (b), (c) and (d) panels shows the probability curve of (the preferred) SSEdetector (see Figure 3), with the blue curve associated with the 2D-CNN.

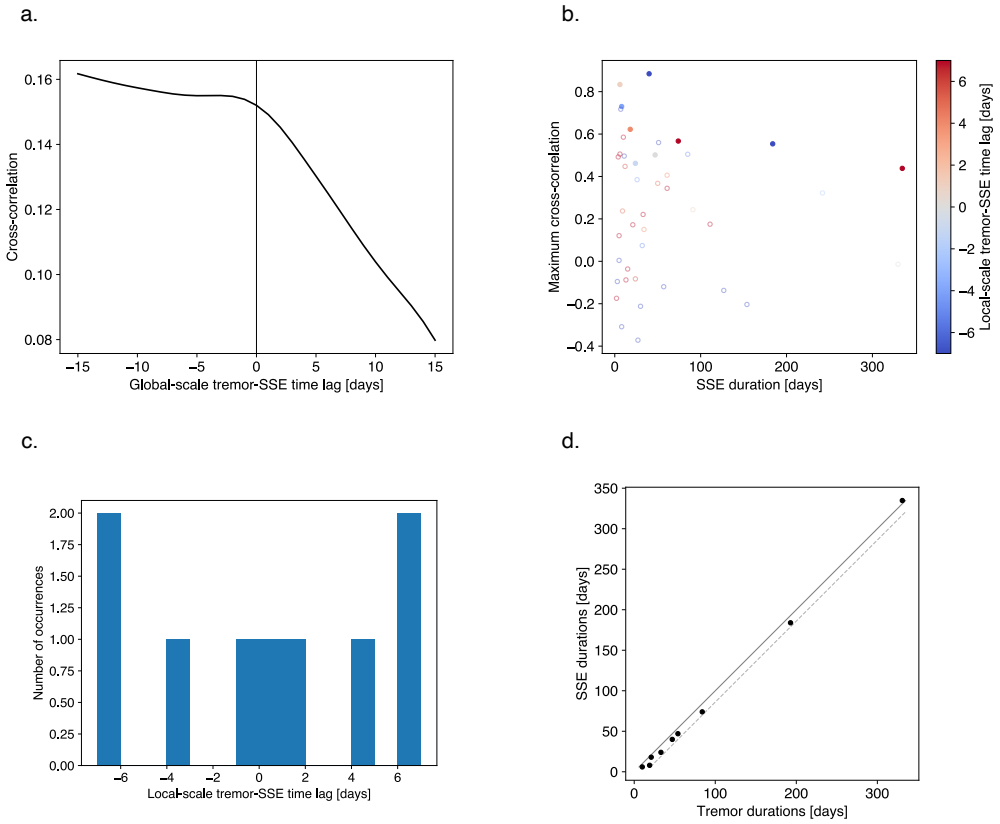


Fig. B.12.: Validation of the performance of a 2D-CNN against tremor activity in 2010-2022.. For a detailed explanation of the figure, refer to Figure 5 in the main text. The maximum global correlation value is very low, therefore the results cannot be analyzed in detail.

Methods: architecture of SSEdetector

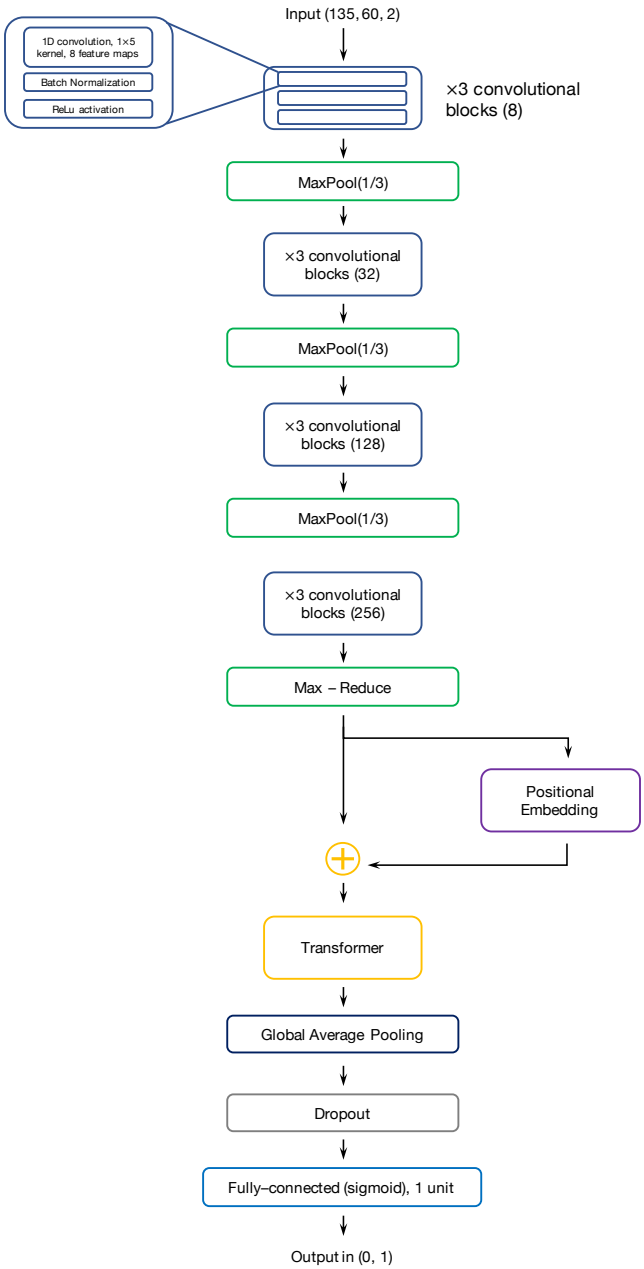


Fig. B.13.: Detailed architecture of SSEdetector.

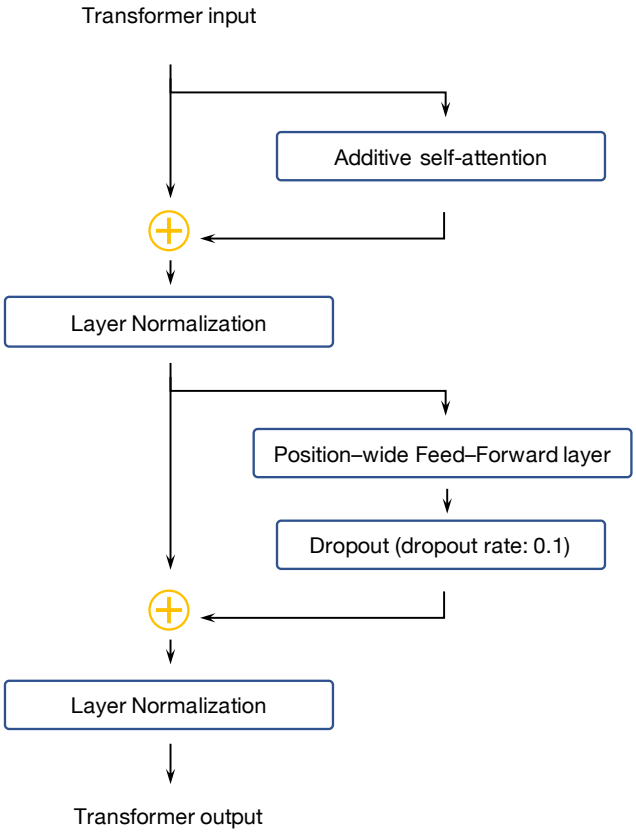


Fig. B.14.: The transformer layer architecture.

Methods: computation of tremor duration

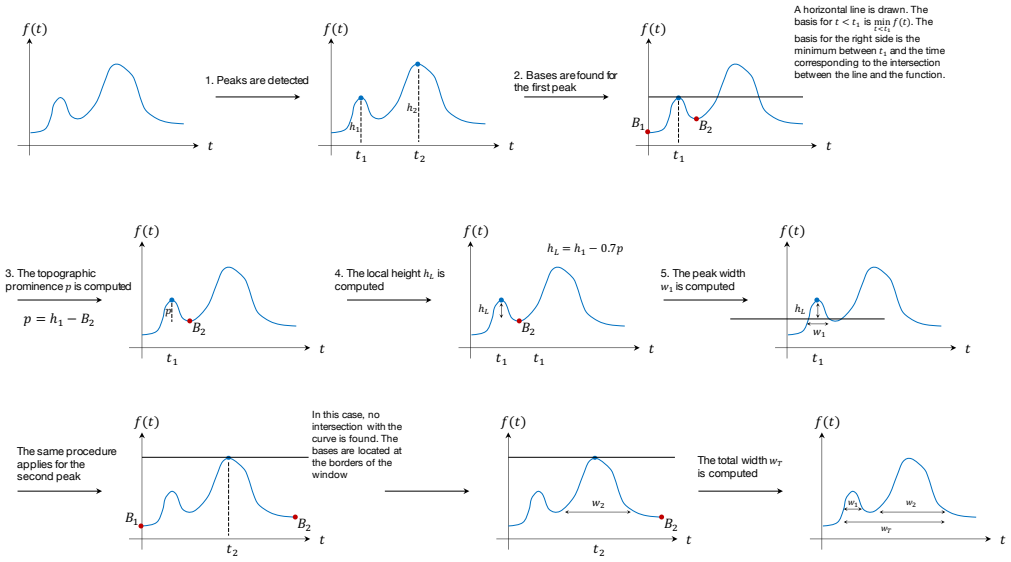


Fig. B.15.: The tremor duration calculation procedure.

Further sensitivity study: test with a subset of GNSS stations in the southern part of Cascadia

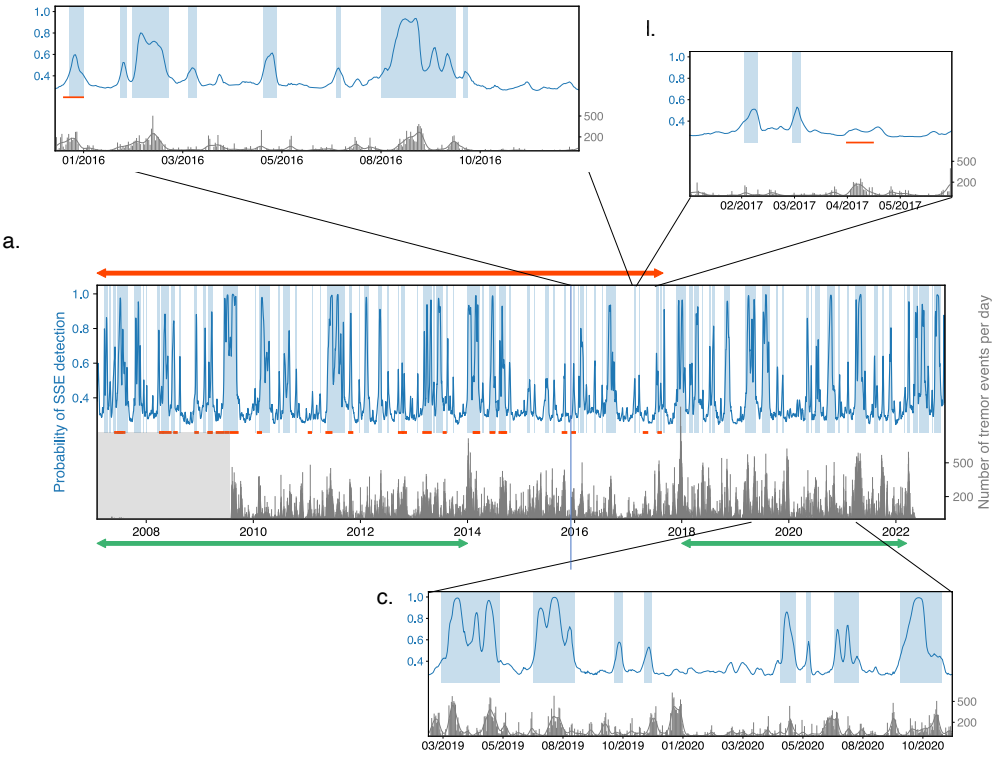


Fig. B.16.: Performance of SSEdetector on real raw GNSS time series, on a subset of stations in south Cascadia (empty triangles in Supplementary Figure 8). For a detailed explanation of the figure, refer to Figure 3 in the main text.

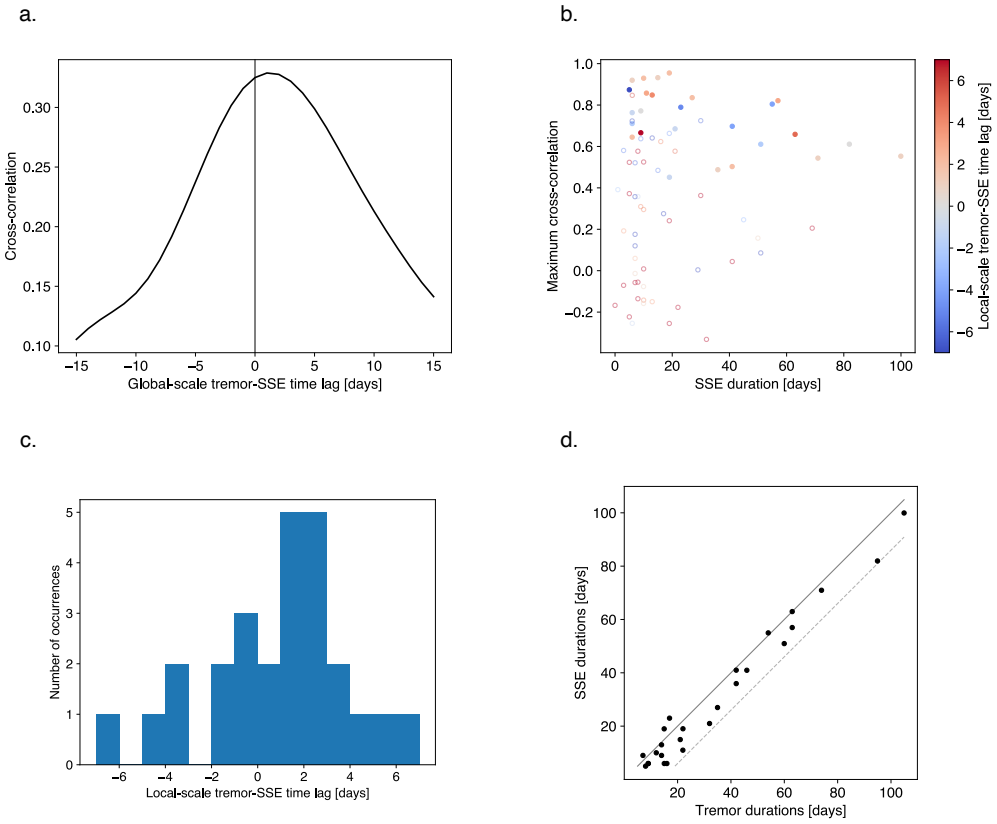


Fig. B.17.: Validation of SSEdetector performance against tremor activity in 2010-2022, from a subset of GNSS stations in south Cascadia (empty triangles in Supplementary Figure 8). For a detailed explanation of the figure, refer to Figure 5 in the main text. The maximum global correlation value is 0.33 (compared to 0.58 obtained for the whole set of 135 stations), with the same global lag value of 2 days. At a local scale, we found an average lag of 2 days (deformation precedes tremors), although the statistical significance is lower than the preferred model.

Further sensitivity study: test with a detection threshold of 0.4

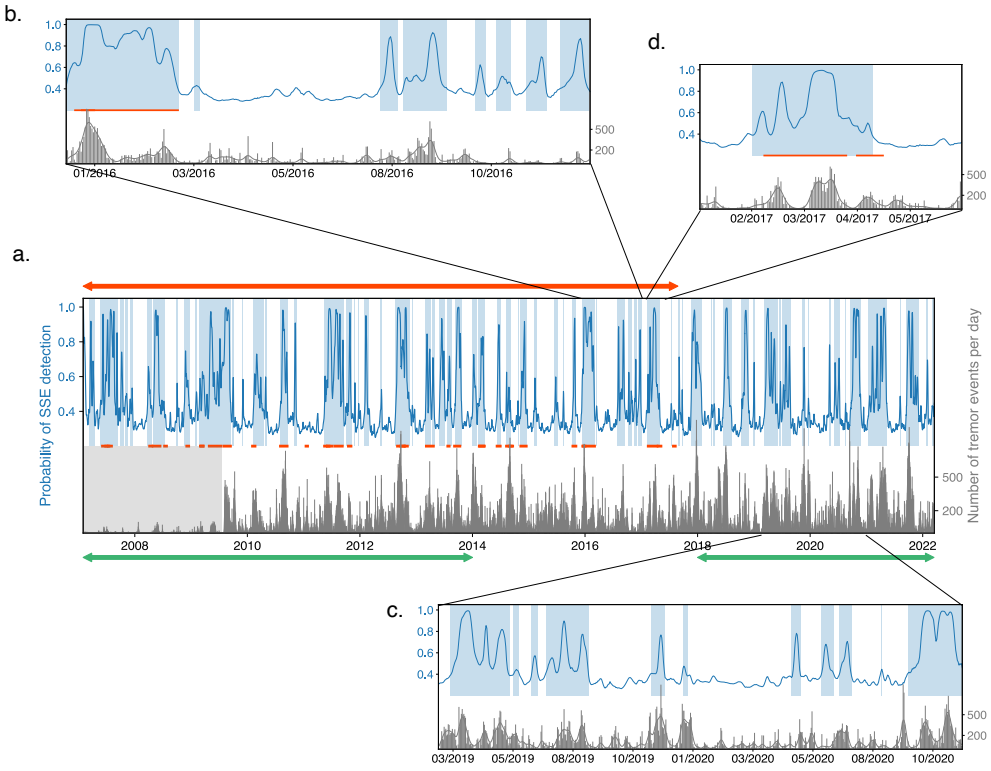


Fig. B.18.: Performance of SSEdetector on raw GNSS time series with a detection threshold of 0.4. For a detailed explanation of the figure, refer to Figure 3 in the main text.

Analysis of SSEdetector's robustness to variation in the source parameters

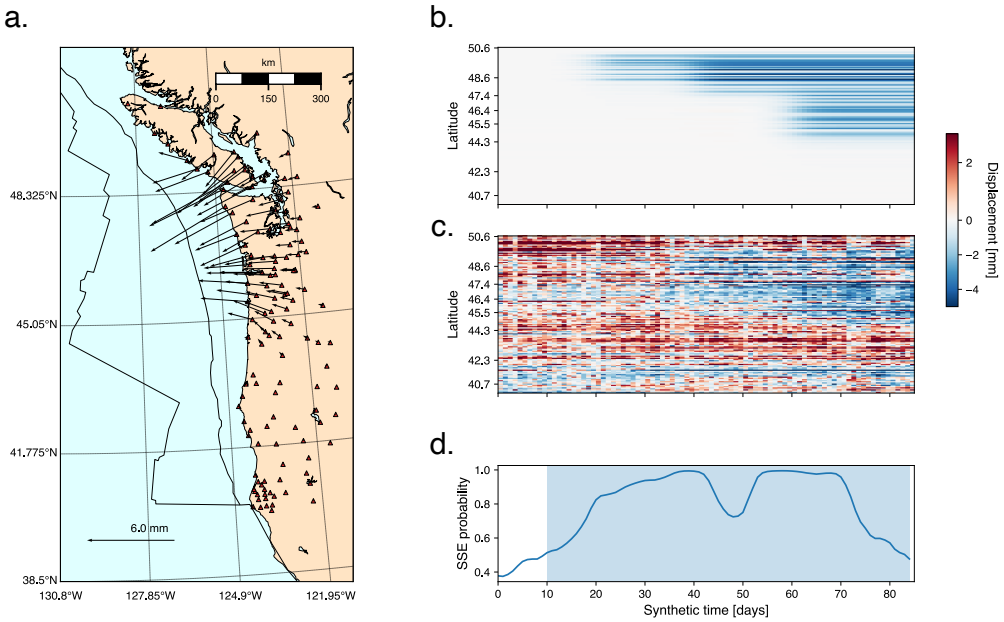


Fig. B.19.: Performance of SSEdetector on a synthetic test case, containing three 20-day SSEs simulating a propagating 60-day event. (a) map showing the (total) displacement field associated with the event (black arrows). (b) modeled time series for the synthetic event. Each row of the matrix represents the displacement at each station, color-coded by GNSS position value. (c) Synthetic time series built by adding the synthetic SSE, in the (b) panel, to the synthetic noise, produced by SSEgenerator. Each row of the matrix represents the detrended position time series, color-coded by position. (d) Probability of detection output by SSEdetector from the synthetic test time series. The blue band represents the inferred proxy of duration (probability exceeding 0.5).

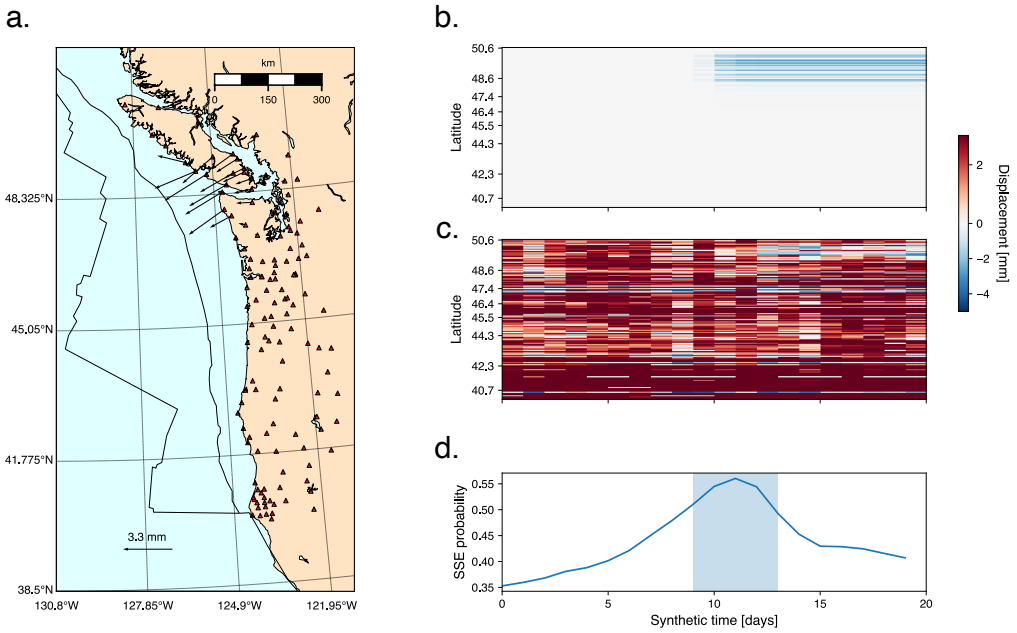


Fig. B.20.: Performance of SSEdetector on a synthetic test case, containing a 3-day SSE. (a) map showing the displacement field associated with the event (black arrows). (b) modeled time series for the synthetic event. Each row of the matrix represents the displacement at each station, color-coded by GNSS position value. (c) Synthetic time series built by adding the synthetic SSE, in the (b) panel, to the synthetic noise, produced by SSEgenerator. Each row of the matrix represents the detrended position time series, color-coded by position. (d) Probability of detection output by SSEdetector from the synthetic test time series. The blue band represents the inferred proxy of duration (probability exceeding 0.5).

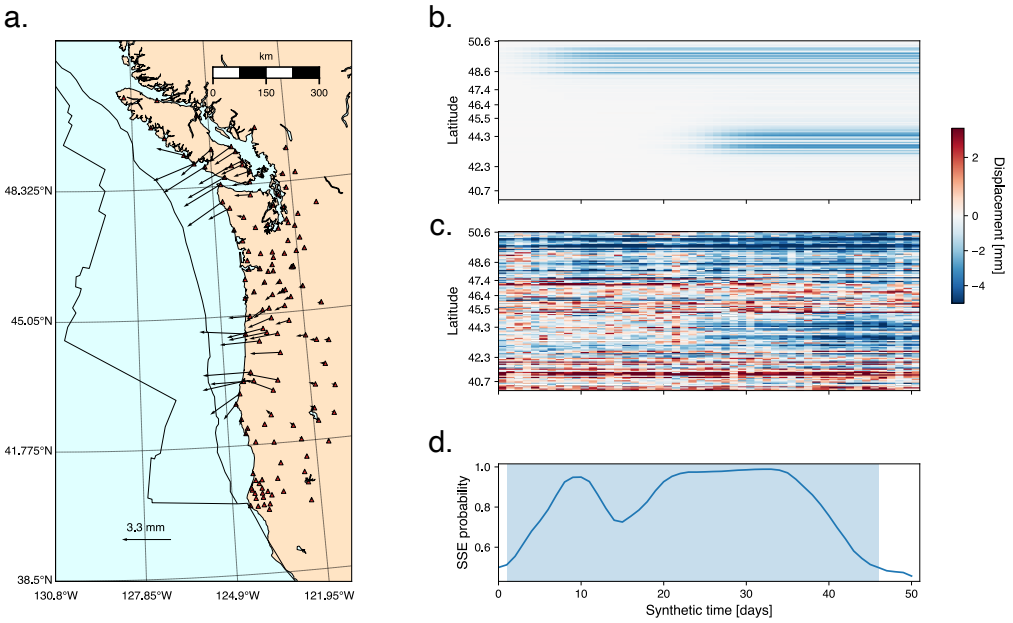


Fig. B.21.: Performance of SSEdetector on a synthetic test case, containing two 20-day SSEs in the same window. (a) map showing the (total) displacement field associated with the events (black arrows). (b) modeled time series for the synthetic events. Each row of the matrix represents the displacement at each station, color-coded by GNSS position value. (c) Synthetic time series built by adding the synthetic SSEs, in the (b) panel, to the synthetic noise, produced by SSEgenerator. Each row of the matrix represents the detrended position time series, color-coded by position. (d) Probability of detection output by SSEdetector from the synthetic test time series. The blue band represents the inferred proxy of duration (probability exceeding 0.5).

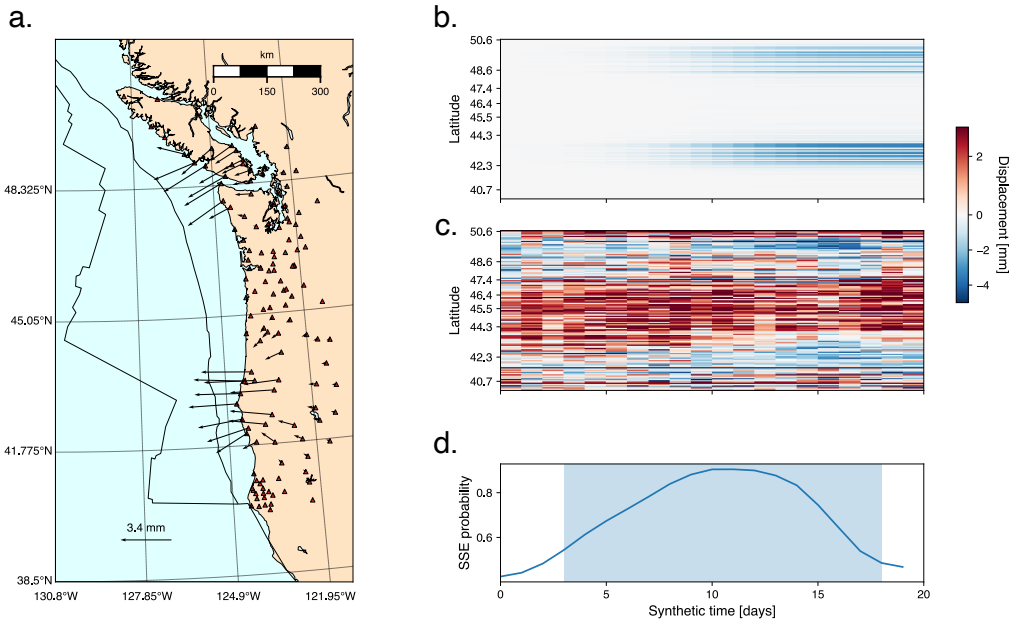


Fig. B.22.: Performance of SSEdetector on a synthetic test case, containing two 20-day SSEs in the same window occurring at the same time. (a) map showing the (total) displacement field associated with the events (black arrows). (b) modeled time series for the synthetic events. Each row of the matrix represents the displacement at each station, color-coded by GNSS position value. (c) Synthetic time series built by adding the synthetic SSEs, in the (b) panel, to the synthetic noise, produced by SSEgenerator. Each row of the matrix represents the detrended position time series, color-coded by position. (d) Probability of detection output by SSEdetector from the synthetic test time series. The blue band represents the inferred proxy of duration (probability exceeding 0.5).

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Abraha, K. E., Teferle, F. N., Hunegnaw, A., & Dach, R. (2017). Gnss related periodic signals in coordinate time-series from precise point positioning. *Geophysical journal international*, *208*(3), 1449–1464.

Aden-Antoniów, F., Frank, W., & Seydoux, L. (2022). An adaptable random forest model for the declustering of earthquake catalogs. *Journal of Geophysical Research: Solid Earth*, *127*(2), e2021JB023254.

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

Aguiar, A. C., Melbourne, T. I., & Scrivner, C. W. (2009). Moment release rate of cascadia tremor constrained by gps. *Journal of Geophysical Research: Solid Earth*, *114*(B7).

Aki, K., & Richards, P. G. (2002). *Quantitative seismology*.

Ammon, C. J., Ji, C., Thio, H.-K., Robinson, D., Ni, S., Hjorleifsdottir, V., Kanamori, H., Lay, T., Das, S., Helmberger, D., et al. (2005). Rupture process of the 2004 sumatra-andaman earthquake. *science*, *308*(5725), 1133–1139.

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.

Anantrasirichai, N., Biggs, J., Albino, F., Hill, P., & Bull, D. (2018). Application of machine learning to classification of volcanic deformation in routinely generated insar data. *Journal of Geophysical Research: Solid Earth*, *123*(8), 6592–6606.

Asano, Y., Saito, T., Ito, Y., Shiomi, K., Hirose, H., Matsumoto, T., Aoi, S., Hori, S., & Sekiguchi, S. (2011). Spatial distribution and focal mechanisms of aftershocks of the 2011 off the pacific coast of tohoku earthquake. *Earth, planets and space*, *63*(7), 669–673.

Asim, K., Martínez-Álvarez, F., Basit, A., & Iqbal, T. (2017). Earthquake magnitude prediction in hindukush region using machine learning techniques. *Natural Hazards*, *85*, 471–486.

Audet, P., Bostock, M. G., Christensen, N. I., & Peacock, S. M. (2009). Seismic evidence for overpressured subducted oceanic crust and megathrust fault sealing. *Nature*, *457*(7225), 76–78.

Avouac, J.-P. (2015). From geodetic imaging of seismic and aseismic fault slip to dynamic modeling of the seismic cycle. *Annual Review of Earth and Planetary Sciences*, *43*, 233–271.

Baba, T., Tanioka, Y., Cummins, P. R., & Uhira, K. (2002). The slip distribution of the 1946 nankai earthquake estimated from tsunami inversion using a new plate model. *Physics of the Earth and Planetary Interiors*, *132*(1-3), 59–73.

Bai, L., et al. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*.

Bartlow, N. M. (2020). A long-term view of episodic tremor and slip in cascadia. *Geophysical Research Letters*, *47*(3), e2019GL085303.

Bartlow, N. M., Miyazaki, S., Bradley, A. M., & Segall, P. (2011). Space-time correlation of slip and tremor during the 2009 cascadia slow slip event. *Geophysical Research Letters*, *38*. https://doi.org/10.1029/2011GL048714

Bartlow, N. M., Wallace, L. M., Beavan, R. J., Bannister, S., & Segall, P. (2014). Time-dependent modeling of slow slip events and associated seismicity and tremor at the hikurangi subduction zone, new zealand. *Journal of Geophysical Research: Solid Earth*, *119*(1), 734–753.

Bedford, J., & Bevis, M. (2018). Greedy automatic signal decomposition and its application to daily gps time series. *Journal of Geophysical Research: Solid Earth*, *123*(8), 6992–7003.

Behr, W. M., & Bürgmann, R. (2021). What's down there? the structures, materials and environment of deep-seated slow slip and tremor. *Philosophical Transactions of the Royal Society A*, *379*(2193), 20200218.

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).

Beroza, G. C., & Ide, S. (2011). Slow earthquakes and nonvolcanic tremor. *Annual review of Earth and planetary sciences*, *39*, 271–296.

Bevis, M., & Brown, A. (2014). Trajectory models and reference frames for crustal motion geodesy. *Journal of Geodesy*, *88*(3), 283–311.

Bilham, R. (1989). Surface slip subsequent to the 24 november 1987 superstition hills, california, earthquake monitored by digital creepmeters. *Bulletin of the Seismological Society of America*, *79*(2), 424–450.

Birant, D., & Kut, A. (2007). St-dbscan: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering*, *60*(1), 208–221.

Bletery, Q., & Nocquet, J.-M. (2020). Slip bursts during coalescence of slow slip events in cascadia. *Nature communications*, *11*(1), 1–6.

Bletery, Q., Thomas, A. M., Hawthorne, J. C., Skarbek, R. M., Rempel, A. W., & Krogstad, R. D. (2017). Characteristics of secondary slip fronts associated with slow earthquakes in cascadia. *Earth and Planetary Science Letters*, *463*, 212–220.

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*, 335–343.

Bock, Y., & Melgar, D. (2016). Physical applications of gps geodesy: A review. *Reports on Progress in Physics*, *79*(10), 106801.

Bolton, D. C., Shokouhi, P., Rouet-Leduc, B., Hulbert, C., Riviére, J., Marone, C., & Johnson, P. A. (2019). Characterizing acoustic signals and searching for precursors during the laboratory seismic cycle using unsupervised machine learning. *Seismological Research Letters*, *90*(3), 1088–1098.

Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Review*, *60*(2), 223–311.

Bouchon, M., Durand, V., Marsan, D., Karabulut, H., & Schmittbuhl, J. (2013). The long precursory phase of most large interplate earthquakes. *Nature geoscience*, *6*(4), 299–302.

Bouchon, M., Marsan, D., Durand, V., Campillo, M., Perfettini, H., Madariaga, R., & Gardonio, B. (2016). Potential slab deformation and plunge prior to the tohoku, iquique and maule earthquakes. *Nature Geoscience*, *9*(5), 380–383.

Bowman, D. D., & King, G. C. (2001). Accelerating seismicity and stress accumulation before large earthquakes. *Geophysical Research Letters*, *28*(21), 4039–4042.

Brace, W., & Byerlee, J. (1966). Stick-slip as a mechanism for earthquakes. *Science*, *153*(3739), 990–992.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Brown, J. R., Beroza, G. C., Ide, S., Ohta, K., Shelly, D. R., Schwartz, S. Y., Rabbel, W., Thorwart, M., & Kao, H. (2009). Deep low-frequency earthquakes in tremor localize to the plate interface in multiple subduction zones. *Geophysical Research Letters*, *36*(19).

Brudzinski, M. R., & Allen, R. M. (2007). Segmentation in episodic tremor and slip all along cascadia. *Geology*, *35*(10), 907–910.

Bürgmann, R. (2018). The geophysics, geology and mechanics of slow fault slip. *Earth and Planetary Science Letters*, *495*, 112–134.

Calvert, A. J., Bostock, M. G., Savard, G., & Unsworth, M. J. (2020). Cascadia low frequency earthquakes at the base of an overpressured subduction shear zone. *Nature communications*, *11*(1), 3874.

Cattania, C., Rivalta, E., Hainzl, S., Passarelli, L., & Aoki, Y. (2017). A nonplanar slow rupture episode during the 2000 miyakejima dike intrusion. *Journal of Geophysical Research: Solid Earth*, *122*(3), 2054–2068.

Chaudhuri, K., & Ghosh, A. (2022). Widespread very low frequency earthquakes (vlfes) activity offshore cascadia. *Geophysical Research Letters*, *49*(13), e2022GL097962.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Chlieh, M., Avouac, J.-P., Sieh, K., Natawidjaja, D. H., & Galetzka, J. (2008). Heterogeneous coupling of the sumatran megathrust constrained by geodetic and paleogeodetic measurements. *Journal of Geophysical Research: Solid Earth*, *113*(B5).

Chollet, F., et al. (2015). *Keras*. https://github.com/fchollet/keras

Chollet, F. (2021). *Deep learning with python*. Simon; Schuster.

Costantino, G., et al. (2023a). Slow slip detection with deep learning in multi-station raw geodetic time series validated against tremors in cascadia. *arXiv preprint arXiv:2305.19720*. https://doi.org/https://doi.org/10.48550/arXiv.2305.19720

Costantino, G., Giffard-Roisin, S., Marsan, D., Marill, L., Radiguet, M., Mura, M. D., Janex, G., & Socquet, A. (2023b). Seismic source characterization from gnss data using deep learning. *Journal of Geophysical Research: Solid Earth*, *128*(4), e2022JB024930. https://doi.org/https://doi.org/10.1029/2022JB024930

Dal Zilio, L., Lapusta, N., & Avouac, J.-P. (2020). Unraveling scaling properties of slow-slip events. *Geophysical Research Letters*, *47*(10), e2020GL087477.

De Castro, L. N. (2006). *Fundamentals of natural computing: Basic concepts, algorithms, and applications*. CRC Press.

Delouis, B., Nocquet, J.-M., & Vallée, M. (2010). Slip distribution of the february 27, 2010 mw= 8.8 maule earthquake, central chile, from static and high-rate gps, insar, and broadband teleseismic data. *Geophysical Research Letters*, *37*(17).

DeMets, C., & Dixon, T. H. (1999). New kinematic models for pacific-north america motion from 3 ma to present, i: Evidence for steady motion and biases in the nuvel-1a model. *Geophysical Research Letters*, *26*(13), 1921–1924.

DeMets, C., Gordon, R. G., & Argus, D. F. (2010). Geologically current plate motions. *Geophysical journal international*, *181*(1), 1–80.

Dodge, D. A., Beroza, G. C., & Ellsworth, W. (1996). Detailed observations of california foreshock sequences: Implications for the earthquake initiation process. *Journal of Geophysical Research: Solid Earth*, *101*(B10), 22371–22392.

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: Solid Earth*, *107*(B4), ETG–9.

Douglas, A., Beavan, J., Wallace, L., & Townend, J. (2005). Slow slip on the northern hikurangi subduction interface, new zealand. *Geophysical Research Letters*, *32*(16).

Dragert, H., Wang, K., & James, T. S. (2001a). A silent slip event on the deeper cascadia subduction interface. *Science*, *292*, 1525–1528. https://doi.org/10.1126/SCIENCE.1060152/ASSET/DD167B7B-24D4-40A4-996F-4603D42C0244/ASSETS/GRAPHIC/SE1919443004.JPEG

Dragert, H., Wang, K., & Rogers, G. (2004). Geodetic and seismic signatures of episodic tremor and slip in the northern cascadia subduction zone. *Earth, planets and space*, *56*(12), 1143–1150.

Dragert, H., Wang, K., & James, T. S. (2001b). A silent slip event on the deeper cascadia subduction interface. *Science*, *292*(5521), 1525–1528.

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: Solid Earth*, *86*(B4), 2825–2852.

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.

Feng, L., Hill, E. M., Banerjee, P., Hermawan, I., Tsang, L. L., Natawidjaja, D. H., Suwargadi, B. W., & Sieh, K. (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.

Frank, W. B. (2016). Slow slip hidden in the noise: The intermittence of tectonic release. *Geophysical Research Letters*, *43*(19), 10–125.

Frank, W. B., & Brodsky, E. E. (2019). Daily measurement of slow slip from low-frequency earthquakes is consistent with ordinary earthquake scaling. *Science advances*, *5*(10), eaaw9386.

Frank, W. B., Radiguet, M., Rousset, B., Shapiro, N. M., Husker, A. L., Kostoglodov, V., Cotte, N., & Campillo, M. (2015). Uncovering the geodetic signature of silent slip through repeating earthquakes. *Geophysical Research Letters*, *42*, 2774–2779. https://doi.org/10.1002/2015GL063685

Frank, W. B., Shapiro, N. M., Husker, A. L., Kostoglodov, V., Bhat, H. S., & Campillo, M. (2015). Along-fault pore-pressure evolution during a slow-slip event in guerrero, mexico. *Earth and Planetary Science Letters*, *413*, 135–143.

Fukuda, J. (2018). Variability of the space-time evolution of slow slip events off the boso peninsula, central japan, from 1996 to 2014. *Journal of Geophysical Research: Solid Earth*, *123*(1), 732–760.

Gao, H., Schmidt, D. A., & Weldon, R. J. (2012). Scaling relationships of source parameters for slow slip events. *Bulletin of the Seismological Society of America*, *102*(1), 352–360.

Gao, X., & Wang, K. (2017). Rheological separation of the megathrust seismogenic zone and episodic tremor and slip. *Nature*, *543*(7645), 416–419.

Ghimire, S., Guéguen, P., Giffard-Roisin, S., & Schorlemmer, D. (2022). Testing machine learning models for seismic damage prediction at a regional scale using building-damage dataset compiled after the 2015 gorkha nepal earthquake. *Earthquake Spectra*, *38*(4), 2970–2993.

Ghosh, A., Huesca-Pérez, E., Brodsky, E., & Ito, Y. (2015). Very low frequency earthquakes in cascadia migrate with tremor. *Geophysical Research Letters*, *42*(9), 3228–3232.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

gnss-products. (2019). Gnss position solutions in japan. "*CNRS, OSUG, ISTERRE*". https://doi.org/10.17178/GNSS.products.Japan

Goebel, T. H., Kwiatek, G., Becker, T. W., Brodsky, E. E., & Dresen, G. (2017). What allows seismic events to grow big?: Insights from b-value and fault roughness analysis in laboratory stick-slip experiments. *Geology*, *45*(9), 815–818.

Goldfinger, C., Galer, S., Beeson, J., Hamilton, T., Black, B., Romsos, C., Patton, J., Nelson, C. H., Hausmann, R., & Morey, A. (2017). The importance of site selection, sediment supply, and hydrodynamics: A case study of submarine paleoseismology on the northern cascadia margin, washington usa. *Marine Geology*, *384*, 4–46.

Goldfinger, C., Nelson, C. H., Morey, A. E., Johnson, J. E., Patton, J. R., Karabanov, E. B., Gutierrez-Pastor, J., Eriksson, A. T., Gracia, E., Dunhill, G., et al. (2012). *Turbidite event history—methods and implications for holocene paleoseismicity of the cascadia subduction zone* (tech. rep.). US Geological Survey.

Gomberg, J., Wech, A., Creager, K., Obara, K., & Agnew, D. (2016a). Reconsidering earthquake scaling. *Geophysical Research Letters*, *43*, 6243–6251. https://doi.org/10.1002/2016GL069967

Gomberg, J., Wech, A., Creager, K., Obara, K., & Agnew, D. (2016b). Reconsidering earthquake scaling. *Geophysical Research Letters*, *43*(12), 6243–6251.

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*, 1970–1981.

Gutenberg, B. (1956). The energy of earthquakes. *Quarterly Journal of the Geological Society*, *112*(1-4), 1–14.

Hanks, T. C., & Kanamori, H. (1979). A moment magnitude scale. *Journal of Geophysical Research: Solid Earth*, *84*(B5), 2348–2350.

Hawthorne, J. C., & Bartlow, N. M. (2018). Observing and modeling the spectrum of a slow slip event. *Journal of Geophysical Research: Solid Earth*, *123*, 4243–4265. https://doi.org/10.1029/2017JB015124

Hawthorne, J. C., & Rubin, A. M. (2013). Tidal modulation and back-propagating fronts in slow slip events simulated with a velocity-weakening to velocity-strengthening friction law. *Journal of Geophysical Research: Solid Earth*, *118*(3), 1216–1239.

Hayes, G. P., Moore, G. L., Portner, D. E., Hearne, M., Flamme, H., Furtney, M., & Smoczyk, G. M. (2018). Slab2, a comprehensive subduction zone geometry model. *Science*, *362*(6410), 58–61.

Haykin, S. (1998). *Neural networks: A comprehensive foundation*. Prentice Hall PTR.

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.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Heki, K., Miyazaki, S., & Tsuji, H. (1997). Silent fault slip following an interplate thrust earthquake at the japan trench. *Nature*, *386*(6625), 595–598.

Hibert, C., Provost, F., Malet, J.-P., Maggi, A., Stumpf, A., & Ferrazzini, V. (2017). Automatic identification of rockfalls and volcano-tectonic earthquakes at the piton de la fournaise volcano using a random forest algorithm. *Journal of Volcanology and Geothermal Research*, *340*, 130–142.

Hirose, H., Hirahara, K., Kimata, F., Fujii, N., & Miyazaki, S. (1999). A slow thrust slip event following the two 1996 hyuganada earthquakes beneath the bungo channel, southwest japan. *Geophysical Research Letters*, *26*(21), 3237–3240.

Hirose, H., Matsushima, T., Tabei, T., & Nishimura, T. (2023). Long-term slow slip events with and without tremor activation in the bungo channel and hyuganada, southwest japan. *Earth, Planets and Space*, *75*(1), 77.

Hirose, H., Matsuzawa, T., Kimura, T., & Kimura, H. (2014). The boso slow slip events in 2007 and 2011 as a driving process for the accompanying earthquake swarm. *Geophysical Research Letters*, *41*(8), 2778–2785.

Hooper, A., Pietrzak, J., Simons, W., Cui, H., Riva, R., Naeije, M., van Scheltinga, A. T., Schrama, E., Stelling, G., & Socquet, A. (2013). Importance of horizontal seafloor motion on tsunami height for the 2011 mw= 9.0 tohoku-oki earthquake. *Earth and Planetary Science Letters*, *361*, 469–479.

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.

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.

Hyndman, R. D., Yamano, M., & Oleskevich, D. A. (1997). The seismogenic zone of subduction thrust faults. *Island Arc*, *6*(3), 244–260.

Ide, S. (2012). Variety and spatial heterogeneity of tectonic tremor worldwide. *Journal of Geophysical Research: Solid Earth*, *117*(B3).

Ide, S., Beroza, G. C., Shelly, D. R., & Uchide, T. (2007a). A scaling law for slow earthquakes. *Nature*, *447*(7140), 76–79.

Ide, S., Beroza, G. C., Shelly, D. R., & Uchide, T. (2007b). A scaling law for slow earthquakes. *Nature 2007 447:7140*, *447*, 76–79. https://doi.org/10.1038/nature05780

Ide, S., Shelly, D. R., & Beroza, G. C. (2007). Mechanism of deep low frequency earthquakes: Further evidence that deep non-volcanic tremor is generated by shear slip on the plate interface. *Geophysical Research Letters*, *34*(3).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 448–456.

Ishii, M., Shearer, P. M., Houston, H., & Vidale, J. E. (2005). Extent, duration and speed of the 2004 sumatra–andaman earthquake imaged by the hi-net array. *Nature*, *435*(7044), 933–936.

Ito, A., Fujie, G., Tsuru, T., Kodaira, S., Nakanishi, A., & Kaneda, Y. (2004). Fault plane geometry in the source region of the 1994 sanriku-oki earthquake. *Earth and Planetary Science Letters*, *223*(1-2), 163–175.

Ito, Y., Hino, R., Kido, M., Fujimoto, H., Osada, Y., Inazu, D., Ohta, Y., Iinuma, T., Ohzono, M., Miura, S., Mishina, M., Suzuki, K., Tsuji, T., & Ashi, J. (2013). Episodic slow slip events in the Japan subduction zone before the 2011 Tohoku-Oki earthquake. *Tectonophysics*, *600*, 14–26. https://doi.org/10.1016/j.tecto.2012.08.022

Itoh, Y., Aoki, Y., & Fukuda, J. (2022a). Imaging evolution of cascadia slow-slip event using high-rate gps. *Scientific reports*, *12*(1), 1–12.

Itoh, Y., Aoki, Y., & Fukuda, J. (2022b). Imaging evolution of cascadia slow-slip event using high-rate gps. *Scientific Reports 2022 12:1*, *12*, 1–12. https://doi.org/10.1038/s41598-022-10957-8

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.

Jiang, Y., Wdowinski, S., Dixon, T. H., Hackl, M., Protti, M., & Gonzalez, V. (2012). Slow slip events in costa rica detected by continuous gps observations, 2002–2011. *Geochemistry, Geophysics, Geosystems*, *13*(4).

Jolivet, R., & Frank, W. (2020). The transient and intermittent nature of slow slip. *AGU Advances*, *1*(1), e2019AV000126.

Jolivet, R., Lasserre, C., Doin, M.-P., Peltzer, G., Avouac, J.-P., Sun, J., & Dailu, R. (2013). Spatio-temporal evolution of aseismic slip along the haiyuan fault, china: Implications for fault frictional properties. *Earth and Planetary Science Letters*, *377*, 23–33.

Kan, L.-Y., Chevrot, S., & Monteiller, V. (2023). Dehydration of the subducting juan de fuca plate and fluid pathways revealed by full waveform inversion of teleseismic p and sh waves in central oregon. *Journal of Geophysical Research: Solid Earth*, *128*(4), e2022JB025506.

Kanamori, H., & Anderson, D. L. (1975). Theoretical basis of some empirical relations in seismology. *Bulletin of the seismological society of America*, *65*(5), 1073–1095.

Kao, H., Shan, S.-J., Dragert, H., & Rogers, G. (2009). Northern cascadia episodic tremor and slip: A decade of tremor observations from 1997 to 2007. *Journal of Geophysical Research: Solid Earth*, *114*(B11).

King, G., Bilham, R., Campbell, J., McKenzie, D., & Niazi, M. (1975). Detection of elastic strainfields caused by fault creep events in iran. *Nature*, *253*(5491), 420–423.

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

Komori, J., Shishikura, M., Ando, R., Yokoyama, Y., & Miyairi, Y. (2017). History of the great kanto earthquakes inferred from the ages of holocene marine terraces revealed by a comprehensive drilling survey. *Earth and Planetary Science Letters*, *471*, 74–84.

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

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

Kostoglodov, V., Singh, S. K., Santiago, J. A., Franco, S. I., Larson, K. M., Lowry, A. R., & Bilham, R. (2003). A large silent earthquake in the guerrero seismic gap, mexico. *Geophysical Research Letters*, *30*(15).

La Rocca, M., Creager, K. C., Galluzzo, D., Malone, S., Vidale, J. E., Sweet, J. R., & Wech, A. G. (2009). Cascadia tremor located near plate interface constrained by s minus p wave times. *Science*, *323*(5914), 620–623.

Lara, P. E. E., et al. (2020). Automatic multichannel volcano-seismic classification using machine learning and emd. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Larson, K. M., Lowry, A. R., Kostoglodov, V., Hutton, W., Sánchez, O., Hudnut, K., & Suárez, G. (2004). Crustal deformation measurements in guerrero, mexico. *Journal of Geophysical Research: Solid Earth*, *109*(B4).

Lawson, A. C., & Reid, H. F. (1908). *The california earthquake of april 18, 1906: Report of the state earthquake investigation commission...* Carnegie institution of Washington.

Lay, T. (2015). The surge of great earthquakes from 2004 to 2014. *Earth and Planetary Science Letters*, *409*, 133–146.

Lay, T. (2018). A review of the rupture characteristics of the 2011 tohoku-oki mw 9.1 earthquake. *Tectonophysics*, *733*, 4–36.

Lay, T., & Kanamori, H. (1981). An asperity model of large earthquake sequences. *Earthquake prediction: An international review*, *4*, 579–592.

Lay, T., & Wallace, T. C. (1995). *Modern global seismology*. Elsevier.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Leonard, L. J., Currie, C. A., Mazzotti, S., & Hyndman, R. D. (2010). Rupture area and displacement of past cascadia great earthquakes from coastal coseismic subsidence. *Bulletin*, *122*(11-12), 2079–2096.

Li, Z., Meier, M.-A., Hauksson, E., Zhan, Z., & Andrews, J. (2018). Machine learning seismic wave discrimination: Application to earthquake early warning. *Geophysical Research Letters*, *45*(10), 4773–4779.

Licciardi, A., Bletery, Q., Rouet-Leduc, B., Ampuero, J.-P., & Juhel, K. (2022). Instantaneous tracking of earthquake growth with elastogravity signals. *Nature*, *606*(7913), 319–324.

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.

Lin, Y.-n. N., Sladen, A., Ortega-Culaciati, F., Simons, M., Avouac, J.-P., Fielding, E. J., Brooks, B. A., Bevis, M., Genrich, J., Rietbrock, A., et al. (2013). Coseismic and postseismic slip associated with the 2010 maule earthquake, chile: Characterizing the arauco peninsula barrier effect. *Journal of Geophysical Research: Solid Earth*, *118*(6), 3142–3159.

Linde, A. T., Gladwin, M. T., Johnston, M. J., Gwyther, R. L., & Bilham, R. G. (1996). A slow earthquake sequence on the san andreas fault. *Nature*, *383*(6595), 65–68.

Liu, Y. (2014). Source scaling relations and along-strike segmentation of slow slip events in a 3-d subduction fault model. *Journal of Geophysical Research: Solid Earth*, *119*(8), 6512–6533.

Lohman, R., & McGuire, J. (2007). Earthquake swarms driven by aseismic creep in the salton trough, california. *Journal of Geophysical Research: Solid Earth*, *112*(B4).

Lorito, S., Romano, F., Atzori, S., Tong, X., Avallone, A., McCloskey, J., Cocco, M., Boschi, E., & Piatanesi, A. (2011). Limited overlap between the seismic gap and coseismic slip of the great 2010 chile earthquake. *Nature Geoscience*, *4*(3), 173–177.

Lowry, A. R., Larson, K. M., Kostoglodov, V., & Bilham, R. (2001a). Transient fault slip in guerrero, southern mexico. *Geophysical Research Letters*, *28*(19), 3753–3756.

Lowry, A. R., Larson, K. M., Kostoglodov, V., & Bilham, R. (2001b). Transient fault slip in guerrero, southern mexico. *Geophysical Research Letters*, *28*, 3753–3756. https://doi.org/10.1029/2001GL013238

Malfante, M., et al. (2018). Machine learning for volcano-seismic signals: Challenges and perspectives. *IEEE Signal Processing Magazine*.

Mangalathu, S., Sun, H., Nweke, C. C., Yi, Z., & Burton, H. V. (2020). Classifying earthquake damage to buildings using machine learning. *Earthquake Spectra*, *36*(1), 183–208.

Mao, A., Harrison, C. G., & Dixon, T. H. (1999). Noise in gps coordinate time series. *Journal of Geophysical Research: Solid Earth*, *104*(B2), 2797–2816.

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.

Marone, C., & Saffer, D. M. (2007). 12. fault friction and the upper transition from seismic to aseismic faulting. In *The seismogenic zone of subduction thrust faults* (pp. 346–369). Columbia University Press.

Marsan, D., Bouchon, M., Gardonio, B., Perfettini, H., Socquet, A., & Enescu, B. (2017). Change in seismicity along the japan trench, 1990–2011, and its relationship with seismic coupling. *Journal of Geophysical Research: Solid Earth*, *122*(6), 4645–4659.

Marsan, D., Reverso, T., Helmstetter, A., & Enescu, B. (2013). Slow slip and aseismic deformation episodes associated with the subducting Pacific plate offshore Japan, revealed by changes in seismicity. *Journal of Geophysical Research: Solid Earth*, *118*, 4900–4909. https://doi.org/10.1002/jgrb.50323

Mavrommatis, A. P., Segall, P., & Johnson, K. M. (2014). A decadal-scale deformation transient prior to the 2011 mw 9.0 tohoku-oki earthquake. *Geophysical Research Letters*, *41*(13), 4486–4494.

McCaffrey, R. (2007). The next great earthquake. *science*, *315*(5819), 1675–1676.

McCaffrey, R., Wallace, L. M., & Beavan, J. (2008). Slow slip and frictional transition at low temperature at the hikurangi subduction zone. *Nature Geoscience*, *1*(5), 316–320.

McKenzie, K., Furlong, K., & Herman, M. (2022). Regional and local patterns of upper-plate deformation in cascadia: The importance of the down-dip extent of locking relative to upper-plate strength contrasts. *Tectonics*, *41*(1), e2021TC007062.

Melnick, D., Moreno, M., Quinteros, J., Baez, J. C., Deng, Z., Li, S., & Oncken, O. (2017). The super-interseismic phase of the megathrust earthquake cycle in chile. *Geophysical Research Letters*, *44*(2), 784–791.

Michel, S., Gualandi, A., & Avouac, J. P. (2019). Similar scaling laws for earthquakes and cascadia slow-slip events. *Nature*, *574*, 522–526. https://doi.org/10.1038/s41586-019-1673-6

Michel, S., Gualandi, A., & Avouac, J.-P. (2019a). Interseismic coupling and slow slip events on the cascadia megathrust. *Pure and Applied Geophysics*, *176*(9), 3867–3891.

Michel, S., Gualandi, A., & Avouac, J.-P. (2019b). Similar scaling laws for earthquakes and cascadia slow-slip events. *Nature*, *574*(7779), 522–526.

Miller, M. M., Melbourne, T., Johnson, D. J., & Sumner, W. Q. (2002). Periodic slow earthquakes from the cascadia subduction zone. *Science*, *295*(5564), 2423–2423.

Miura, S., Ueki, S., Sato, T., Tachibana, K., & Hamaguchi, H. (2000). Crustal deformation associated with the 1998 seismo-volcanic crisis of iwate volcano, northeastern japan, as observed by a dense gps network. *Earth, planets and space*, *52*(11), 1003–1008.

Miyazaki, S., McGuire, J. J., & Segall, P. (2003). A transient subduction zone slip episode in southwest japan observed by the nationwide gps array. *Journal of Geophysical Research: Solid Earth*, *108*(B2).

Moon, T. K., & Stirling, W. C. (2000). *Mathematical methods and algorithms for signal processing*.

Moreno, M., Rosenau, M., & Oncken, O. (2010). 2010 maule earthquake slip correlates with pre-seismic locking of andean subduction zone. *Nature*, *467*(7312), 198–202.

Mousavi, S. M., & Beroza, G. C. (2020). A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*, *47*(1), e2019GL085976.

Mousavi, S. M., & Beroza, G. C. (2022). Deep-learning seismology. *Science*, *377*(6607), eabm4470.

Mousavi, S. M., & Beroza, G. C. (2023). Machine learning in earthquake seismology. *Annual Review of Earth and Planetary Sciences*, *51*, 105–129.

Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C. (2020a). Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature communications*, *11*(1), 1–12.

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

Müller, R. D., Sdrolias, M., Gaina, C., & Roest, W. R. (2008). Age, spreading rates, and spreading asymmetry of the world's ocean crust. *Geochemistry, Geophysics, Geosystems*, *9*(4).

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.

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.

Münchmeyer, J., Woollam, J., Rietbrock, A., Tilmann, F., Lange, D., Bornstein, T., Diehl, T., Giunchi, C., Haslinger, F., Jozinović, D., et al. (2022). Which picker fits my data? a quantitative evaluation of deep learning based seismic pickers. *Journal of Geophysical Research: Solid Earth*, *127*(1), e2021JB023499.

Nishikawa, T., Matsuzawa, T., Ohta, K., Uchida, N., Nishimura, T., & Ide, S. (2019). The slow earthquake spectrum in the Japan Trench illuminated by the S-net seafloor observatories. *Science*, *365*(6455), 808–813. https://doi.org/10.1126/science.aax5618

Nishimura, T. (2014). Short-term slow slip events along the Ryukyu Trench, southwestern Japan, observed by continuous GNSS. *Progress in Earth and Planetary Science*, *1*, 22. https://doi.org/10.1186/s40645-014-0022-5

Nishimura, T. (2021). Slow Slip Events in the Kanto and Tokai Regions of Central Japan Detected Using Global Navigation Satellite System Data During 1994–2020. *Geochemistry, Geophysics, Geosystems*, *22*, e2020GC009329. https://doi.org/10.1029/2020GC009329

Nishimura, T., Matsuzawa, T., & Obara, K. (2013). Detection of short-term slow slip events along the nankai trough, southwest japan, using gnss data. *Journal of Geophysical Research: Solid Earth*, *118*(6), 3112–3125.

Nishimura, T., Sagiya, T., & Stein, R. S. (2007). Crustal block kinematics and seismic potential of the northernmost philippine sea plate and izu microplate, central japan, inferred from gps and leveling data. *Journal of Geophysical Research: Solid Earth*, *112*(B5).

Obana, K., Scherwath, M., Yamamoto, Y., Kodaira, S., Wang, K., Spence, G., Riedel, M., & Kao, H. (2015). Earthquake activity in northern cascadia subduction zone off vancouver island revealed by ocean-bottom seismograph observations. *Bulletin of the Seismological Society of America*, *105*(1), 489–495.

Obara, K. (2002). Nonvolcanic deep tremor associated with subduction in southwest japan. *Science*, *296*(5573), 1679–1681.

Obara, K. (2011). Characteristics and interactions between non-volcanic tremor and related slow earthquakes in the nankai subduction zone, southwest japan. *Journal of Geodynamics*, *52*(3-4), 229–248.

Obara, K. (2020). Characteristic activities of slow earthquakes in japan. *Proceedings of the Japan Academy, Series B*, *96*(7), 297–315.

Obara, K., Hirose, H., Yamamizu, F., & Kasahara, K. (2004). Episodic slow slip events accompanied by non-volcanic tremors in southwest japan subduction zone. *Geophysical Research Letters*, *31*(23).

Obara, K., & Kato, A. (2016). Connecting slow earthquakes to huge earthquakes. *Science*, *353*(6296), 253–257.

Obara, K., Tanaka, S., Maeda, T., & Matsuzawa, T. (2010). Depth-dependent activity of non-volcanic tremor in southwest japan. *Geophysical Research Letters*, *37*(13).

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.

Okada, Y., Nishimura, T., Tabei, T., Matsushima, T., & Hirose, H. (2022). Development of a detection method for short-term slow slip events using gnss data and its application to the nankai subduction zone. *Earth, Planets and Space*, *74*(1), 1–18.

Ozawa, S., Murakami, M., & Tada, T. (2001). Time-dependent inversion study of the slow thrust event in the nankai trough subduction zone, southwestern japan. *Journal of Geophysical Research: Solid Earth*, *106*(B1), 787–802.

Ozawa, S., Nishimura, T., Munekane, H., Suito, H., Kobayashi, T., Tobita, M., & Imakiire, T. (2012). Preceding, coseismic, and postseismic slips of the 2011 tohoku earthquake, japan. *Journal of Geophysical Research: Solid Earth*, *117*(B7).

Ozawa, S., Yarai, H., Imakiire, T., & Tobita, M. (2013). Spatial and temporal evolution of the long-term slow slip in the bungo channel, japan. *Earth, Planets and Space*, *65*(2), 67–73.

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: Solid Earth*, *114*(B1).

Pearson, R. K., Neuvo, Y., Astola, J., & Gabbouj, M. (2016). Generalized hampel filters. *EURASIP Journal on Advances in Signal Processing*, *2016*(1), 1–18.

Peng, Z., & Gomberg, J. (2010). An integrated perspective of the continuum between earthquakes and slow-slip phenomena. *Nature geoscience*, *3*(9), 599–607.

Peterson, C. L., & Christensen, D. H. (2009). Possible relationship between nonvolcanic tremor and the 1998–2001 slow slip event, south central alaska. *Journal of Geophysical Research: Solid Earth*, *114*(B6).

Pétrélis, F., Chanard, K., Schubnel, A., & Hatano, T. (2023). Earthquake magnitude distribution and aftershocks: A statistical geometry explanation. *Physical Review E*, *107*(3), 034132.

Prichard, D., & Theiler, J. (1994). Generating surrogate data for time series with several simultaneously measured variables. *Physical review letters*, *73*(7), 951.

Radiguet, M., Cotton, F., Vergnolle, M., Campillo, M., Walpersdorf, A., Cotte, N., & Kostoglodov, V. (2012a). Slow slip events and strain accumulation in the guerrero gap, mexico. *Journal of Geophysical Research: Solid Earth*, *117*, 4305. https://doi.org/10.1029/2011JB008801

Radiguet, M., Perfettini, H., Cotte, N., Gualandi, A., Valette, B., Kostoglodov, V., Lhomme, T., Walpersdorf, A., Cabral Cano, E., & Campillo, M. (2016). Triggering of the 2014 mw7. 3 papanoa earthquake by a slow slip event in guerrero, mexico. *Nature Geoscience*, *9*(11), 829–833.

Radiguet, M., Cotton, F., Vergnolle, M., Campillo, M., Valette, B., Kostoglodov, V., & Cotte, N. (2011). Spatial and temporal evolution of a long term slow slip event: The 2006 guerrero slow slip event. *Geophysical Journal International*, *184*(2), 816–828.

Radiguet, M., Cotton, F., Vergnolle, M., Campillo, M., Walpersdorf, A., Cotte, N., & Kostoglodov, V. (2012b). Slow slip events and strain accumulation in the guerrero gap, mexico. *Journal of Geophysical Research: Solid Earth*, *117*(B4).

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.

Rogers, G., & Dragert, H. (2003a). Episodic tremor and slip on the cascadia subduction zone: The chatter of silent slip. *Science*, *300*(5627), 1942–1943.

Rogers, G., & Dragert, H. (2003b). Episodic tremor and slip on the cascadia subduction zone: The chatter of silent slip. *Science*, *300*, 1942–1943. https://doi.org/10.1126/SCIENCE.1084783/ASSET/DCC117CE-4CAD-4799-90C0-B9DFE257EA05/ASSETS/GRAPHIC/SE2431617002.JPEG

Ross, Z. E., Cochran, E. S., Trugman, D. T., & Smith, J. D. (2020). 3d fault architecture controls the dynamism of earthquake swarms. *Science*, *368*, 1357–1361. https://doi.org/10.1126/SCIENCE.ABB0779/SUPPL_FILE/ABB0779_ROSS_SM.PDF

Ross, Z. E., Trugman, D. T., Hauksson, E., & Shearer, P. M. (2019). Searching for hidden earthquakes in southern california. *Science*. https://doi.org/10.1126/SCIENCE.AAW6888/SUPPL_FILE/AAW6888_ROSS_SM.PDF

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.

Rouet-Leduc, B., Hulbert, C., Bolton, D. C., Ren, C. X., Riviere, J., Marone, C., Guyer, R. A., & Johnson, P. A. (2018). Estimating fault friction from seismic signals in the laboratory. *Geophysical Research Letters*, *45*(3), 1321–1329.

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.

Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C. J., & Johnson, P. A. (2017). Machine learning predicts laboratory earthquakes. *Geophysical Research Letters*, *44*(18), 9276–9282.

Rouet-Leduc, B., Hulbert, C., McBrearty, I. W., & Johnson, P. A. (2020). Probing slow earthquakes with deep learning. *Geophysical research letters*, *47*(4), e2019GL085870.

Rouet-Leduc, B., Jolivet, R., Dalaison, M., Johnson, P. A., & Hulbert, C. (2021a). Autonomous extraction of millimeter-scale deformation in insar time series using deep learning. *Nature communications*, *12*(1), 1–11.

Rouet-Leduc, B., Jolivet, R., Dalaison, M., Johnson, P. A., & Hulbert, C. (2021b). Autonomous extraction of millimeter-scale deformation in insar time series using deep learning. *Nature Communications 2021 12:1*, *12*, 1–11. https://doi.org/10.1038/s41467-021-26254-3

Rousset, B., Campillo, M., Lasserre, C., Frank, W. B., Cotte, N., Walpersdorf, A., Socquet, A., & Kostoglodov, V. (2017a). A geodetic matched filter search for slow slip with application to the mexico subduction zone. *Journal of Geophysical Research: Solid Earth*, *122*, 10, 498–10, 514. https://doi.org/10.1002/2017JB014448

Rousset, B. (2019). Months-long subduction slow slip events avoid the stress shadows of seismic asperities. *Journal of Geophysical Research: Solid Earth*, *124*(7), 7227–7230.

Rousset, B., Campillo, M., Lasserre, C., Frank, W. B., Cotte, N., Walpersdorf, A., Socquet, A., & Kostoglodov, V. (2017b). A geodetic matched filter search for slow slip with application to the mexico subduction zone. *Journal of Geophysical Research: Solid Earth*, *122*(12), 10–498.

Rousset, B., Jolivet, R., Simons, M., Lasserre, C., Riel, B., Milillo, P., Çakir, Z., & Renard, F. (2016). An aseismic slip transient on the north anatolian fault. *Geophysical Research Letters*, *43*(7), 3254–3262.

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.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Satake, K., Wang, K., & Atwater, B. F. (2003). Fault slip and seismic moment of the 1700 cascadia earthquake inferred from japanese tsunami descriptions. *Journal of Geophysical Research: Solid Earth*, *108*(B11).

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*.

Schmidt, D., & Gao, H. (2010). Source parameters and time-dependent slip distributions of slow slip events on the cascadia subduction zone from 1998 to 2008. *Journal of Geophysical Research: Solid Earth*, *115*(B4).

Scholz, C. (1968). The frequency-magnitude relation of microfracturing in rock and its relation to earthquakes. *Bulletin of the seismological society of America*, *58*(1), 399–415.

Scholz, C. H. (1998). Earthquakes and friction laws. *Nature*, *391*(6662), 37.

Schreiber, T., & Schmitz, A. (1996). Improved surrogate data for nonlinearity tests. *Physical review letters*, *77*(4), 635.

Schreiber, T., & Schmitz, A. (2000). Surrogate time series. *Physica D: Nonlinear Phenomena*, *142*(3-4), 346–382.

Schurr, B., Asch, G., Hainzl, S., Bedford, J., Hoechner, A., Palo, M., Wang, R., Moreno, M., Bartsch, M., Zhang, Y., et al. (2014). Gradual unlocking of plate boundary controlled initiation of the 2014 iquique earthquake. *Nature*, *512*(7514), 299–302.

Schwartz, S. Y., & Rokosky, J. M. (2007a). Slow slip events and seismic tremor at circum-pacific subduction zones. *Reviews of Geophysics*, *45*(3).

Schwartz, S. Y., & Rokosky, J. M. (2007b). Slow slip events and seismic tremor at circum-pacific subduction zones. *Reviews of Geophysics*, *45*. https://doi.org/10.1029/2006RG000208

Segall, P., Desmarais, E. K., Shelly, D., Miklius, A., & Cervelli, P. (2006). Earthquakes triggered by silent slip events on kīlauea volcano, hawaii. *Nature*, *442*(7098), 71–74.

Sekine, S., Hirose, H., & Obara, K. (2010). Along-strike variations in short-term slow slip events in the southwest japan subduction zone. *Journal of Geophysical Research: Solid Earth*, *115*(B9).

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.

Shelly, D. R. (2010). Migrating tremors illuminate complex deformation beneath the seismogenic san andreas fault. *Nature*, *463*(7281), 648–652.

Shelly, D. R., Beroza, G. C., & Ide, S. (2007). Non-volcanic tremor and low-frequency earthquake swarms. *Nature*, *446*(7133), 305–307.

Shelly, D. R., Beroza, G. C., Ide, S., & Nakamula, S. (2006). Low-frequency earthquakes in shikoku, japan, and their relationship to episodic tremor and slip. *Nature*, *442*(7099), 188–191.

Shelly, D. R., Beroza, G. C., Zhang, H., Thurber, C. H., & Ide, S. (2006). High-resolution subduction zone seismicity and velocity structure beneath ibaraki prefecture, japan. *Journal of Geophysical Research: Solid Earth*, *111*(B6).

Shi, L., et al. (2019). Two-stream adaptive graph convolutional networks for skeleton-based action recognition. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12026–12035.

Shi, P., Seydoux, L., & Poli, P. (2021). Unsupervised learning of seismic wavefield features: Clustering continuous array seismic data during the 2009 l'aquila earthquake. *Journal of Geophysical Research: Solid Earth*, *126*(1), e2020JB020506.

Shishikura, M. (2014). History of the paleo-earthquakes along the sagami trough, central japan: Review of coastal paleoseismological studies in the kanto region. *Episodes Journal of International Geoscience*, *37*(4), 246–257.

Simons, M., Minson, S. E., Sladen, A., Ortega, F., Jiang, J., Owen, S. E., Meng, L., Ampuero, J.-P., Wei, S., Chu, R., et al. (2011). The 2011 magnitude 9.0 tohoku-oki earthquake: Mosaicking the megathrust from seconds to centuries. *science*, *332*(6036), 1421–1425.

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.

Socquet, A., Valdes, J. P., Jara, J., Cotton, F., Walpersdorf, A., Cotte, N., Specht, S., Ortega-Culaciati, F., Carrizo, D., & Norabuena, E. (2017). An 8 month slow slip event triggers progressive nucleation of the 2014 chile megathrust. *Geophysical Research Letters*, *44*(9), 4046–4053.

Soubestre, J., Shapiro, N. M., Seydoux, L., de Rosny, J., Droznin, D. V., Droznina, S. Y., Senyukov, S. L., & Gordeev, E. I. (2018). Network-based detection and classification of seismovolcanic tremors: Example from the klyuchevskoy volcanic group in kamchatka. *Journal of Geophysical Research: Solid Earth*, *123*(1), 564–582.

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.

Sun, T., Wang, K., Iinuma, T., Hino, R., He, J., Fujimoto, H., Kido, M., Osada, Y., Miura, S., Ohta, Y., et al. (2014). Prevalence of viscoelastic relaxation after the 2011 tohoku-oki earthquake. *Nature*, *514*(7520), 84–87.

Suzuki, W., Aoi, S., Sekiguchi, H., & Kunugi, T. (2011). Rupture process of the 2011 tohoku-oki mega-thrust earthquake (m9. 0) inverted from strong-motion data. *Geophysical Research Letters*, *38*(7).

Takagi, R., Uchida, N., & Obara, K. (2019). Along-strike variation and migration of long-term slow slip events in the western nankai subduction zone, japan. *Journal of Geophysical Research: Solid Earth*, *124*(4), 3853–3880.

Tan, Y. J., & Marsan, D. (2020). Connecting a broad spectrum of transient slip on the san andreas fault. *Science Advances*, *6*, 2489–2503. https://doi.org/10.1126/SCIADV.ABB2489/SUPPL_FILE/ABB2489_SM.PDF

Tan, Y. J., Waldhauser, F., Ellsworth, W. L., Zhang, M., Zhu, W., Michele, M., Chiaraluce, L., Beroza, G. C., & Segou, M. (2021). Machine-learning-based high-resolution earthquake catalog reveals how complex fault structures were activated during the 2016–2017 central italy sequence. *The Seismic Record*, *1*, 11–19. https://doi.org/10.1785/0320210001

Tan, Y. J., Waldhauser, F., Tolstoy, M., & Wilcock, W. S. (2019). Axial seamount: Periodic tidal loading reveals stress dependence of the earthquake size distribution (b value). *Earth and Planetary Science Letters*, *512*, 39–45.

Thomas, A., Melgar, D., Dybing, S. N., & Searcy, J. R. (2023). Deep learning for denoising high-rate global navigation satellite system data. *Seismica*, *2*(1).

Vaca, S., Vallée, M., Nocquet, J.-M., Battaglia, J., & Régnier, M. (2018). Recurrent slow slip events as a barrier to the northward rupture propagation of the 2016 pedernales earthquake (central ecuador). *Tectonophysics*, *724*, 80–92.

Vallée, M., Nocquet, J.-M., Battaglia, J., Font, Y., Segovia, M., Régnier, M., Mothes, P., Jarrin, P., Cisneros, D., Vaca, S., et al. (2013). Intense interface seismicity triggered by a shallow slow slip event in the central ecuador subduction zone. *Journal of Geophysical Research: Solid Earth*, *118*(6), 2965–2981.

van den Ende, M. P., & Ampuero, J.-P. (2020). Automated seismic source characterization using deep graph neural networks. *Geophysical Research Letters*, *47*(17), e2020GL088690.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 5998–6008.

Vergnolle, M., Walpersdorf, A., Kostoglodov, V., Tregoning, P., Santiago, J., Cotte, N., & Franco, S. (2010). Slow slip events in mexico revised from the processing of 11 year gps observations. *Journal of Geophysical Research: Solid Earth*, *115*(B8).

Vigny, C., Simons, W., Abu, S., Bamphenyu, R., Satirapod, C., Choosakul, N., Subarya, C., Socquet, A., Omar, K., Abidin, H., et al. (2005). Insight into the 2004 sumatra–andaman earthquake from gps measurements in southeast asia. *Nature*, *436*(7048), 201–206.

Vigny, C., Socquet, A., Peyrat, S., Ruegg, J.-C., Métois, M., Madariaga, R., Morvan, S., Lancieri, M., Lacassin, R., Campos, J., et al. (2011). The 2010 m w 8.8 maule megathrust earthquake of central chile, monitored by gps. *Science*, *332*(6036), 1417–1421.

Villegas-Lanza, J. C., Nocquet, J.-M., Rolandone, F., Vallée, M., Tavera, H., Bondoux, F., Tran, T., Martin, X., & Chlieh, M. (2016). A mixed seismic–aseismic stress release episode in the andean subduction zone. *Nature Geoscience*, *9*(2), 150–154.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wallace, L. M. (2020). Slow slip events in new zealand. *Annual Review of Earth and Planetary Sciences*, *48*, 175–203.

Wallace, L. M., & Beavan, J. (2010). Diverse slow slip behavior at the hikurangi subduction margin, new zealand. *Journal of Geophysical Research: Solid Earth*, *115*(B12).

Wang, K., & Tréhu, A. M. (2016). Invited review paper: Some outstanding issues in the study of great megathrust earthquakes—the cascadia example. *Journal of Geodynamics*, *98*, 1–18.

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.

Wdowinski, S., Bock, Y., Zhang, J., Fang, P., & Genrich, J. (1997). Southern california permanent gps geodetic array: Spatial filtering of daily positions for estimating coseismic and postseismic displacements induced by the 1992 landers earthquake. *Journal of Geophysical Research: Solid Earth*, *102*(B8), 18057–18070.

Wech, A. G. (2010). Interactive tremor monitoring. *Seismological Research Letters*, *81*, 664–669. https://doi.org/10.1785/GSSRL.81.4.664

Wech, A. G., & Bartlow, N. M. (2014). Slip rate and tremor genesis in cascadia. *Geophysical Research Letters*, *41*(2), 392–398.

Wech, A. G., Creager, K. C., & Melbourne, T. I. (2009). Seismic and geodetic constraints on cascadia slow slip. *Journal of Geophysical Research: Solid Earth*, *114*(B10).

Weng, H., & Ampuero, J.-P. (2022). Integrated rupture mechanics for slow slip events and earthquakes. *Nature Communications*, *13*(1), 7327.

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.

Williams, S. D., Bock, Y., Fang, P., Jamason, P., Nikolaidis, R. M., Prawirodirdjo, L., Miller, M., & Johnson, D. J. (2004). Error analysis of continuous gps position time series. *Journal of Geophysical Research: Solid Earth*, *109*(B3).

Woollam, J., Münchmeyer, J., Tilmann, F., Rietbrock, A., Lange, D., Bornstein, T., Diehl, T., Giunchi, C., Haslinger, F., Jozinović, D., Michelini, A., Saul, J., & Soto, H. (2022). Seisbench-a toolbox for machine learning in seismology. *Seismological Research Letters*, *93*, 1695–1709. https://doi.org/10.1785/0220210324

Yamanaka, Y., & Kikuchi, M. (2003). Source process of the recurrent tokachi-oki earthquake on september 26, 2003, inferred from teleseismic body waves. *Earth, Planets and Space*, *55*(12), e21–e24.

Yu, B., et al. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

Zhang, J., Bock, Y., Johnson, H., Fang, P., Williams, S., Genrich, J., Wdowinski, S., & Behr, J. (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.

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.

Zhang, X., Reichard-Flynn, W., Zhang, M., Hirn, M., & Lin, Y. (2022). Spatiotemporal graph convolutional networks for earthquake source characterization. *Journal of Geophysical Research: Solid Earth*, *127*(11), e2022JB024401.

Zhang, Y., Burton, H. V., Sun, H., & Shokrabadi, M. (2018). A machine learning framework for assessing post-earthquake structural safety. *Structural safety*, *72*, 1–16.

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.

Zou, C., Zhao, L., Xu, M., Chen, Y., & Geng, J. (2021). Porosity prediction with uncertainty quantification from multiple seismic attributes using random forest. *Journal of Geophysical Research: Solid Earth*, *126*(7), e2021JB021826.