



**SAPIENZA**  
UNIVERSITÀ DI ROMA

**Sapienza University of Rome**

Department of Computer, Control and Management Engineering  
PhD in Data Science

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

# **Artificial Intelligence in Geoscience Applications**

AI Models for Seismic Denoising,  
Earthquake Source Characterization,  
and Ground-Motion Prediction using  
the Italian Seismic Network

Thesis Advisor

**Prof. Fabio Galasso**

Co-Advisors

**Dott. Alberto Michelini**

**Prof.ssa Elisa Tinti**

Candidate

**Daniele Trappolini**

**1710415**

**Academic Year 2024-2025 (XXXVIII cycle)**





# Abstract

Earthquakes remain challenging to analyze in real time because waveforms are noisy, source parameters are uncertain, and shaking patterns evolve rapidly. This thesis advances the seismic monitoring pipeline with deep learning (DL) methods that improve data quality, reduce latency, and raise predictive accuracy on operational datasets and networks. First, we introduce CDIFFSD, a cold diffusion model for seismic denoising that learns non-Gaussian degradation processes directly from station noise. On the STEAD-based test, CDIFFSD increases median signal-to-noise ratio and cross-correlation over a strong deep-denoiser baseline, while markedly increasing P-pick recall within  $\pm 50$  samples (from  $\sim 0.60$  to  $\sim 0.90$  at  $T=300$ ). Second, we propose a masked Graph Convolutional Network for rapid shaking prediction in Italy. Using only the first 10 s of three-component waveforms from national-scale networks (INSTANCE), the model yields up to 6% and 5.5% mean-squared-error reductions for PGA and PGV, respectively, near-zero median residuals across intensity measures, and mitigates biases relative to GMM (an Italy-specific ground-motion model). Third, we develop LLM4GEO, a domain-adapted large language model that estimates epicenter, hypocenter, origin time, and magnitude within five seconds of first detection. On 3,094 test events, average errors are  $\sim 4$  km (epicenter),  $\sim 6.8$  km (hypocenter),  $\sim 0.6$  s (origin time), and  $\sim 0.2$  magnitude units, outperforming early-stage automated INGV solutions while using fewer picks. Finally, we present SEISMICAE, a waveform autoencoder that transfers audio-domain representation learning to seismology. The learned embeddings improve few-shot classification and reduce error in ground-motion regression relative to models trained on raw traces. In the end, we build a prototype *AI agent* that combines different AI models for specific tasks (expert) in a single workflow (orchestration), giving compatible results for the evaluation of operational seismic risk.

**Keywords:** AI, Deep Learning, Seismology, Earthquake Early Warning, Denoising, Ground Motion, Representation Learning, Foundation Models.



# Contents

|                                                                                  |            |
|----------------------------------------------------------------------------------|------------|
| <b>List of Figures</b>                                                           | <b>v</b>   |
| <b>List of Tables</b>                                                            | <b>xii</b> |
| <b>Acronyms</b>                                                                  | <b>xv</b>  |
| <b>1 Introduction</b>                                                            | <b>1</b>   |
| <b>2 Cold Diffusion Model for Seismic Denoising</b>                              | <b>3</b>   |
| 2.1 Overview and Related works . . . . .                                         | 3          |
| 2.2 Methods . . . . .                                                            | 5          |
| 2.2.1 Proposed Method: Cold Diffusion Seismic Denoising (CDiffSD) . . . . .      | 5          |
| 2.3 Data Sources and Selection . . . . .                                         | 10         |
| 2.4 Results . . . . .                                                            | 12         |
| 2.4.1 Quantitative Results . . . . .                                             | 13         |
| 2.4.2 Qualitative Results . . . . .                                              | 19         |
| 2.5 Model assessment: Assessing the Impact of Exclusive Noise Input . . . . .    | 24         |
| 2.6 Discussion . . . . .                                                         | 26         |
| <b>3 Masked graph neural network for rapid ground motion prediction in Italy</b> | <b>28</b>  |
| 3.1 Introduction . . . . .                                                       | 28         |
| 3.2 Data . . . . .                                                               | 30         |
| 3.2.1 Quality Control and Filtering . . . . .                                    | 31         |
| 3.2.2 Preparation for Model Training . . . . .                                   | 32         |
| 3.3 Method and Training . . . . .                                                | 33         |
| 3.3.1 Model Proposed . . . . .                                                   | 33         |
| 3.3.2 Mask Method: . . . . .                                                     | 35         |
| 3.4 Results . . . . .                                                            | 36         |
| 3.4.1 Evaluating the Impact of Additional Knowledge . . . . .                    | 36         |
| 3.4.2 Model performance . . . . .                                                | 37         |
| 3.4.3 Examples of event data prediction . . . . .                                | 37         |
| 3.4.4 Benchmarking the Model Against Baselines . . . . .                         | 38         |
| 3.4.5 Effect of Window length on Model Performance . . . . .                     | 40         |
| 3.5 Discussion . . . . .                                                         | 43         |
| 3.6 Conclusions . . . . .                                                        | 44         |
| <b>4 LLM4Geo</b>                                                                 | <b>46</b>  |
| 4.1 Introduction . . . . .                                                       | 46         |
| 4.2 An LLM for earthquakes . . . . .                                             | 49         |
| 4.2.1 INSTANCE dataset . . . . .                                                 | 49         |
| 4.2.2 Data selection . . . . .                                                   | 50         |
| 4.2.3 Data formatting . . . . .                                                  | 50         |

|          |                                                                      |            |
|----------|----------------------------------------------------------------------|------------|
| 4.2.4    | Training . . . . .                                                   | 51         |
| 4.3      | Results . . . . .                                                    | 52         |
| 4.3.1    | Test set performance . . . . .                                       | 52         |
| 4.3.2    | Toward EEWS: comparison of LLM and rapid automated results . . . . . | 53         |
| 4.4      | Discussion . . . . .                                                 | 56         |
| <b>5</b> | <b>SeismicAE: autoencoder for seismic waveforms embedding</b>        | <b>58</b>  |
| 5.1      | Overview and Related works . . . . .                                 | 58         |
| 5.2      | Data and Methods . . . . .                                           | 59         |
| 5.2.1    | Data . . . . .                                                       | 60         |
| 5.2.2    | Methods . . . . .                                                    | 60         |
| 5.3      | Results . . . . .                                                    | 62         |
| 5.3.1    | Reconstruction . . . . .                                             | 62         |
| 5.3.2    | Foreshock - Aftershock classification . . . . .                      | 67         |
| 5.3.3    | Ground motion prediction . . . . .                                   | 69         |
| 5.4      | Discussion . . . . .                                                 | 70         |
| <b>6</b> | <b>Conclusion</b>                                                    | <b>72</b>  |
| 6.1      | Final Remarks . . . . .                                              | 73         |
| 6.2      | Code and Data Availability . . . . .                                 | 73         |
| <b>7</b> | <b>Future Works</b>                                                  | <b>75</b>  |
| 7.1      | Future development directions . . . . .                              | 76         |
|          | <b>Bibliography</b>                                                  | <b>78</b>  |
| <b>A</b> | <b>Supplementary material for Chapter 2</b>                          | <b>92</b>  |
| A.0.1    | Diffusion Model overview . . . . .                                   | 92         |
| A.0.2    | Cold Diffusion Model overview . . . . .                              | 94         |
| A.0.3    | Gaussian Diffusion Model . . . . .                                   | 95         |
| A.0.4    | Qualitative Comparison . . . . .                                     | 95         |
| A.0.5    | Comparison with other methods . . . . .                              | 95         |
| A.0.6    | Training & Testing Details . . . . .                                 | 96         |
| A.0.7    | Supplementary tables . . . . .                                       | 98         |
| A.0.8    | Supplementary figures . . . . .                                      | 99         |
| <b>B</b> | <b>Supplementary Material for Chapter 3</b>                          | <b>109</b> |
| <b>C</b> | <b>Supplementary Material for Chapter 4</b>                          | <b>111</b> |
| C.0.1    | Model input and output . . . . .                                     | 111        |
| C.0.2    | Test set error distributions . . . . .                               | 111        |
| C.0.3    | Model performance and explainable machine learning . . . . .         | 112        |
| <b>D</b> | <b>Supplementary Material for Chapter 5</b>                          | <b>117</b> |
| D.0.1    | SeisCLIP . . . . .                                                   | 117        |
| D.0.2    | SeisT . . . . .                                                      | 117        |
| D.0.3    | SeisLM . . . . .                                                     | 118        |

# List of Figures

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |    |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | The figure represents the overall framework of our model. Specifically, it shows the handling of the input throughout the entire process: 1.Forward process definition: in the yellow box at the top left, we see an example of forward diffusion with $T=5$ ( $T = 5$ for graphical reasons). 2.Uniform Sampler involves uniform sampling $[1, T]$ . 3.Batch: At the bottom left, we find an example of a drawn batch (batch size = 4). This batch provides us with different levels of noise at the extracted time $T$ . The batch, noised according to the rules of the previously defined forward process, is passed to 4. 4.Denoising UNet model, which returns 5. 5.Predicion. This prediction is then compared with the original input in 6. 6.Compute Loss. . . . . | 7  |
| 2.2 | <i>CDiffSD</i> combines convolutional layers, <i>ResNet</i> blocks, attention mechanisms, and positional encoding (with positional encoding detailed in the bottom plots) for effective one-dimensional data processing. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 8  |
| 2.3 | The maps show the subset of stations of the <i>STEAD</i> used for the recorded <i>EQ</i> signal (upper) and the recorded noise (bottom). . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 11 |
| 2.4 | The histograms illustrate the frequency distribution of <i>EQ</i> magnitudes within our dataset, with the left panel representing the training and validation sets and the right panel the test set. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 11 |
| 2.5 | The image presents two maps of the geographical distribution of seismic events used in our study, with the upper map illustrating the events for the training and validation sets marked in red, and the bottom map showing the events for the test set in blue. The color intensity on each map corresponds to the number of events, with darker shades indicating a lower concentration of events in that location. . . . .                                                                                                                                                                                                                                                                                                                                               | 12 |
| 2.6 | <i>SNR</i> comparisons using box-plots for various model configurations applied to the test set. The original signals ("earthquake") and the ones with added noise ("eqk+noise") have respectively the higher and lower <i>SNR</i> , as expected. The different denoising models appear overall aligned, with direct and sampling showing slightly higher median values and tighter distributions with respect to <i>DD</i> . . . . .                                                                                                                                                                                                                                                                                                                                       | 13 |
| 2.7 | Distributions of <i>SNR</i> values of denoised waveforms for different ranges of input <i>SNR</i> . The <i>SNR</i> statistics after denoising are computed on 2dB wide ranges of input <i>SNR</i> . <i>CDiffSD</i> models show higher performances in low <i>SNR</i> scenarios, while <i>DD</i> is superior for the higher <i>SNR</i> signals. We study the range: $SNR < -2.0$ and $SNR > 16.0$ , which covers 99% of data. Solid bars within each model (color) show the median value. . . . .                                                                                                                                                                                                                                                                            | 14 |
| 2.8 | <i>CC</i> comparisons for various model configurations applied to the test set. Higher <i>CC</i> values indicate greater similarity between the denoised trace and the original signal. All <i>CDiffSD</i> models show similar performance and that they are consistently higher than <i>DD</i> . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 15 |
| 2.9 | Distribution of <i>CC</i> values between original and denoised signals (y axis) as a function of <i>CC</i> before application of denoising (x axis). <i>CDiffSD</i> models outperform <i>DD</i> across all <i>CC</i> ranges. The difference is more noticeable especially at low pre-denoising <i>CC</i> values. Statistics are computed for ranges of 0.1. Note that the distribution of samples for 'CC before denoising' is identical to "eqk+noise" in Fig. 2.8. Consequently, the 0.1-wide bins may encompass significantly varying numbers of samples. . . . .                                                                                                                                                                                                        | 15 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.10 | The histograms display the distributions of P-wave (blue) and S-wave (orange) arrival time differences between automated PhaseNet detections and label picks (in samples). The results obtained using the original seismograms, "eqk+noise" and <i>DD</i> are shown for comparison in the first, second and third row, respectively. The remaining rows show the results for different <i>CDiffSD</i> models applied to the same data subset, offering insights into the accuracy of wave arrival time detection by each model. Central tendency metrics, such as mean and median, are indicated in these histograms, highlighting any potential skewness in the distribution towards either early or late picks for both P and S waves. . . . . | 16 |
| 2.11 | Comparison of recall rates for P and S waves between the different methods within a fixed window of 50 samples. S-waves recall rates are aligned almost for all models, indicating that the noise level is not enough to affect the S-waves because of the greater amplitude. P-waves recall rates instead show significant disparities between the <i>DD</i> approach and other methods, suggesting a lower performance of <i>DD</i> in preserving the P onset in these cases. The 'sampling-300' method is confirmed as the one with better performances. . . . .                                                                                                                                                                              | 18 |
| 2.12 | Comparison of a trace processed using 'direct_300' and 'sampling_300' methods. Notably, the 'direct_300' retains some of the noise preceding the P wave arrival, which is instead filtered out in the 'sampling_300' results. This noise before the P-wave retained with the the 'direct_300' method explains the tendency for this model to cause early picks (as seen in Fig. 2.10), as the residual noise can lead to earlier detections. . . . .                                                                                                                                                                                                                                                                                             | 20 |
| 2.13 | Comparison between a seismic trace processed with 'deep_denoiser' and 'sampling_300' methods. The 'sampling_300' method demonstrates a closer match to label phase picks and a more precise amplitude preservation, despite the substantial noise present in the original signal. <i>DD</i> also retains a high amplitude noise signal at around 1500 samples that 'sampling_300' manages to filter out almost completely. . . . .                                                                                                                                                                                                                                                                                                               | 22 |
| 2.14 | Comparison of a seismic trace processed with 'direct_300' and 'sampling_300' methods. It is particularly significant that the 'sampling_300' technique demonstrates an enhanced ability for amplitude reconstruction compared to the 'direct_300' method. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 23 |
| 2.15 | Example of the outputs of direct (in black) and sampling (in red) methods in case of a noise only input (in grey). No retraining is performed here, i.e. the models have never been exposed to zero-traces as ground truth for noise-only input. The direct method fails in recovering a zero-trace since it introduces artificial signals. In contrast, the sampling method reconstructs successfully an output that resembles a zero-trace. . . . .                                                                                                                                                                                                                                                                                            | 25 |
| 2.16 | Example of the comparison between the sampling method (in red) and the direct method (in black), the input (in gray) for both methods is only noise. In this case the models have been retrained with zero-traces as ground truth for noise-only traces. The sampling method succeeds in reconstructing a zero-trace. On the other hand, the direct method outputs noise, indicating a less accurate reconstruction in this scenario. . . . .                                                                                                                                                                                                                                                                                                    | 25 |
| 3.1  | Spatial distribution of (a) the 565 stations (yellow triangles) together with (b) the 975 earthquakes (red dots) used in this study. We have selected earthquakes with magnitude $M \geq 3$ and depth $< 30$ km occurred in the study area bounded by latitude $[42^\circ, 43.75^\circ]$ and longitude $[12.3^\circ, 14^\circ]$ between January 2005 and January 2020. . . . .                                                                                                                                                                                                                                                                                                                                                                   | 31 |
| 3.2  | Distribution of earthquake (a) magnitudes, (b) depths and (c) years for the selected events. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 31 |
| 3.3  | The architecture of the proposed model for IM prediction. Boxes shaded grey represent inputs, shaded light blue represent convolutional layers, shaded blue represent operators, shaded green represent graph convolutional layers and shaded violet represent fully connected layers. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 34 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.4 | Boxplots for the residuals $[\log_{10} \frac{IM_{obs}}{IM_{pred}}]$ of the test set (195 events): (a) masked GCN model results for the observed IMs and (b) GMM results. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 38 |
| 3.5 | Observed versus predicted scatterplot of the testing dataset for: (a) PGA, (b) PGV, (c) SA(0.3), (d) SA(1.0) and (e) SA(3.0). Scatter points are both color and size-coded, with black indicating low-density areas and yellow representing high-density areas. The colorbar shows the total number of data points in each area. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 39 |
| 3.6 | Our model results for (a) the 26 August 2016 M 4.8 and (b) the 15 September 2016 M 3.7. The top row presents spatial distributions of observed PGA (left), predicted PGA (center), and their differences (right) across the stations, with color gradients representing the respective values. The bottom row includes scatter plots of PGA values as a function of epicentral distance: observed vs. predicted values (left) and their differences (right). Key performance metrics, such as MAE, MSE, and the range of observed and predicted PGA values, are summarized in the accompanying table (bottom right). . . . .                                                                                                                                                                                                              | 40 |
| 3.7 | Our model results for (a) the 26 October 2016 M 3.1 and (b) the 14 December 2016 M 3.8. Refer to the caption of Figure 3.6 for further details. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 41 |
| 3.8 | Our model results for (a) the 6 April 2009 M 6.1 and (b) the 13 June 2013 M 3.8. Refer to the caption of Figure 3.6 for further details. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 42 |
| 3.9 | MSE at 10s and 5s input window lengths for each IM. Our model achieves approximately the same MSE values even with half the input, demonstrating the effectiveness of GCNs in processing spatial information. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 44 |
| 4.1 | LLM model structure and workflow. a) Model summary showing 4 input values (station latitude and longitude, $\Delta T$ , and PGV), the decoder block followed by fine-tuning using the Low-Rank Adaptation (LoRA) method of freezing part of the pre-trained model weights, and 5 output values (earthquake latitude, longitude, depth, magnitude and origin time). b) Map showing an earthquake (yellow star) and local seismic stations (triangles, red stations were not used because s-waves were not detected). c) Seismograms from 3 stations showing vertical component of ground velocity and time delay between p-wave arrivals $\Delta T$ . Zoom shows time window for measurement of peak ground velocity (PGV). . . . .                                                                                                        | 49 |
| 4.2 | Map of earthquake epicenters determined by the LLM. Symbol size is proportional to estimated magnitude and color is binned by the epicentral location error, with error ranges (in kilometers) specified in the legend. Note that most event locations are within 10 km of the INGV bulletin values, which are considered our ground-truth values. Inset shows depth error distribution, and indicates that depth errors are small for most events. . . . .                                                                                                                                                                                                                                                                                                                                                                               | 52 |
| 4.3 | Comparison of INGV bulletin magnitudes with those estimated by the LLM. Symbol color is based on the number of events at each location (scale bar at right). The gray region above M 4 highlights the range where there are few events and estimates. . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 53 |
| 4.4 | Comparison of earthquake epicenter location error for our LLM (a) and the INGV '0' rapid solution (b). The LLM uses only a subset of the seismic phases used for the INGV '0' solution. Thus, these data differ from data shown in Fig. 4.2, both in terms of pick timing and the set of involved stations. Symbol size scales with the INGV bulletin magnitude, while color indicates the epicentral error range. Both plots display the percentage of events falling within each error interval. c) Depth error comparison. d) Comparison of the number of seismic phase picks used in input by the LLM and the INGV automatic method for the '0' estimates. Note that more of the LLM solutions have a depth error < 10 km (Panel c) and that the LLM solution '0' is generally obtained using fewer seismic phases (Panel d). . . . . | 55 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |     |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 4.5 | Comparison of magnitudes predicted by the LLM using PGV from P-waves and S-waves. PGV for P-waves is computed on the vertical component within a 0.2 s window centered on the P-wave arrival, while PGV for S-waves is calculated as the average of the two horizontal components within a 2.0 s window around the S-wave arrival . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                          | 57  |
| 5.1 | <i>SeismicAE</i> model architecture showing the encoder, decoder, and quantizer, with the six losses defined as the time series reconstruction ( $l_t$ ), mel-spectrogram ( $l_s$ ), spectra ( $l_{sp}$ ), quantizer ( $l_w$ ), the perceptual loss from the discriminator ( $l_d$ and $l_g$ ). . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 59  |
| 5.2 | Three examples of reconstruction for seismic traces from <i>STEAD</i> . The overall shape and major features of the waveforms are well-preserved in the reconstructions. Some high-frequency details are smoothed out in the reconstructed versions. The metrics (MSE, RMSE, MAE, R2) provide quantitative reconstruction quality measures. . . .                                                                                                                                                                                                                                                                                                                                                                                                                    | 64  |
| 5.3 | Three examples of reconstruction for seismic traces from <i>STEAD</i> , excluding the quantization. The reconstructions (orange) match the original traces (blue) more closely than in Figure 2. High-frequency details are better preserved without the quantizer. The metrics confirm that the overall fit is better. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                      | 65  |
| 5.4 | 2 examples of similar trace retrieved based on embedded and original seismic waveform. We display the best and worst examples based on difference between the selected P arrival, S arrival and Magnitude, versus the other detected similar traces. Only channel E is presented. The "Best example" shows a very close match between the target trace and the retrieved similar trace, preserving important seismic features like P and S wave arrivals and overall envelope shape. The "Worst example" still shows similarity in overall shape, but with more noticeable differences in timing and peaks. . . . .                                                                                                                                                  | 66  |
| 5.5 | Models used for our downstream tasks, combining the embedding coming from the autoencoder, concatenated with a final set of layers. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 68  |
| 5.6 | Peak Ground Acceleration (PGA) values across various stations in the Central Italy (CI) network. The plots compare real PGA values and those predicted by different models: <b>Left:</b> Real PGA values. <b>Center Left:</b> Predictions from the <i>GCN</i> model enhanced with our <i>SeismicAE</i> for encoding the traces. <b>Center Right Left:</b> Predictions from the original <i>GCN</i> model. <b>Right:</b> Predictions from the <i>CNN</i> model combined with <i>SeismicAE</i> for trace encoding. The circles represent the wave fronts after 10 seconds for the P wave ( <b>red</b> ) and the S wave ( <b>blue</b> ). If a trace contains the arrival of the S wave, it implicitly means that it also contains the P wave within 10 seconds. . . . . | 69  |
| 7.1 | Agentic EEW architecture ( <i>Seismo Agents</i> ). <i>PhaseNet</i> listens in for P-wave detection and activates <i>LLM4Geoscience</i> for speedy prediction of source parameters (latitude, longitude, depth, magnitude). Subsequently, the <i>Masked GCN</i> can predict shaking metrics (PGA/PGV/SA) throughout the station network. The pipeline works on discrete time windows (e.g., 5–10 s) and utilizes pretrained and task-specific models to provide modular orchestration and near real-time risk management support. . . .                                                                                                                                                                                                                               | 76  |
| A.1 | This example clearly highlights that the model has to be trained with real noise to be able to remove it accurately. This also proves that the seismic noise we want to remove is not just Gaussian noise, and the method has to be refined according to our final objective. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 99  |
| A.2 | <i>SNR</i> increased by more than 10 dB after denoising. P arrival is preserved and the automatic pick on the denoised signal is aligned with the manual label. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 100 |



|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |     |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| A.3  | comparison of 'direct_300' and 'sampling_300' methods. The 'direct_300' retains some of the noise before the P-wave, which is present also in the original trace. The 'sampling_300' method instead effectively removes it, demonstrating its superior noise reduction capability in the pre-P-wave segment. . . . .                                                                                                                                                                                                                                                                                                                                                            | 101 |
| A.4  | Comparison between <i>DD</i> and 'sampling_300' in a high <i>SNR</i> conditions. This example elaborates on the situation described in the main text about cases with high <i>SNR</i> in input. This figure provides an example of the fact that the value of the output <i>SNR</i> could be misleading sometimes in evaluating the actual performance of the denoiser. Here the <i>SNR</i> values on denoised traces do not reflect the actual quality of the denoising: while <i>DD</i> shows a higher <i>SNR</i> value, it clearly deletes one second of signal after the labeled P wave arrival. 'sampling_300' instead does a better job in preserving the P wave. . . . . | 102 |
| A.5  | The automatic S pick on the horizontal channel N is early with respect to the labeled one. This could indicate either a very good denoising performance of the 'sampling_300' model leading to a better S pick or unfiltered noise which is mistaken by PhaseNet for an S- arrival. . . . .                                                                                                                                                                                                                                                                                                                                                                                     | 103 |
| A.6  | A low-amplitude noise that was not present in the original trace is instead visible in the denoised trace before the P wave arrival. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 103 |
| A.7  | The P wave is treated as noise and filtered out, this results in an incorrect late P pick by PhaseNet. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 104 |
| A.8  | High amplitude noise before the P wave arrival is not filtered out. As a result the P pick is placed almost 100 samples early with respect to the true pick. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 104 |
| A.9  | Same <i>SNR</i> boxplot as in the main text, but with additional methods: Att <i>DD</i> (3 versions) and Wiener filter. Wiener filter shows lower performances than any other method. Att <i>DD</i> exhibit a high <i>SNR</i> , due to the very strong denoising of the waveform before the P arrival. . . . .                                                                                                                                                                                                                                                                                                                                                                  | 105 |
| A.10 | Same <i>CC</i> boxplot as in the main text, but with additional methods: Att <i>DD</i> (3 versions) and Wiener filter. Wiener filter shows lower performances than any other method. The performances of Att <i>DD</i> are aligned with the original <i>DD</i> . . . . .                                                                                                                                                                                                                                                                                                                                                                                                        | 105 |
| A.11 | Histograms of picks difference between PhaseNet and manual picks comparing additional methods. Wiener filter shows little improvement with respect to applying PhaseNet directly on the noisy waveforms (eqk+noise). The behaviour of the 3 Att <i>DD</i> methods exhibit double maxima in the distributions, highlighting a frequent early pick around 20 samples before the labeled one. . . . .                                                                                                                                                                                                                                                                              | 106 |
| A.12 | Comparison of <i>DD</i> vs Att <i>DD</i> . This figure highlights both the reason for the very high <i>SNR</i> of Att <i>DD</i> and the double maxima shown in A.11. Despite a lower <i>SNR</i> , the waveform denoised with <i>DD</i> picks the P arrival correctly. In Att <i>DD</i> instead PhaseNet picks a P as soon as the amplitude rises from the almost-zero level at the beginning. . . . .                                                                                                                                                                                                                                                                           | 107 |
| A.13 | Comparison of <i>DD</i> vs Att <i>DD</i> . In this example <i>DD</i> doesn't denoise the waveform enough to enable a P pick. Att <i>DD</i> does a better job in denoising the first portion of the waveform, but the transition from noise to signal is too abrupt, giving again rise to an early pick. . . . .                                                                                                                                                                                                                                                                                                                                                                 | 108 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |     |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| B.1 | This map illustrates the application of the mask on the nearly 600 seismic stations from the INSTANCE benchmark dataset (Michelini et al., 2021). A sample earthquake from those used in this study was examined: 9068331. All green triangles denote stations that detected the P-wave arrival within 10 seconds for this event according to the INSTANCE metadata, and for this reason, they are all included by the mask during the training phase. Conversely, all red triangles indicate stations where the P-wave does not arrive within the first 10 seconds starting from the P arrival at the first recording station, and for this reason, they are excluded by the mask in the training phase. In the test phase, the stations included by the mask are represented exclusively by the green triangles within the blue bounding box, located approximately $\pm 0.58$ degrees from the first recording station. This setup replicates an operational scenario in which the model estimates travel times assuming a P-wave velocity of 6.5 km/s based on the P arrival at the closest station provided in the INSTANCE metadata. The yellow bounding box defines the area used for metric calculations, set at approximately $\pm 0.7$ degrees from the first recording station. . . . | 110 |
| C.1 | Location Error Distributions on the Test Set . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 112 |
| C.2 | Magnitude Error Distribution on the Test Set . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 112 |
| C.3 | Location Error. On the x-axis there are the percentile of the left edge of the epicentral error interval taken in consideration C.1. Each interval contains 5% of the test set events. On the y-axis are the mean and median of, from left to right, the number of stations used in the input (a), the azimuthal gap (b) and the target depth (c). . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 113 |
| C.4 | (a) shows the epicentral error as a function of the year of the event. (b) shows the absolute magnitude error as a function of the year of the event. In both cases, a blue line is overlaid representing the median of the errors for each year, and years with a number of events less than 100 have been excluded. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 115 |
| C.5 | Both plots show the comparison between model predictions when using the real stations coordinates and the transposed ones. (a) is related to an originally deep event located in the sea area between Sicily and Calabria and then transposed in central Italy. (b) is related to an originally shallow event located in central Italy and then transposed further north. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 116 |
| D.1 | Three examples of waveform reconstruction from STEAD. The trace names in the dataset are (left to right) "AMT.HP_20120416124803_EV", "KHIT.AK_20140110013613_EV", "AKGG.AV_20111028181955_EV". The input signal is shown in blue, in orange is the reconstruction without the quantizer, and in green is the reconstruction with the quantizer. The listed metrics are mean-squared-error (MSE, the lower the better), root-mean-squared-error (RMSE), mean-absolute-error (MAE, the lower the better), and correlation (R2, the higher the better) provide quantitative reconstruction quality measures. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 119 |
| D.2 | Two examples of similar traces retrieved based on embedded and original seismic waveform. Here we display the best (panel a) and worst (panel b) examples with the 10 most similar traces based on the difference between the selected P arrival, S arrival and Magnitude, versus the other detected similar traces. Only channel E is presented. The "Best example" shows a very close match between the target trace and the retrieved similar trace, preserving important seismic features like P and S wave arrivals and overall envelope shape. The "worst example" still shows similarity in overall shape, but with more noticeable differences in timing and peaks. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 120 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| D.3 | Distribution of residuals ( $\Delta t$ ) of the tested models on STEAD dataset. Panel (a) PhaseNet. P-arrival: mean=0.070797, std=0.110022; S-arrival: mean=0.051755, std=0.138495. Panel (b) Frozen ENC + head. P-arrival: mean=-0.013342, std=0.247148; S-arrival: mean=-0.002217, std=0.253981. Panel (c) Pre-trained ENC + DEC. P-arrival: mean=0.000846, std=0.115684; S-arrival: mean=0.005798, std=0.145941. Panel (d) Random ENC + DEC. P-arrival: mean=0.007378, std=0.129801; S-arrival: mean=0.014513, std=0.151914. . . . .                                                                                                                        | 121 |
| D.4 | Distribution of residuals ( $\Delta t$ ) of the tested models on the INSTANCE dataset, to evaluate generalizability. Panel (a) PhaseNet. P-arrival: mean=0.082230, std=0.105384; S-arrival: mean=0.082885, std=0.123061. Panel (b) Frozen ENC + head. P-arrival: mean=0.010321, std=0.236151; S-arrival: mean=0.030189, std=0.247748. Panel (c) Pre-trained ENC + DEC. P-arrival: mean=-0.044369, std=0.128707; S-arrival: mean=-0.010979, std=0.153109. Panel (d) Random ENC + DEC. P-arrival: mean=-0.010560, std=0.136742; S-arrival: mean=0.023156, std=0.151109. . . . .                                                                                  | 122 |
| D.5 | In panel (a), we present a detailed qualitative comparison among models, we randomly selected one example from the test set of the INSTANCE dataset. We display here the Z channel of the waveform with trace name “10233131.IV.T1256..HH”. The input signal is shown in black, the vertical lines are the real or predicted phases for each tested model. We also zoom around the P (highlighted in pink) and S (highlighted in green) arrival, for a better visual analysis. In panel (b), we compare the pretrained and random models in a few shot scenario. Boxplots refer to 10 runs for each configuration. 10 in total for each configuration. . . . . | 123 |
| D.6 | Three qualitative examples to compare the performance of the analyzed models. Rows are the four compared models, columns are three examples (trace names). We only display the Z channel of three randomly selected earthquakes from the test set of the STEAD dataset. . . . .                                                                                                                                                                                                                                                                                                                                                                                | 124 |
| D.7 | Three qualitative examples to compare the performance of the analyzed models. Rows are the four compared models, columns are three examples (trace names). We only display the Z channel of three randomly selected earthquakes from the INSTANCE dataset, to evaluate generalizability. . . . .                                                                                                                                                                                                                                                                                                                                                               | 125 |

# List of Tables

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Summary of the metrics obtained with different denoising methods. Best score for each metric in bold. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 19 |
| 4.1 | Earthquakes used for the LLM. Here, intervals are denoted [a, b), where a is included and b is excluded. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 50 |
| 4.2 | Training Parameters . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 51 |
| 4.3 | Comparison of results from the LLM and INGV automated methods. In both cases the INGV methods 0, 1, and 2 are used. <i>Fine-Tuned LLM</i> columns show our model performance using data recovered using INGV methods 0, 1, 2. <i>INGV</i> columns show performances for the same three methods. Note that the LLM model performs better on average for epicentral and hypocentral location and that magnitude estimates are similar. . . . .                                                                                                                                                                                                                                                                                                                                                                                                 | 54 |
| 4.4 | Magnitude Estimation Results . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 56 |
| 5.1 | Performance metrics for models trained on <i>STEAD</i> and <i>INSTANCE</i> datasets, including <i>STEAD</i> noise, and on <i>STEAD</i> only (without noise). We evaluate the models on the reconstruction task, showing the results with (99 % compression) and without (60 % compression) the quantizer. Metrics used: <b>MSE</b> (Mean Squared Error) measures the average squared prediction error. <b>RMSE</b> (Root Mean Squared Error) indicates the standard deviation of prediction errors. <b>MAE</b> (Mean Absolute Error) measures average prediction accuracy. <b>R<sup>2</sup></b> (R-squared) indicates the proportion of variance explained by the model, ranging from -1 to 1, with higher values indicating better fit. The Table compares performance across <i>STEAD</i> , <i>INSTANCE</i> , and Norcia datasets. . . . . | 62 |
| 5.2 | For each trace of the test dataset, we make a comparison with all the others, as explained in section 3.1 Retrieval. Here we compute the absolute difference value between features of interest (P arrival, S arrival, Magnitude) of each target trace vs the 10 most similar traces. The lower the better. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 66 |
| 5.3 | Performance results (ratio of correctly classified events to total events) for the testing phase of the <i>DL</i> model trained to distinguish foreshock, aftershock, and <i>TTF</i> . This is to compare the models trained on 70% of the dataset (5076 samples) vs the few shot case, i.e. 2% of the dataset (128 samples). Comparing the model from that is a 7 layers <i>CNN</i> ( <i>7CNN</i> ) vs the model made of the embedding concatenated with a 1 layer <i>CNN</i> ( <i>Enc+1CNN</i> ). Our results are averaged over 3 runs in the standard case (training on 70% of the dataset), and averaged over 10 runs in the case of few shot training. . . . .                                                                                                                                                                          | 67 |
| 5.4 | Results measured in MSE. Our results are averaged over 10 runs. STD is around $10^{-2}$ . In the standard case we have 586 examples in training, while in the few shot case we have 80 examples in training. <i>GCN</i> results refer to the original work [17]. <i>Enc+CNN</i> (or <i>GCN</i> ) is made of the embedding concatenated with a <i>CNN</i> (or <i>GCN</i> ). . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 69 |
| A.1 | Training Time and Hardware Details . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 98 |
| A.2 | Sensitivity Analysis on hyper-parameters . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 98 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |     |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| B.1 | MSE results for the eight experiments on our data set. The letters A, B, and C denote the coordinates of the first station that records the P-wave, the interstation distances and the maximum amplitude information, respectively. These values are inserted as additional metadata into the flattened layer. . . . .                                                                                                                                      | 109 |
| B.2 | IM residual statistics $\log_{10} \frac{IM_{obs}}{IM_{pred}}$ for the Masked GCN predictions and for the estimates obtained using the Bindi et al. (2011) GMM. The comparison was performed solely for the observed IMs at stations with available recorded data. The statistics are calculated after the outlier residuals (those with a residual $>  1 $ ) have been removed. The number of outliers of the GCN is 5.8% and for the GMM is 2.4% . . . . . | 109 |
| B.3 | MAE and MSE results of each IM for the CNN model from Jozinovic et al. (2020), the TISER-GCN model from Bloemheuvel et al. (2023) and our proposed model (Masked TISER-GCN). . . . .                                                                                                                                                                                                                                                                        | 109 |
| C.1 | Error Interval used in Fig.C.3 . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                    | 114 |
| C.2 | Test Results with zero PGVs . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                       | 116 |

# Acronyms

**3C** Three-Component (seismic waveform)

**AE** Acoustic Emission

**CC** Cross Correlation

**CDiffSD** Cold Diffusion Model for Seismic Denoising

**CNN** Convolutional Neural Network

**DD** Deep Denoiser

**DL** Deep Learning

**DM** Diffusion Model

**EEW** Earthquake Early Warning

**EQ** Earthquake

**FM** Foundation Model

**GCN** Graph Convolutional Network

**INSTANCE** Italian Seismic Dataset for Machine Learning

**LLM** Large Language Model

**LSTM** Long Short-Term Memory

**ML** Machine Learning

**NN** Neural Network

**NRF** Noise Reduce Factor

**PGA** Peak Ground Acceleration

**PGV** Peak Ground Velocity

**ResNet** Residual Neural Network

**SA** Spectral Acceleration

**SeismicAE** Seismic Waveform Auto-Encoder

**SNR** Signal-to-Noise Ratio

**STEAD** STanford EArthquake Dataset

**STFT** Short-Time Fourier Transform

**TTF** Time To Failure

# Chapter 1

## Introduction

Earthquakes are still a leading natural hazard; even modest improvements in seismic data processing and estimation of source and shaking parameters would improve public safety as well as risk mitigation. The recent evolution of machine learning (ML) and deep learning (DL) has made it possible to address four key scenarios: waveform denoising, phase picking, rapid source characterization, and ground-motion prediction. This thesis supports that agenda by developing and testing DL methods throughout the seismic monitoring pipeline to provide real improvement in data quality, inference accuracy, and timeliness on real networks and events.

In addition to earthquake signals seismic waves contain non-Gaussian noise that complicates the downstream tasks of detection, phase picking, and inversion. To cope with these problems, we developed Cold Diffusion model for Seismic Denoising (CDIFFSD) [158] a cold-diffusion approach that processes seismograms and their non-Gaussian noise. CDIFFSD differs from standard diffusion models, which assume Gaussian noise and perturbations. Our approach trains CDIFFSD using inverse natural degradation processes estimated from recorded station noise. CDIFFSD increases median signal-to-noise ratio (SNR) and zero-lag cross-correlation (CC) for a strong baseline: deep-denoiser (DD) [187] across a large dataset: STanford EArthquake Dataset (STEAD) [116]. Remarkably, the median P-phase pick recall increases from  $\sim 0.60$  (DD) within  $\pm 50$  samples to  $\sim 0.90$  (sampling,  $T=300$ ), an improvement of  $\sim 50\%$  that is relevant for onset-sensitive tasks. We see an  $\sim 18\%$  increase in SNR and  $\sim 6\%$  improvement in cross-correlation (CC) compared to DD. Among the different variants, the sampling with  $T = 300$  provides the most beneficial performance overall for SNR, CC and pick fidelity in terms of general balance than the other choices. Even at noise-dominated inputs, this behaviour applies: the sampling model generates outputs nearly to zero at  $\sim 90\%$  even under noise-only cases even after briefly retraining on zero traces.

We pursue the quick shaking prediction in the context of short-window waveforms (e.g., the first 10 s). There, we introduce a masked graph convolutional network for Italy [159] that ingests 10s three-component waveforms (from origin time) as input data at a national scale ( $\sim 600$  stations in INSTANCE (The Italian seismic dataset for machine learning [106]) with no prior knowledge of magnitude, distance, and location. The station mask hides the places where the P wave has not yet developed, so Graph Convolutional Netowrk (GCN) is able to capture changing spatial data as wavefront gets bigger. We minimize mean squared error (MSE) relative to no mask baseline



(up to 6% for peak ground acceleration (PGA) and 5.5% for peak ground velocity (PGV)), obtain a nearly null median residual for intensity measures and alleviate the systematic bias when using Italian based ground motion model. Results show a reliable  $\sim 96\%$  prediction in particular of events by case studies, except that it is much worse for large magnitudes with weak station coverage.

To reduce the end-to-end latency of Earthquake Early Warning (EEW) in those cases, we extend Large Language Model for Geoscience application (LLM4GEO) [12]. This model estimates the epicenter, hypocenter, origin time, and magnitude using only stations within five seconds of each first detected phase, using a domain-adapted approach. Overall, on 3,094 INSTANCE test events, average epicentral and hypocentral errors are  $\sim 4$  km and  $\sim 6.8$  km, respectively, with MAE  $\sim 0.6$  s for origin time and  $\sim 0.2$  magnitude units. Relative to INGV's quick automated "0/1/2" solution, the LLM realizes small location errors with fewer phase picks; depth errors are lower throughout most of Italy, but higher off Calabria where station density is lower, from a mismatch perspective.

Finally, the thesis also introduced Seismic Auto-Encoder (SEISMICAE) [88], a waveform autoencoder that applies audio-domain representational learning to seismology and provides compact embeddings to preserve the relevant structure of the task (e.g. P / S timing and magnitude proxies). SEISMICAE improves few-shot performance and can be put to use for a variety of downstream tasks including trace reconstruction, classification, and ground-motion regression. For classification, with SEISMICAE, replacing a deep CNN with a shallow head over SEISMICAE embeddings boosts accuracy dramatically in few-shot regimes. For ground-motion prediction, concatenation of embeddings combined with Convolutional Neural Netowk (CNN)/GCN heads leads to a decrease in MSE for PGA/PGV/SA metrics compared with baseline training on raw traces.

**Contributions.** To summarize, the thesis includes: (i) a denoising framework (CDIFFSD) that mitigates non-Gaussian seismic noise, and makes a meaningful difference in onset preservation and downstream picking; (ii) a masked GCN with no source parameters pre-supplied for fast, network scale ground motion prediction; (iii) LLM4GEO to provide source estimates that can be equivalent to (or better), than even in certain cases, operational automated solutions that are dependent on reduced number of phases; and (iv) SEISMICAE as a reusable representation that is able to learn fast and outperform baseline estimates under data scarcity, serving as a model for a foundation for seismology.

**Toward operations.** Ideally, the project achieves a prototype of an AI agent that integrates the masked GCN, LLM4GEO, and other AI models to develop a pipeline for end-to-end operating conditions. The agent orchestrates data ingestion, feature extraction, quick source characterization, and real-time shaking prediction to provide outputs directly actionable for seismic-risk assessment and situational awareness. Thanks to modularity, this system seamlessly complements and integrates with existing systems, as well as supporting incremental upgrades when newer models and data come onto the scene.

## Chapter 2

# Cold Diffusion Model for Seismic Denoising

In this Chapter, we develop a tool for seismic denoising. Seismic waves contain clues about *Earthquake (EQ)*s and what's beneath the Earth's surface, but any recording of these waves is often mixed with unwanted noise or disturbances to varying degrees. It's important to filter out these disturbances from the *EQ* recordings to improve their clarity and, as a result, make any further analysis more accurate. However, this can be tricky because the nature of these disturbances can change over time, including their amplitude, or by analogy to audio: how loud they are and their pitch of high and low notes. Recently, advanced computer methods that are good for improving images and sounds have shown promising results. But, these methods usually look for disturbances that follow a certain pattern, which does not always work for more complex disturbances found in *EQ* data. To address this, we introduce a strategy based on *Cold Diffusion Model (DM)* for Seismic Denoising, which removes noise and thus cleans up recordings to make them more understandable.

We describe the first *Cold Diffusion Model for Seismic Denoising (CDiffSD)*, including key design aspects, model architecture, and noise handling. *Cold Diffusion Model for Seismic Denoising (CDiffSD)* significantly improved the *Signal-to-Noise Ratio (SNR)* by about 18% compared to previous models. It also enhanced *Cross Correlation (CC)* by 6%, showing a better match between denoised and original signals. Moreover, testing revealed a 50% increase in the recall of P-wave picks for seismic picking. Our work show that *CDiffSD* outperforms existing benchmarks, further underscoring its effectiveness in seismic data denoising and analysis. Additionally, the versatility of this model suggests its potential applicability across a range of tasks and domains, such as GNSS, Lab *Acoustic Emission (AE)*, and DAS (Distributed Acoustic Sensing) data, offering promising avenues for further utilization.

## 2.1 Overview and Related works

Seismograms contain signals generated by *EQ*s and by other unidentified sources categorized in general as 'noise' (e.g., oceanic waves, wind, vehicular traffic, sonic booms, quarry activities, and instrument malfunctions.). It is standard practice in seismology to denoise waveforms to improve the performance of the subsequent analyses, such as P- and S-wave onset picking, *EQ* source moment tensor inversion, and techniques of exploration seismology. Seismic denoising is also decisive to

improve seismic catalogs and enhance their completeness, to consequentially enlighten fault structures. A starting point is to separate the noise from the EQ signal by denoising traces, this can also allow us to detect subtle precursory signals, which can be subtle in the noise.

In routine analysis, denoising is performed most commonly through bandpass filtering. However recent works have proposed several more sophisticated schemes to "clean" seismic traces. These include methods based on the independent component analysis (ICA) [27, 35, 108], beamforming methods [24, 26, 55], and MUltiple Signal Classification (MUSIC) [13, 142]. All of these methods, however, can fall short when the noise shares frequencies with the *EQ* generated signal.

Denoising models have evolved to incorporate time-frequency methods, with techniques like the Wavelet transform [29, 51, 96, 115, 147, 182], the *Short-Time Fourier Transform (STFT)* [114], the S-transform [161], and other transformation-decomposition methods [14, 32, 59, 62, 97, 121, 143, 153]. These techniques have proven useful but the emergence of *Deep Learning (DL)* has provided new strategies with improved performance. A notable development in this area is the *Deep Denoiser (DD)* model [187]. The *DD* approach is based on a UNet architecture, which generates dual masks for seismic and noise signals, enhancing waveform extraction. Another notable approach is that of [163] who employed *DL* to denoise Fiber-optic DAS data. They demonstrate the potency of *DL* to enhance the quality of DAS and seismic data. Similarly, the [122] project, utilizing a Dual-Path Recurrent Neural Network, led to another substantial stride in the application of *DL* for seismic signal denoising. These studies not only validate the efficacy of *DL* methods in seismic noise reduction but also pave the way for further innovations in this field.

Here we built on this topic, drawing parallels with techniques used in speech enhancement, a field closely related to seismic denoising. Speech enhancement has recently seen the use of models such as GANs (Generative Adversarial Networks) [28, 40, 83, 125] and VAE (Variational Autoencoder) [15, 48, 91, 92]. However, the recent trend points to the growing success of *DM* [63, 149], which are now outperforming their predecessors (GAN & VAE) see [99, 133]. Using techniques like cold diffusion or Gaussian diffusion for denoising presents several advantages over approaches that use binary masks, especially in terms of flexibility, reconstruction quality, and the ability to handle complex noise; while binary generally retain advantages in terms of simplicity, speed, interpretability, and computational efficiency. Here, we investigate the application of *DM* for seismic denoising. These models typically transform the input into an isotropic Gaussian distribution through the consistent addition of Gaussian noise. In the reverse process, diffusion probabilistic models aim to remove the anticipated noise from the corrupted input, thus recovering the original signal. A pioneering approach to seismic denoising using *DM* with Gaussian noise was introduced by [41], specifically applied on shot gathers used for seismic imaging and exploration.

Recognizing the limitations of traditional *DM* that assume a Gaussian distribution of noise, which is often not the case in seismic applications, we introduce a "cold" variant of *DM*. This adaptation is specifically tailored to restore clean and noisy seismic traces more effectively. The emerging Cold *DM* [11, 178] adapts the diffusion process by replacing Gaussian noise with other types of noise and signal degradation processes. The Cold *DM* demonstrates how *DM* can effectively restore signals impaired by various types of degradation. Its inherent properties make it particularly suitable for tasks such as speech source separation in practical settings with non-Gaussian noise. Building on this, our research aims to adapt the cold diffusion paradigm for seismic trace denoising,

even in scenarios where background noise levels are nearly as high as the *EQ* signals themselves. This adaptation involves specific modifications, primarily in the sampling algorithm, adapting it to handle noise directly recorded from seismic stations, to suit the unique challenges of seismic data.

## 2.2 Methods

The model we propose is based on a generalization of *DM*, termed the Cold *DM*. In this section we introduce the model. Additional details about *DM* and Cold *DM* are found in Appendix A.0.1, B, and A.0.3.

### 2.2.1 Proposed Method: Cold Diffusion Seismic Denoising (CDiffSD)

#### Problem formulation

The core of our *CDiffSD* involves degrading a one-dimensional *EQ*, in the form of a seismic record,  $x_0$  (the target), with recorded seismic noise  $x_n$ , to produce  $x_T$  (noisy signal):

$$x_T = x_0 + x_n * NRF \quad (2.1)$$

Here,  $x_0$  represents an *EQ* recorded by a seismometer, while  $x_0$  serves as a 'clean' sample in our context, it's important to note that it inherently contains some level of noise, given its real (non-synthetic) origin. Since we deal with normalized earthquakes and noises in the range of  $[-1, 1]$  (for more details on the normalization process see 2.2.1); the *Noise Reduce Factor (NRF)* is a key element in our specific analysis. It's responsible for calibrating the amplitude of the noise signal ( $x_n$ ) in relation to the *EQ* signal's amplitude, often indicated by the amplitude of S-waves in the data. By choosing a *NRF* value within the range 0.4 to 0.65, we ensure that the noise does not dominate the trace compared to the *EQ*. We work with data from different stations that independently record noise and *EQ* signals. It's worth mentioning that we mix *EQ*  $x_0$  and noise  $x_n$  recorded from different seismic stations, to improve generalizability and robustness.

#### CDiffSD Training

Regarding the specific operation of cold *DM*, our approach is delineated using the improved training algorithm proposed by [178]:

Concerning the forward diffusion process degradation see Appendix A.0.1, we can rephrase the degradation at time  $t$  as follows:

$$x_t = D_{x_T}(x_0, t) = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}x_T \quad (2.2)$$

where  $x_0$  is the recorded *EQ*,  $x_t = x_0 + x_n * NRF$  and  $\alpha \in [0, 1]$  is the parameter interpolation weight.  $\alpha$  can also be regarded as the amount of information retained in the diffusion process, and it can alternatively be defined as  $1 - \beta$ , where  $\beta$  represents the amount of noise introduced in the degradation, such parameters are defined a priori by a scheduler:  $\{\beta_t \in (0, 1)_{t=1}^T\}$ .

**Algorithm 1** Cold Diffusion Enhanced Training**for**  $n = 1, \dots, N_{iter}$  **do**Sample clean data  $x_0$ Sample  $t \sim \text{Uniform}(\{1, \dots, T\})$  $x_t \leftarrow D(x_0, t), \hat{x}_0 \leftarrow R_\theta(x_t, t)$ Sample  $t' \sim \text{Uniform}(\{1, \dots, t\})$  $\hat{x}_{t'} \leftarrow D(\hat{x}_0, t'), \hat{\hat{x}}_0 \leftarrow R_\theta(\hat{x}_{t'}, t')$ Take gradient descent step on  $\nabla_\theta(\|\hat{x}_0 - x_0\|_1 + \|\hat{\hat{x}}_0 - x_0\|_1)$ **end for**

- $t$ : Represents a chosen random timestep within the predefined range  $[1, T]$
- $t'$ : Signifies an earlier timestep than  $t$ , facilitating a recursive learning process where the model iterates through noise addition and removal at progressively earlier moments.
- $x_t$ : Result of the degradation applied to the signal  $x_0$  following the forward process as a function of  $t$
- $D(x_0, t)$ : Refers to the degradation applied to the signal  $x_0$  following the forward process as a function of  $t$
- $R_\theta(x_t, t)$ : Refers to the reconstruction applied to the signal  $x_t$  and it results in producing:  $\hat{x}_0$
- $\hat{x}_0$ : the result of the Restoration:  $R_\theta(x_t, t)$
- $\hat{x}_{t'}$ : Result of the degradation applied to the signal  $\hat{x}_0$  following the forward process as a function of  $t'$
- $\hat{\hat{x}}_0$ : the result of the Restoration:  $R_\theta(\hat{x}_{t'}, t')$

This method improves the model's ability to learn during training, especially when dealing with unusual, non-Gaussian noise. In the training stage, the model picks a random timestep  $t$  from the range  $[1, T]$ . At this time, noise is added to the signal, followed by a cleaning step. This is important because it teaches the model to remove noise, mimicking the denoising process. The model's learning is deepened by repeating these steps at an earlier moment  $t'$ , where  $t' < t$ . Here, the model works not with the original  $EQ$  data but with the signal that was cleaned in the previous step. This signal is made noisy again up to the new time  $t'$  and cleaned once more. By doing this over different times, the model learns more effectively, getting better at handling the complex types of noise found in real data. The training approach is designed to be forgiving of mistakes that can happen when choosing moments to sample from. As outlined in Algorithm 1, the training includes using  $\hat{x}_{t'}$ , the cleaned signal. This introduces a way to deal with potential alignment mistakes that might happen during sampling.

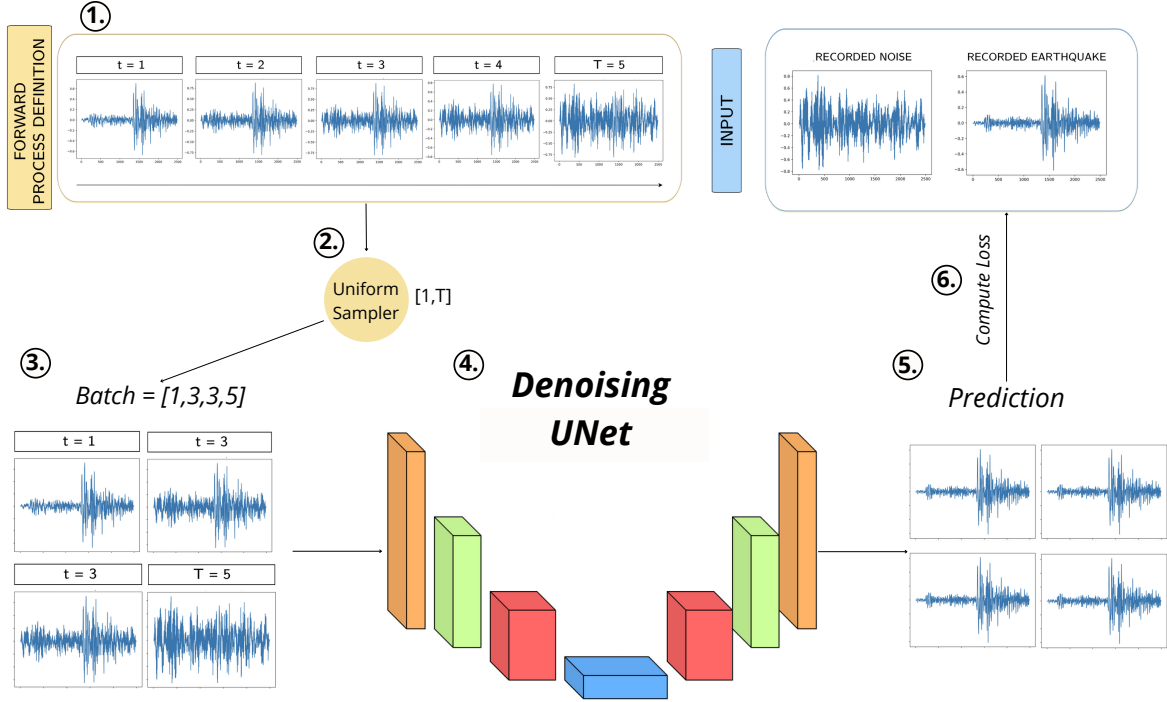
**Sampling:**

Regarding the sampling, our model employs the same approach as used in "Cold diffusion for speech enhancement" [178]. Specifically, we take the sampling algorithm 4 in Appendix B and substitute

how  $x_{t-1}$  is calculated. In detail, our new  $x_{t-1}$  becomes:

$$x_{t-1} \leftarrow \sqrt{\alpha_{t-1}}\hat{x}_0 + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_t}}(x_t - \sqrt{\alpha_t}\hat{x}_0) \quad (2.3)$$

### Input Assumptions



**Figure 2.1:** The figure represents the overall framework of our model. Specifically, it shows the handling of the input throughout the entire process:

1. Forward process definition: in the yellow box at the top left, we see an example of forward diffusion with  $T=5$  ( $T = 5$  for graphical reasons).
2. Uniform Sampler involves uniform sampling  $[1, T]$ .
3. Batch: At the bottom left, we find an example of a drawn batch (batch size = 4). This batch provides us with different levels of noise at the extracted time  $T$ . The batch, noised according to the rules of the previously defined forward process, is passed to 4.
4. Denoising UNet model, which returns 5.
5. Prediction. This prediction is then compared with the original input in 6.
6. Compute Loss.

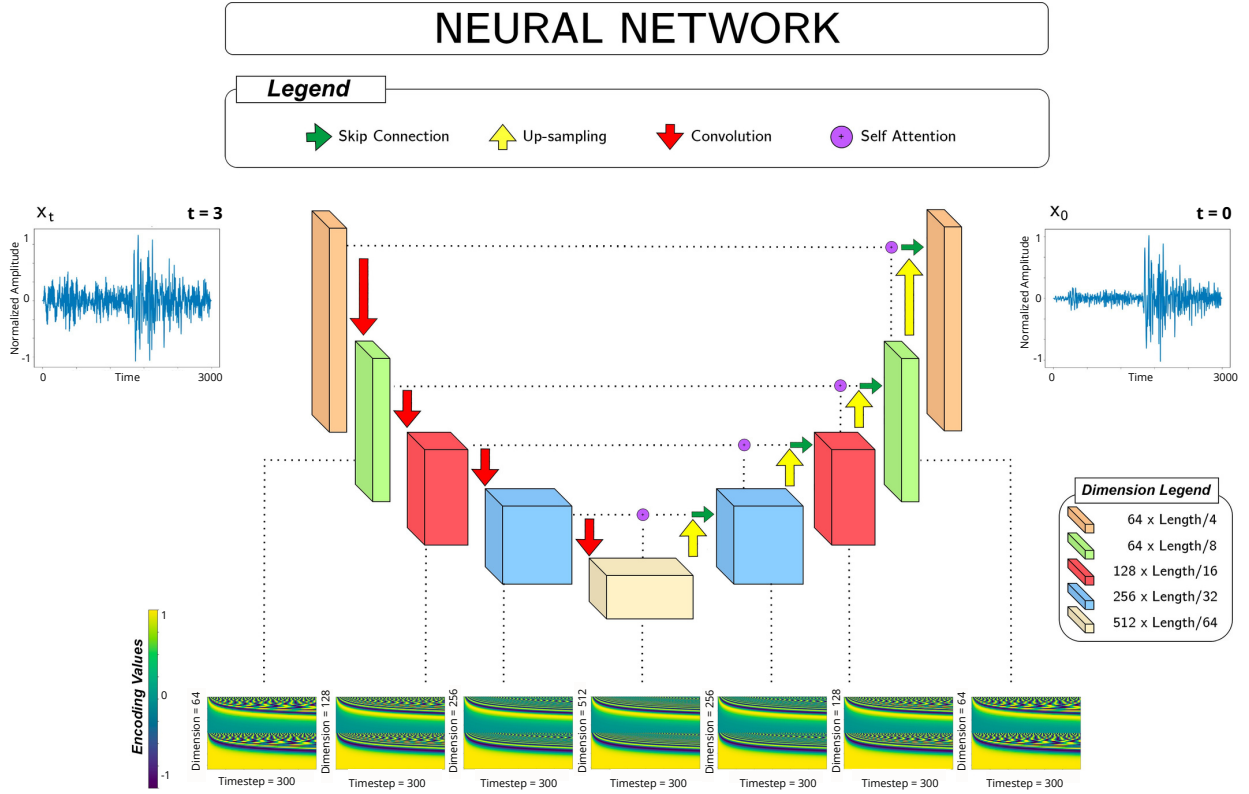
In our seismic denoising approach, we separately normalize the noise and *EQ* data. We adopt a trace-specific method, normalizing each seismic trace (*EQ* and noise) across its East-West (E), North-South (N), and Vertical (Z) channels. This normalization process aligns the maximum and minimum values within these channels, standardizing the data to a range of  $[-1, 1]$ . Such an approach ensures that each component retains its relative amplitude, enabling precise and balanced analysis. This also enhances generalizability for each type of seismic trace that the end user wants to denoise.

In the training phase for each seismic trace, we begin by merging a normalized *EQ* trace with a normalized noise trace. The noise component is scaled using the *NRF*, adjusting its intensity in the noisy signal before the forward process is applied.

The creation of the 'noisy signal'  $x_T$ , a combination of the *EQ* and scaled noise signals, leads to

the 'forward process' see Fig. 2.1. Here, a stochastic variable  $t$ , ranging from 0 to a predetermined maximum  $T$ , is chosen for further noise modulation. At  $t = 0$ , we have a recorded  $EQ$  signal with no additional noise, whereas at  $t = T$ , the noise is at its full scale.

### Model Configuration



**Figure 2.2:** *CDiffSD* combines convolutional layers, *Residual Neural Network (ResNet)* blocks, attention mechanisms, and positional encoding (with positional encoding detailed in the bottom plots) for effective one-dimensional data processing.

In this subsection, we detail the model Figure 2.2 described in step 4, the Denoising UNet model, of Fig. 2.1. As a building block of the *DM*, we adopt a *Neural Network (NN)* model inspired by the 1D U-Net [135] design, for the processing of one-dimensional data streams such as time series or audio signals. The network begins with 1D convolutional layers, each equipped with 64 filters of a kernel dimension of 7, instrumental for the initial extraction of salient features. This is followed by the integration of temporal processing units that leverage sinusoidal positional encoding to effectively capture the temporal intricacies inherent within the data. These units then employ two linear layers dedicated to feature refinement and are paired with the GELU (Gaussian Error Linear Unit) [61] activation function to instill the requisite non-linearity. As the architecture progresses, it introduces dimensionality manipulation layers consisting of *ResNet* modules [60] pivotal for feature conservation during downsampling and 1D convolutional layers for further data refinement. Post-downsampling, a series of upsampling layers are implemented, designed to elevate data dimensionality by merging *ResNet* blocks with dedicated upsampling operations. A noteworthy feature of our design is the mid-level blocks, each outfitted with dual residual units. They exploit attention mechanisms important for highlighting pertinent data characteristics. The network culminates with

terminal residual blocks that are succeeded by 1D convolutional layers, making definitive outputs typically manifest as singular channels.

The U-Net block is applied for each iteration of the  $DM$  from each  $t_i$  to 0 and then again from  $t_{i-1}$  to 0 and so on until the end of the process.

We trained models with 3 configurations:  $T = 20, T = 100, T = 300$ . These diverse scheduler assumptions allowed us to evaluate how performance metrics vary with increasing  $T$ , highlighting the trade-off between model performance and computation time, which is a crucial consideration in seismic monitoring room operations where balancing processing speed and precision is essential.

Particularly in the inference phase, understanding the impact of  $T$  on both model performance and computational efficiency is vital. For applications requiring rapid trace processing, like real-time seismic monitoring, a preference for speed may be necessary, though it could impact precision. Conversely, in tasks where accuracy is the priority, such as dataset cleaning, a greater emphasis on precision may be warranted, even at the expense of longer processing times.

We compared our approach using the same seismic dataset with  $DD$ , that we consider as benchmark. For this task,  $DD$  underwent comprehensive training for 400 epochs, while our model completed its training in just 150 epochs. This difference was due to our model’s learning dynamics and efficiency. We initiated our model’s training with a learning rate of 1e-3 and employed a scheduler to reduce this rate gradually, ensuring controlled and stable convergence.

### Inference with Direct and Sampling Reconstruction

Cold  $DM$  involve distinct methods to reconstruct the signal including the adoption of direct or sampling reconstruction. These methods represent approaches within the framework of  $DM$ , each with unique operational mechanisms and implications for model performance. Understanding the nuances of these methods is vital for comprehending the overall efficacy and application potential of  $DM$ .

For the range of configurations used in training our models ( $T = [20, 100, 300]$ ), we applied these configurations to both direct and sampling reconstruction. In the context of  $DM$ , the distinction between ‘direct’ and ‘sampling’ approaches is pronounced, marked by their differing operational mechanisms.

The ‘**direct**’ method involves applying the reverse process using the U-Net architecture to transition from a specific timestep  $t_n$  directly to zero. Conversely, the ‘**sampling**’ method incrementally applies this reverse transition from a specific timestep  $t_n$  to zero, but crucially, it traverses through all intermediate timesteps  $t_i$ , where  $i \in [n - 1, 0]$ . This results in applying the U-Net architecture multiple times ( $n$ ).

A key aspect of the cold diffusion paradigm is evaluating the effectiveness of the sampling procedure, which is hypothesized to outperform the direct approach. If the direct method, particularly using U-Net alone, yields comparable results, it would call into question the necessity of the complex training infrastructure typically associated with  $DM$ . We provide a detailed comparison between the direct and sampling methods in section 5.3.



## Metrics

For enhanced clarity, we define here the metrics used in our study now and then in Section 5.3 we provide a detailed commentary on the results.

1. *SNR* is a measure used to compare the level of a signal (*EQ* in this case) to the level of background noise. A higher *SNR* indicates that the seismic signal stands out clearly from the background noise, facilitating accurate analysis and interpretation. We defined *SNR* as in [187]:

$$10 \log_{10} \frac{\sigma_{signal}}{\sigma_{noise}}.$$

where  $\sigma_{noise}$  and  $\sigma_{signal}$  are the standard deviation of waveforms before and after the P arrival, respectively.

2. *CC* is a widely used measure of similarity between two signals. We compute the zero-lag *CC* between the recorded *EQ* signals (before noise is added) that represents our ground truth  $x_0$  and the denoised ones to evaluate the performance of the different models in reconstructing the recorded waveform.
3. To evaluate the picking performances of the proposed method, we applied the *DL* phase picker PhaseNet [185] to the waveforms and compared the retrieved arrival times with the labeled picked phases of the catalog ( $\sim 70\%$  of manually picked and  $\sim 30\%$  of automatic picked). In this way we can assess the impact of the denoiser on P and S arrival determination, the accuracy of which enables the calculation of a well constrained location.

We evaluated picking performance by analyzing the distribution of time differences between picks identified by PhaseNet on the denoised traces and the labeled picks within the *Stanford Earthquake Dataset (STEAD)* dataset. Additionally we employed a "recall" metric that is calculated as the number of picks falling within  $\pm 50$  samples of the nearest labeled pick, divided by the total number of labeled P or S arrivals.

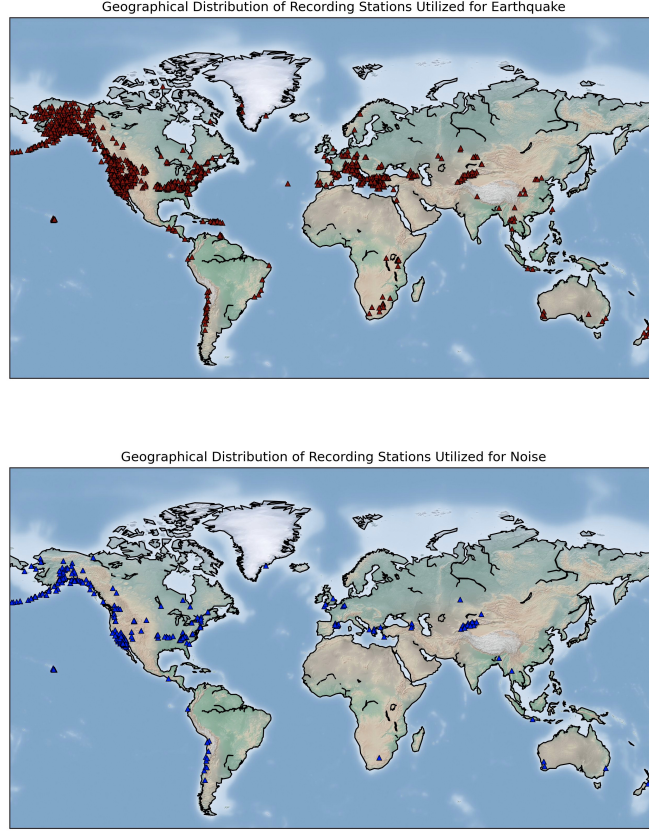
## 2.3 Data Sources and Selection

In our study, we focus on a subset extracted from the *STEAD* [116]. This section is dedicated to elucidating the composition of the subset, detailing the following components:

1. We selected specific seismic stations to gather *EQs* and others for noise, with some overlap, providing a clear trace of the data's origin for our analysis (Fig. 2.3).
2. The distribution of seismic events across the globe (Fig. 2.5) is mapped out, with these events sorted into training, validation, and test sets. This classification helps us to assess the model's effectiveness and its generalizability across different regions.
3. We applied constraints to the dataset, including the magnitude (Fig. 2.4) and proximity to the seismic stations.

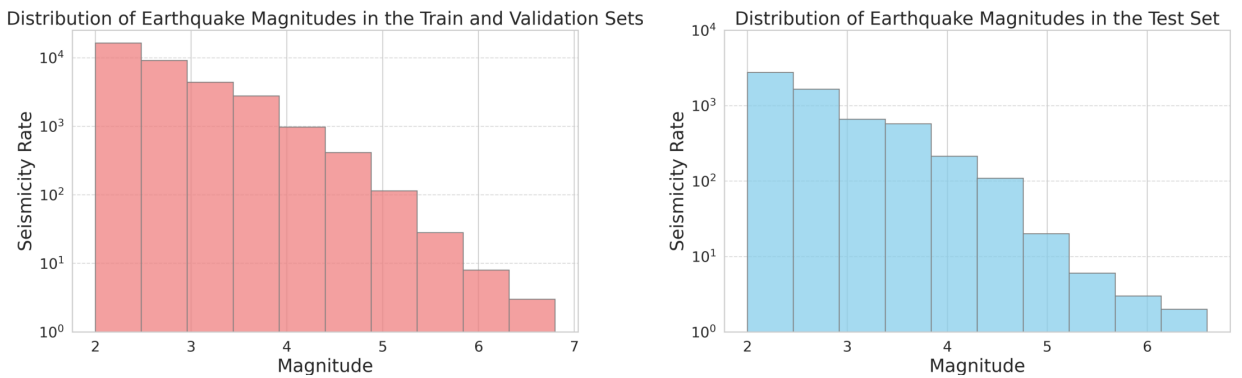
*STEAD* features a significantly larger number of stations for *EQ* data compared to those used for noise. Moreover, the majority of these stations are concentrated within the U.S. territory. In our

study, we utilize a ratio of (1786/2613) stations for the extraction of *EQ* data, representing a fraction of the total available. For seismic noise, we have selected a subset corresponding to 306 stations dedicated to noise recording.



**Figure 2.3:** The maps show the subset of stations of the *STEAD* used for the recorded *EQ* signal (upper) and the recorded noise (bottom).

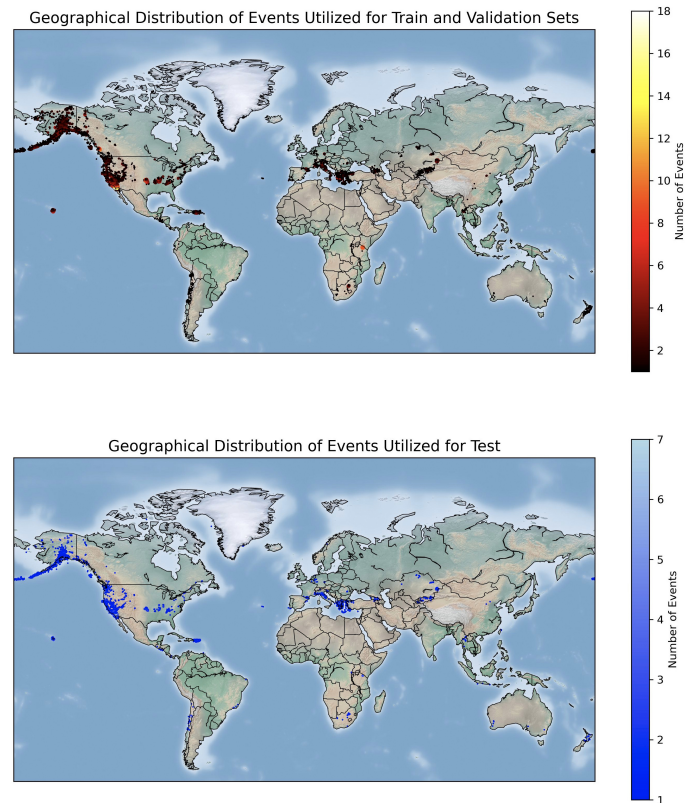
Throughout our analysis, we consistently sample seismic traces of 30-second durations, based on the following criteria: magnitude  $> 2$ , *EQ*-station distance  $< 100$  km, and P-wave arrival after 7 seconds. Figure 5 shows the frequency-magnitude statistics for our data set.



**Figure 2.4:** The histograms illustrate the frequency distribution of *EQ* magnitudes within our dataset, with the left panel representing the training and validation sets and the right panel the test set.

We chose an inclusive approach for training, leveraging the full spectrum of available data, without any *SNR* selection criteria. While this might seem disadvantageous initially, a model that performs well under these conditions can be versatile across various scenarios. For researchers looking to retrain this model on their datasets, especially when specific datasets are limited, it may be advantageous not to put restrictive filters such as *SNR*.

Our dataset was divided into training (30491 traces), validation (3441 traces), and test (5994 traces). Such a division in *Machine Learning (ML)* ensures model reliability and generalizability. The training set aids the model’s primary learning, the validation set is used for hyperparameter adjustments, and the test set objectively evaluates the model’s performance on unseen data.



**Figure 2.5:** The image presents two maps of the geographical distribution of seismic events used in our study, with the upper map illustrating the events for the training and validation sets marked in red, and the bottom map showing the events for the test set in blue. The color intensity on each map corresponds to the number of events, with darker shades indicating a lower concentration of events in that location.

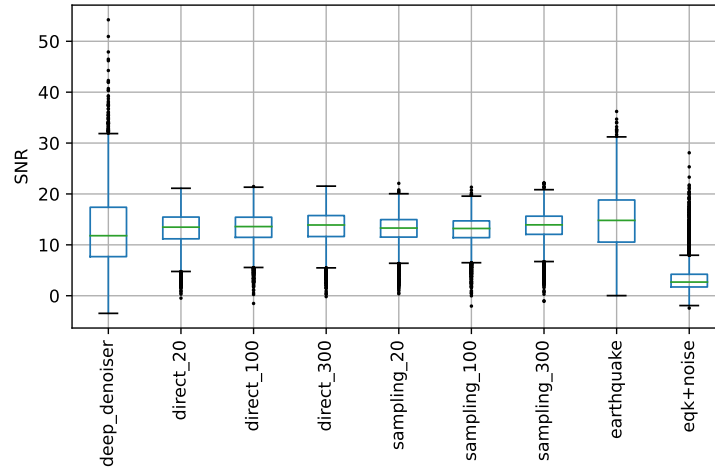
## 2.4 Results

In the following we present our results and discuss the validity of our model by adopting quantitative and qualitative categories. The metrics used for each are provided in Section 2.2.1.

### 2.4.1 Quantitative Results

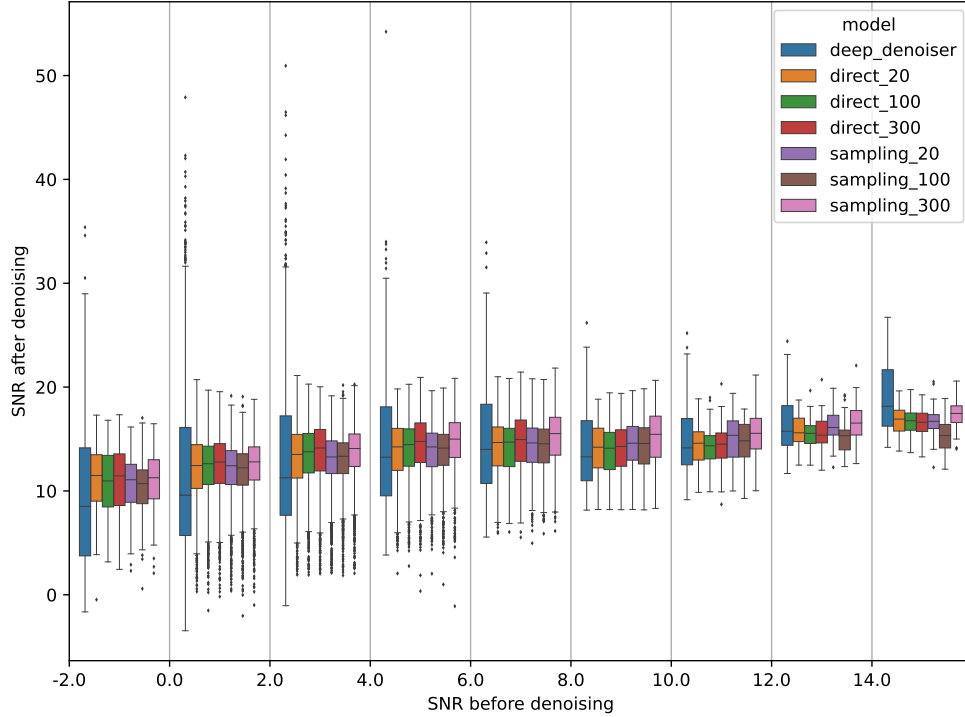
#### Signal to Noise Ratio (SNR)

A comparison of the  $SNR$  metric for the denoised waveforms obtained with different models and configurations is shown in Fig. 2.6. Note that Figure 2.6 includes the same metric for the original  $EQ$  signals (labeled "earthquake") and those with added noise (labeled "eqk + noise"). The latter are the inputs to the denoiser algorithm. The performances of the different models appear aligned, with  $DD$  differing by a slightly lower median but greater variability in output  $SNR$ .



**Figure 2.6:**  $SNR$  comparisons using box-plots for various model configurations applied to the test set. The original signals ("earthquake") and the ones with added noise ("eqk+noise") have respectively the higher and lower  $SNR$ , as expected. The different denoising models appear overall aligned, with direct and sampling showing slightly higher median values and tighter distributions with respect to  $DD$ .

In Fig. 2.7 we classified the noisy observations as a function of the  $SNR$  before denoising to highlight the effectiveness of our models in cleaning the seismic traces. The performance of our  $CDiffSD$  are consistently superior with respect to  $DD$  in low  $SNR$  scenarios. This aspect is crucial, given that low  $SNR$  conditions correspond to more complex and heavily noisy seismic traces precisely where an effective denoising solution is most needed. The high-quality performance of our model in these low  $SNR$  environments is demonstrated in Fig. 2.7. We note in particular model reliability and efficacy in extracting correct signals from noisy data. This proficiency is important in real-world seismological applications, especially for discovering lower magnitude  $EQ$ s often hidden in the noise.

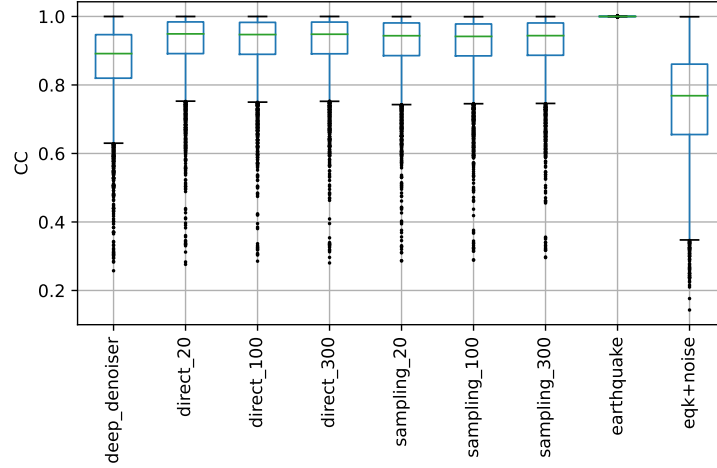


**Figure 2.7:** Distributions of  $SNR$  values of denoised waveforms for different ranges of input  $SNR$ . The  $SNR$  statistics after denoising are computed on 2dB wide ranges of input  $SNR$ .  $CDiffSD$  models show higher performances in low  $SNR$  scenarios, while  $DD$  is superior for the higher  $SNR$  signals. We study the range:  $SNR < -2.0$  and  $SNR > 16.0$ , which covers 99% of data. Solid bars within each model (color) show the median value.

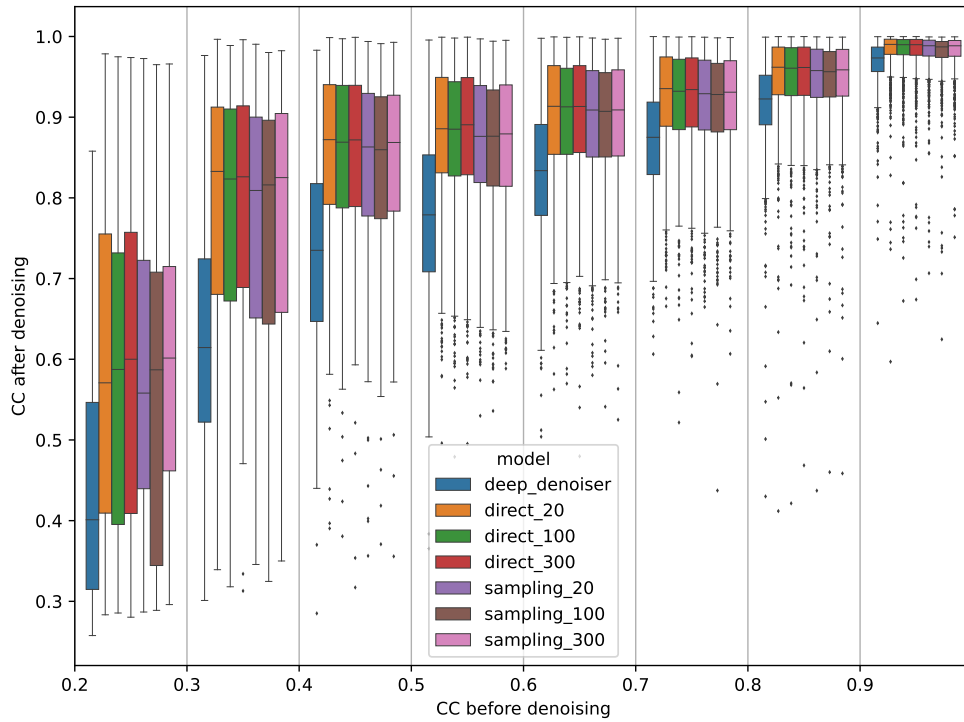
While the cold diffusion approach excels in low  $SNR$  scenarios, the binary mask-based method  $DD$  exhibits greater variability and tends to perform better in higher  $SNR$  conditions, benefiting from its ability to provide a clear-cut signal delineation (Fig. 2.6 and Fig. 2.7). In particular,  $DD$  shows improved performance when the input  $SNR$  is higher than  $\sim 14$  and is get worse at lower input  $SNR$  while our models remain consistently effective for a large range of input  $SNR$ . An example of high input  $SNR$  conditions can be found in Appendix A.0.4.

### Cross Correlation

We evaluate the similarity between original signals and denoised signals, by showing the statistics of the computed  $CC$  values, in Fig. 2.8. A higher  $CC$  indicates a greater similarity between the denoised trace and the original signal. In this figure, we see that all  $CDiffSD$  models show similar performance and they are all consistently higher than  $DD$ . To better highlight the variability of  $CC$  values obtained from the different traces of the test set, in Fig. 2.9 we show the distribution of  $CC$  values between denoised and original traces as a function of  $CC$  of the noisy traces with original signals (x axis), that is,  $CC$  of traces before denoising is applied. The performances for both direct and sampling are higher than  $DD$  for every considered range of  $CC$  before denoising.



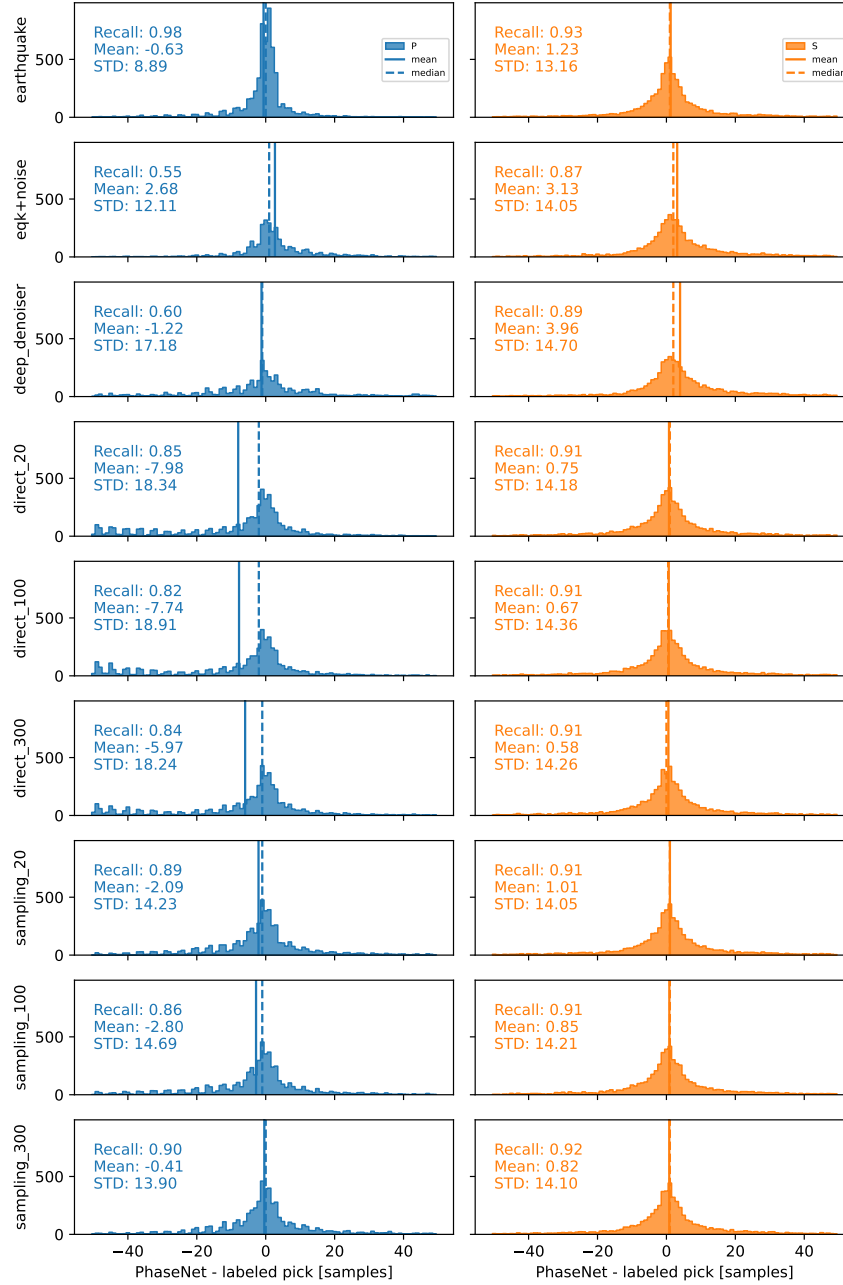
**Figure 2.8:**  $CC$  comparisons for various model configurations applied to the test set. Higher  $CC$  values indicate greater similarity between the denoised trace and the original signal. All  $CDiffSD$  models show similar performance and that they are consistently higher than  $DD$ .



**Figure 2.9:** Distribution of  $CC$  values between original and denoised signals (y axis) as a function of  $CC$  before application of denoising (x axis).  $CDiffSD$  models outperform  $DD$  across all  $CC$  ranges. The difference is more noticeable especially at low pre-denoising  $CC$  values. Statistics are computed for ranges of 0.1. Note that the distribution of samples for 'CC before denoising' is identical to "eqk+noise" in Fig. 2.8. Consequently, the 0.1-wide bins may encompass significantly varying numbers of samples.

For each model considered we see better performance, with higher values of  $CC$  after denoising (Fig. 2.9). Another noteworthy aspect is that at higher noise levels, thus lower  $CC$  before denoising (values from 0.2 to 0.3), models with  $T = 300$  outperform their counterparts. As expected, these performance disparities tend to converge with an increase in  $CC$  before denoising, corresponding to a relative reduction in noise compared to the signal.

## Phase arrival picks



**Figure 2.10:** The histograms display the distributions of P-wave (blue) and S-wave (orange) arrival time differences between automated PhaseNet detections and label picks (in samples). The results obtained using the original seismograms, "eqk+noise" and *DD* are shown for comparison in the first, second and third row, respectively. The remaining rows show the results for different *CDiffSD* models applied to the same data subset, offering insights into the accuracy of wave arrival time detection by each model. Central tendency metrics, such as mean and median, are indicated in these histograms, highlighting any potential skewness in the distribution towards either early or late picks for both P and S waves.

The histograms in Fig. 2.10 provide a visual representation of the efficacy of different seismic signal denoising methods — "direct", "sampling", and *DD* — in retrieving a signal and preserve P- and S-

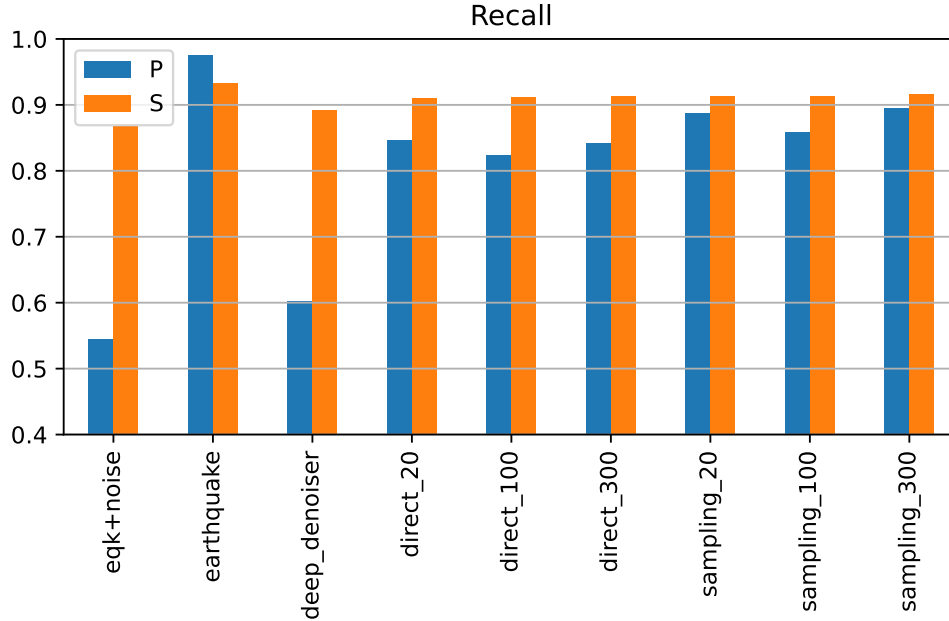
wave onsets. The accuracy of automated P and S-wave arrival time picks by PhaseNet is compared to label picks. The histograms are organized by method and parameter variations, displaying the distribution of arrival time discrepancies measured in samples.

In the case of *EQ* (i.e., no noise added, top histogram), the P-wave pick difference distribution exhibits spreads that are narrower than those of the S-wave and this is in full agreement with the expected behavior.

When noise is introduced, the pick difference distributions for P-waves and S-waves tend to converge towards a more similar pattern. This convergence can be attributed to the primary impact of noise on P-waves, owing to their lower amplitude compared to S-waves. As a result, the performance with added noise on P waves detection is much more degraded than on S waves detection with the same level of noise because P-waves have also smaller amplitudes. This observation is further supported by the recall values for S waves, which remain greater than 0.85 not only for all the denoising methods, but also for the noisy traces ("eqk+noise"). In contrast, the recall rate for P-waves is consistently lowered by the presence of noise (Fig. 2.11). For these reasons we focus our analysis on P-wave picks.

As seen in Fig. 2.10 the distribution of the "direct" methods show pronounced negative skews, with mean values far from 0. This indicates a tendency of PhaseNet to pick P-waves slightly before the labeled picks for the waveforms denoised with "direct" methods. The reason of this behavior is most likely to be attributed to noise remaining in the denoised traces processed with the "direct method". This in turn can mislead PhaseNet to an early detection (see the "direct" example in Fig. 2.12). This tendency, however, is mitigated completely by the *CDiffSD* "sampling" method, as shown in Fig. 2.12. In particular, we see that the "sampling" methods display recall rates that are consistently high for both P and S, especially the 300 configuration, indicating a good denoising performance and the ability to recover the labeled phases.





**Figure 2.11:** Comparison of recall rates for P and S waves between the different methods within a fixed window of 50 samples. S-waves recall rates are aligned almost for all models, indicating that the noise level is not enough to affect the S-waves because of the greater amplitude. P-waves recall rates instead show significant disparities between the *DD* approach and other methods, suggesting a lower performance of *DD* in preserving the P onset in these cases. The 'sampling-300' method is confirmed as the one with better performances.

From the comparison of the results obtained with the "direct", "sampling", and *DD* methods, it is evident that each method influences the automated pick accuracy differently. The "sampling" method, particularly at higher parameter settings, demonstrates a notable alignment with label picks, suggesting its superiority in mitigating noise and enhancing the precision of automated picking systems. It is also noteworthy that the recall values for P-waves shown in Fig. 2.11 are higher than *DD* for both "sampling" and "direct" methods, which suggests that in these cases *DD* does not preserve accurate P-wave onsets.

A comprehensive evaluation that considers all the proposed metrics in conjunction is essential for gaining a clear understanding of the various methods' performances. While examining each metric individually offers valuable insights into specific aspects of performance, a truly clear picture only emerges when we analyze these metrics together. For instance, *SNR* alone offers no insight into denoising quality. This metric turns out to be the least informative in our analysis, as evidenced by the lack of clear separation between methods. Conversely, *CC* assesses the similarity between original and denoised signals, providing a valuable but general measure of output quality. Here, *CDiffSD*'s improvements are evident. Finally, the picking analysis tackles the crucial aspect of seismic wave onsets, focusing on the critical waveform portion where the noise-to-signal transition requires careful handling. Here the improvements of sampling versus direct and *DD* methods are clearly highlighted. A summary of the quantitative results discussed above is presented in Table 2.1.

| Model               | SNR<br>(median) | CC<br>(median) | P picks diff<br>(mean, samples) | P picks diff<br>(STD, samples) | P picks<br>(recall) |
|---------------------|-----------------|----------------|---------------------------------|--------------------------------|---------------------|
| DD                  | 11.796          | 0.891          | -1.22                           | 17.18                          | 0.60                |
| CDiffSD direct20    | 13.476          | 0.949          | -7.98                           | 18.34                          | 0.85                |
| CDiffSD direct100   | 13.589          | 0.947          | -7.74                           | 18.91                          | 0.82                |
| CDiffSD direct300   | 13.900          | <b>0.948</b>   | -5.97                           | 18.24                          | 0.84                |
| CDiffSD sampling20  | 13.298          | 0.943          | -2.09                           | 14.23                          | 0.89                |
| CDiffSD sampling100 | 13.216          | 0.941          | -2.80                           | 14.69                          | 0.86                |
| CDiffSD sampling300 | <b>13.928</b>   | 0.943          | <b>-0.41</b>                    | <b>13.90</b>                   | <b>0.90</b>         |

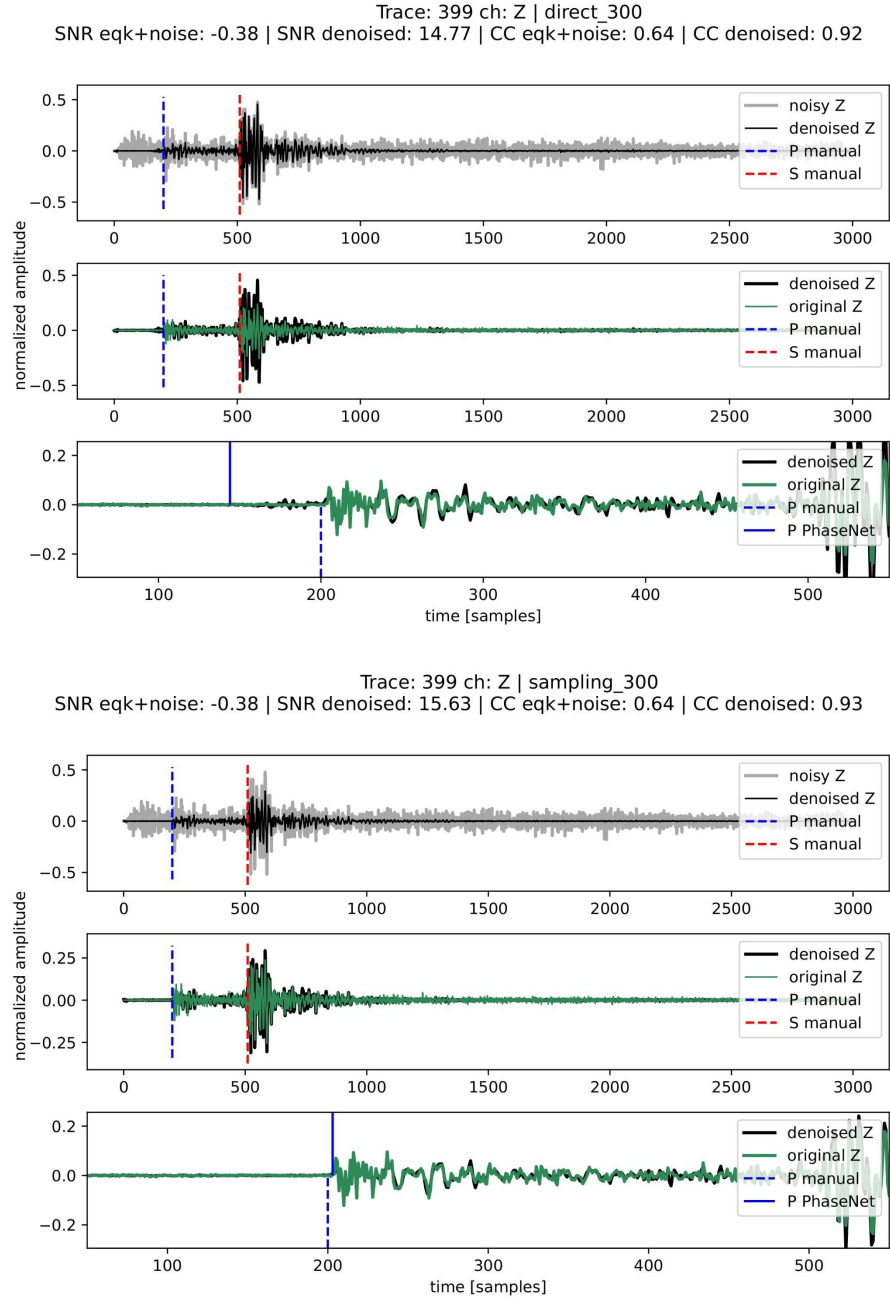
**Table 2.1:** Summary of the metrics obtained with different denoising methods. Best score for each metric in bold.

### 2.4.2 Qualitative Results

Qualitative factors are useful as side-by-side comparisons between the outcomes of different approaches. In Appendix A.0.4, we delve deeper into the analysis of seismic traces, examining the impact of denoising on picking. This includes a thorough examination of both the strengths and limitations of our model. We highlight instances where our model excels in denoising, as well as situations in which it does not perform optimally.

The examples below and those in Appendix A.0.4 are organized with the same layout: in the top panel we compare the noisy signal (grey) with the denoised signal (black); in the middle panel we compare the original signal (green) with the denoised signal (black); the bottom panel is a zoom on the P-wave arrival.

### Qualitative Picker Analysis: Direct Vs Sampling

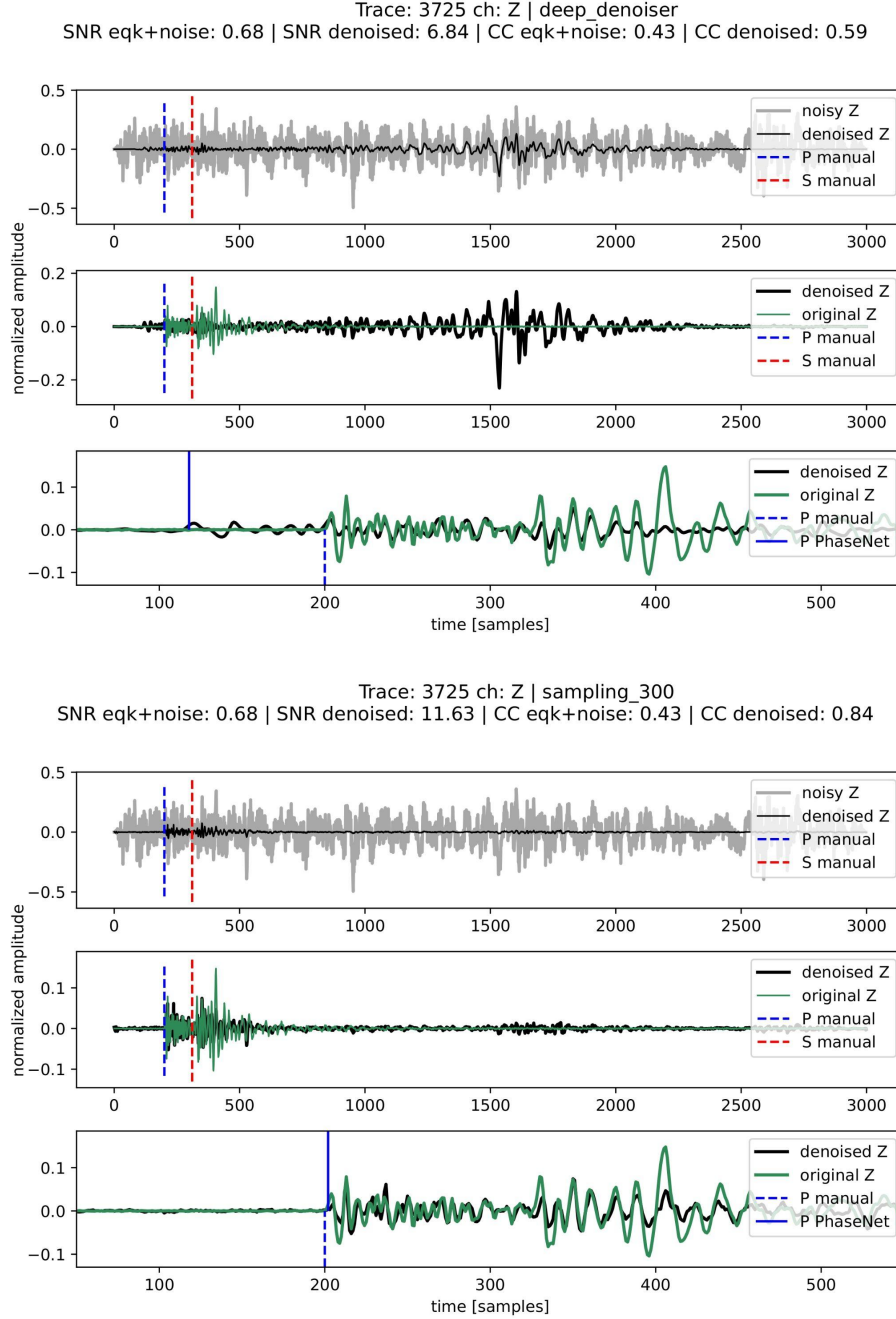


**Figure 2.12:** Comparison of a trace processed using 'direct\_300' and 'sampling\_300' methods. Notably, the 'direct\_300' retains some of the noise preceding the P wave arrival, which is instead filtered out in the 'sampling\_300' results. This noise before the P-wave retained with the the 'direct\_300' method explains the tendency for this model to cause early picks (as seen in Fig. 2.10), as the residual noise can lead to earlier detections.

In the first example shown in Fig. 2.12 we compare the 'direct\_300' and the 'sampling\_300' methods. Here "sampling" method is found to be more effective than the "direct" method in denoising the seismic signal, and this is particularly evident from the middle and bottom panels, where the denoised signal in the "sampling" method match more closely the original signal. In

contrast, the "direct" method shows more significant deviation from the original, especially before the P-wave arrival. This example is also useful because it provides insight into the tendency of the "direct" methods to cause spurious early P-picks. The direct method, in fact, retains some pre-arrival noise, which can trigger an early pick in automatic approaches such as PhaseNet. This is less of an issue in the sampling method, as seen in the lower set of traces, where the denoised signal is cleaner, and the P-wave arrivals are closer to the labels. The implication for seismic processing is significant since the sampling method appears to produce cleaner signals and more accurate P-wave arrival times as a direct consequence. We note that this is crucial for various seismological applications such as *EQ* location and tomographic imaging.

### Qualitative Picker Analysis: DD Vs Sampling

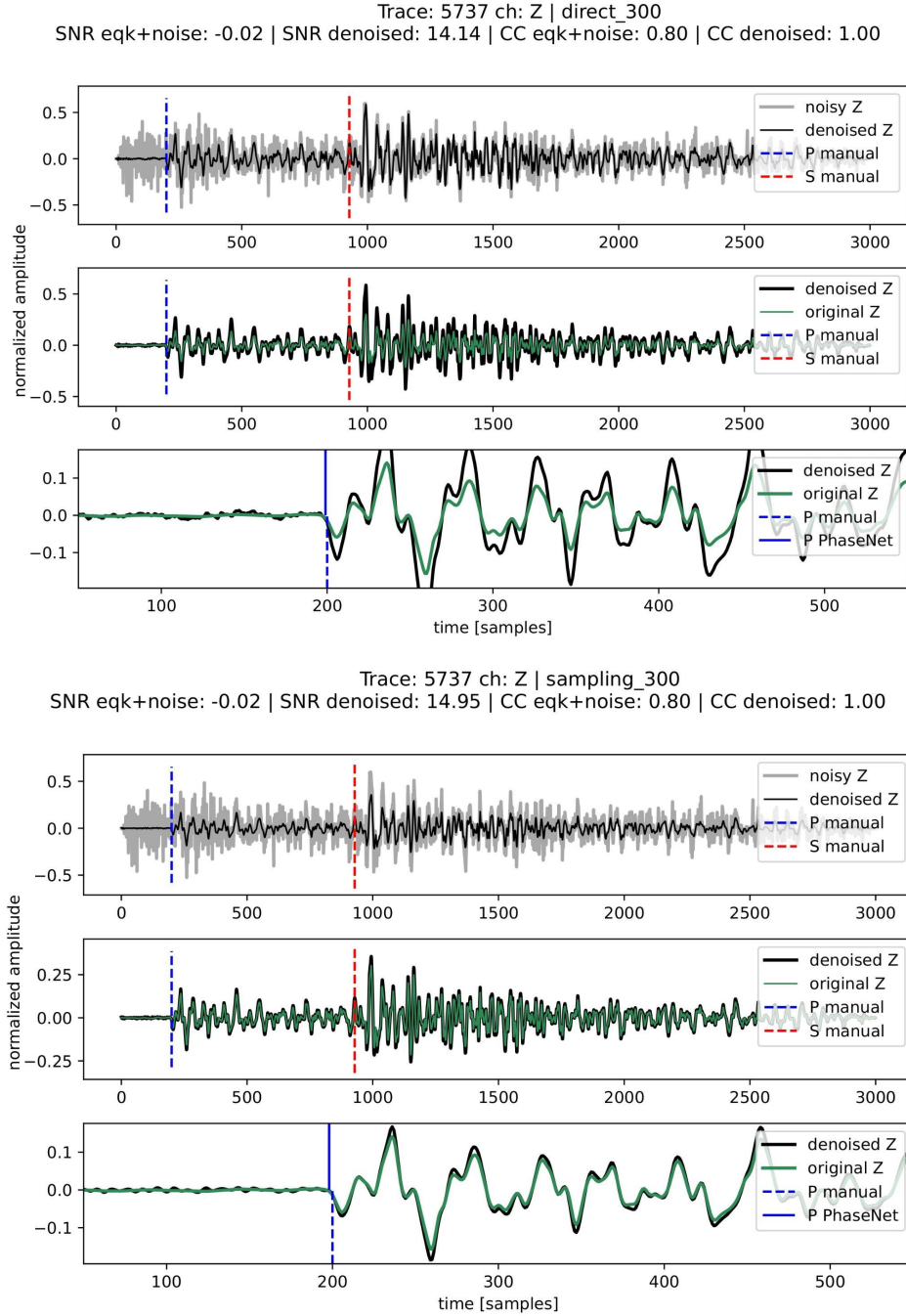


**Figure 2.13:** Comparison between a seismic trace processed with 'deep\_denoiser' and 'sampling\_300' methods. The 'sampling\_300' method demonstrates a closer match to label phase picks and a more precise amplitude preservation, despite the substantial noise present in the original signal. *DD* also retains a high amplitude noise signal at around 1500 samples that 'sampling\_300' manages to filter out almost completely.

Fig. 2.13 exemplifies the concepts previously discussed in Fig. 2.7, highlighting the performance of our model compared to that of *DD* in scenarios with very low *SNR* before denoising. The figure demonstrates clearly how an extreme noise situation can lead to an error in phase picking for *DD*, whereas the 'sampling' method is capable to reconstruct accurately the correct P wave arrival despite

the presence of significant noise.

### Qualitative Amplitude Analysis: Direct Vs Sampling



**Figure 2.14:** Comparison of a seismic trace processed with 'direct\_300' and 'sampling\_300' methods. It is particularly significant that the 'sampling\_300' technique demonstrates an enhanced ability for amplitude reconstruction compared to the 'direct\_300' method.

In Fig. 2.14 we show a comparative analysis of seismic signal denoising methods to investigate the importance of amplitude preservation. The Cold *DM* employing a sampling strategy ('sampling\_300') demonstrates a superior performance in maintaining the amplitudes of the seismic

signal. In practice, the denoised signal aligns more accurately with the original waveform, preserving the integrity of the amplitude across the signal’s duration. This is particularly evident in the detailed zoomed-in analysis, where the ‘sampling\_300’ method displays remarkable congruence with the original signal, as evidenced by the minimal and consistent residuals. In contrast, the direct application of a U-Net model (‘direct\_300’) displays a slight but discernible attenuation in amplitude, most noticeable in segments with higher amplitude peaks. The increased residuals associated with the ‘direct\_300’ method suggest a more significant alteration of the signal after the denoising process. Therefore, the Cold *DM* with sampling stands out as the most effective method for seismic data denoising (amongst those tested here), especially where the preservation of amplitude is critically important.

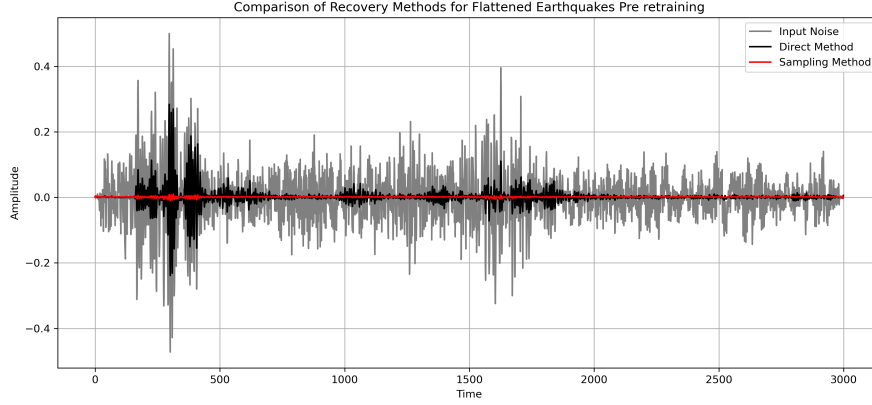
## 2.5 Model assessment: Assessing the Impact of Exclusive Noise Input

In this section we aim to test the behaviour of the model in no-*EQ* scenarios, i.e. with inputs containing only noise. This is done in order to verify whether the model doesn’t generate any artifacts in the absence of signal generating false *EQ*s.

Cold Diffusion is based on the model’s ability to learn the broad data distribution during training, which generally includes a variety of seismic traces with different levels of noise. Therefore, the model should be able to generalize and identify traces that are entirely dominated by noise, even without direct exposure to specific types of *EQ* samples where there is no *EQ* signal. Based on these assumptions, we seek to verify if our results align with the theoretical expectations.

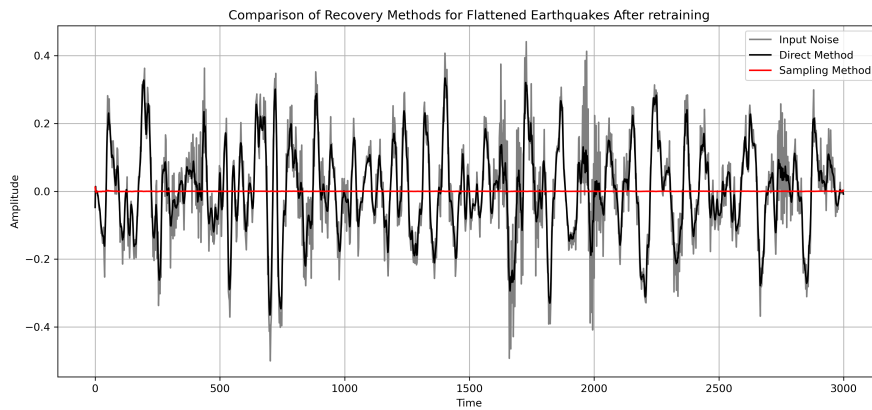
We have used the entire noise test set as input, without combining it with the *EQ* data. Theoretically, with a perfect denoising, the expected output would be a trace composed exclusively of zeros, in the real context the trace should approach zero.

We applied the model without retraining, meaning the model’s weights have never been exposed to the absence of *EQ* traces as ground truth. To assess the correctness of the output we set an amplitude threshold between  $\pm 0.02$  to decide whether the output could resemble a trace of zeros. The direct and sampling methods have correctly reconstructed the expected signal in 60.3% and 88.6% of cases, respectively. This different performance highlights the sampling method’s superior capability in recognizing the absence of *EQ* signals and adapting to it.



**Figure 2.15:** Example of the outputs of direct (in black) and sampling (in red) methods in case of a noise only input (in grey). No retraining is performed here, i.e. the models have never been exposed to zero-traces as ground truth for noise-only input. The direct method fails in recovering a zero-trace since it introduces artificial signals. In contrast, the sampling method reconstructs successfully an output that resembles a zero-trace.

Given the promising results just described, we further explored this scenario by retraining the model including no-signal traces as ground truth. We focused only on a single channel for this test and incorporated 3% of the entire training set with zeroed traces to represent the absence of seismic events. The results align with our expectations, indicating an improvement in performance in the presence of noise alone. Specifically, the cases where zero traces are retrieved increases to 68.2% and 90.5% for direct and sampling methods, respectively. The direct method exhibits a more substantial improvement, starting from a lower baseline performance, whereas the sampling method shows a smaller increase, likely due to its performance already approaching saturation.



**Figure 2.16:** Example of the comparison between the sampling method (in red) and the direct method (in black), the input (in gray) for both methods is only noise. In this case the models have been retrained with zero-traces as ground truth for noise-only traces. The sampling method succeeds in reconstructing a zero-trace. On the other hand, the direct method outputs noise, indicating a less accurate reconstruction in this scenario.

Regarding the results post-retraining, it should be noted that the output trace of the sampling method shown in Figure 2.16 is indeed close to the expected zero-trace. On the contrary, low amplitude noise was still present in the output of the non retrained-case shown in Figure 2.15. This



highlights the importance of including flat traces during the training.

In this evaluation of the *CDiffSD* on these cases comprised solely of noise, we proved that it is not imperative to include such examples in training to accurately discern between noise and genuine seismic signals. However, including these kind of signals in training, improves the capability of effectively identifying traces that are comprised solely of noise.

## 2.6 Discussion

Our study demonstrates promising results and affirms the validity of cold diffusion denoising for seismological applications. We employed three key metrics to quantify the enhancement brought by the *CDiffSD* model. Specifically, the *CDiffSD* model showcased a substantial improvement over *DD* in denoising seismic traces (see Table 2.1), enhancing the *SNR* by approximately 18%. Furthermore, we observed a 5.84% increase in *CC*, indicating a higher congruence between the denoised signals and the original ones. Finally, as a third metric, our approach significantly enhanced the accuracy of seismic event detection, achieving a 50% improvement in recall for P-wave picks, indicating a much better preservation of the P-onset even for noisy seismic data. Regarding the evaluation of different *CDiffSD* versions, it is important to highlight that, despite *SNR* and *CC* metrics aligning between the "direct" and "sampling" configurations, the "sampling" systematically demonstrates its superiority in applied contexts, such as P-phase picking. Focusing on "direct" versus "sampling" we observe significant enhancements: e.g. comparing both configurations with  $T = 300$ , the average difference from labeled picks reduces significantly from -5.97 to -0.41 samples. Similarly, the standard deviation drops from 18.2 to 13.9. This trend of improved performance holds true across all levels of  $T$ . Moreover, "sampling" yields a notable 6.7% increase in P picks recall compared to the "direct" method. However, it is noteworthy that the computation time increases with an increase in  $T$ . A more detailed study is reported in Appendix A.0.6. Therefore, the size of  $T$  should be considered when using these models in real-time scenarios such as in a seismic monitoring room. We emphasize the importance of looking at the results as a whole. That is, while *SNR* and *CC* are important metrics for assessing, respectively, the raw denoising power and the quality of the reconstructed signal, in fact, the preservation of the integrity of the P- and S-wave arrivals is of critical importance for a reliable denoising technique. Among the models evaluated in Section 5.3, the one utilizing "sampling" with  $T=300$  emerged as the most effective according to the three combined metrics. The model's fidelity in preserving seismic trace characteristics, especially at the signal-to-noise transition, highlights its practical advantages in real-world seismological applications. Appendix A.0.5 presents comparisons with Wiener Filter, a traditional method, and an improved version of *DD* with Attention, namely: *AttDD*.

We note that while our results provide an important advance, they should be regarded as a preliminary step towards addressing a broader spectrum of open questions and potential model enhancements. A significant direction for future advancement lies in applying these techniques to broader datasets. Our initial explorations aimed to establish the feasibility of these methods.

Moving forward we could potentially develop a more generalized model by retraining on large datasets such as *Italian Seismic Dataset for Machine Learning (INSTANCE)* [104] and *STEAD*,

which encompass several million traces compared to the  $\sim 40k$  traces used in this study. The use of larger datasets would allow treatment of noise in a wide range of seismological contexts without the need for further retraining, thus significantly boosting model applicability and robustness across diverse seismic scenarios.

Our model exhibits significant potential for cleaning and enhancing seismic traces. Moreover, it holds promise for recovering *EQs* hidden by noise that may have eluded both human and automatic detection. Such capability could contribute to expanding seismic catalogs. While further refinements are conceivable, this method, which is borrowed from speech enhancement tasks, has proven its validity in the intricate domain of seismological analysis. This cross-disciplinary innovation underscores the model's versatility and suggests broader applicability in extracting and analyzing subtle seismic signals.

## Chapter 3

# Masked graph neural network for rapid ground motion prediction in Italy

In this Chapter, we present an enhanced version of a graph-based deep learning model for rapid ground motion prediction. Specifically, we revisit and expand *TISER-GCN*, a Graph Neural Network (GCN) designed to estimate peak ground motion intensity measures (IMs) directly from the first 10 seconds of raw three-component seismic waveforms, starting from the origin time of the earthquake. Notably, this prediction does not rely on any prior information about the event’s location, magnitude, or distance making it suitable for real-time applications.

The original implementation of *TISER-GCN* focused on a limited network of 39 stations in central Italy. Here, we significantly scale up the model to nearly 600 seismic stations, leveraging the broad coverage provided by the *INSTANCE* benchmark dataset. To ensure high-quality inputs, the data were carefully selected to minimize sensor saturation and maintain waveform integrity.

To improve prediction performance, we introduce a masking strategy that excludes stations where the P-wave arrives within the 10-second input window. This, combined with the integration of additional contextual information, leads to a notable reduction in mean squared error: up to 6% for Peak Ground Acceleration (PGA) and 5.5% for Peak Ground Velocity (PGV), compared to an unmasked baseline. Furthermore, the model achieves near-zero median residuals across all IMs, effectively correcting the systematic underestimation typically observed when using a standard ground motion model tailored for Italy.

These results show that the updated *TISER-GCN* retains the predictive power of the original model while removing the constraint of a fixed seismic network geometry. This flexibility makes it a valuable tool for early ground motion estimation in regions with varying station configurations or incomplete event metadata.

### 3.1 Introduction

The rapid and accurate assessment of earthquake generated ground shaking can significantly improve emergency response times and public awareness, mitigating the impact of seismic events. As a core topic in seismology and earthquake engineering, it focuses on estimating the intensity or time series of shaking at a target location. Ground-motion prediction techniques are essential for seismic hazard assessment, early warning, and post-event damage evaluation.

In recent years, artificial intelligence (AI) approaches have shown significant potential in enhancing early warning systems by improving the speed and accuracy of ground motion predictions. [21] applied artificial neural networks (ANNs) to estimate key seismic parameters, including peak ground velocity (PGV), epicentral distance, and earthquake magnitude using the acceleration, velocity, and displacement waveforms. In a complementary approach, [66] utilized support vector regression (SVR) to predict peak ground acceleration (PGA) by extracting six critical features from the initial P-wave's vertical component, such as peak acceleration, peak velocity, peak displacement, effective predominant period, cumulative absolute velocity, and the integral of the square velocity. [52] used a hybrid method coupling genetic programming and orthogonal least squares to model ground motion prediction from source parameters, paving the way for more sophisticated approaches like those explored by [39], who integrated ANNs for enhanced ground shaking predictions. [68] employed Random Forest, which uses an ensemble of decision trees, to predict ground-motion intensity by analyzing the initial P-wave data, taking advantage of the model's ability to handle noisy data and avoid overfitting. [69] used Gradient Boosting Regressor (GBR) to predict PGA, and hypocentral distance starting from P-wave features.

In particular, Machine Learning (ML) has been applied to the prediction of the final ground-motion intensity at a target site using the first few seconds of its observations [e.g., 151]. [124] used a deep learning model based on surrounding stations to improve real-time intensity prediction. [77, 78] and [179] applied Convolutional Neural Networks (CNNs) for the rapid prediction of earthquake ground shaking intensity using only the first few seconds of raw waveform data. [49] combined CNNs with Voronoi tessellation to reconstruct ground shaking fields. CNNs are well-known for their ability to extract meaningful patterns directly from raw data, without relying on handcrafted features. This makes them particularly effective for processing seismic waveforms, which are often complex and noisy. CNNs can learn time-frequency features linked to shaking intensity, enabling fast and accurate predictions of ground-motion parameters, even without prior knowledge of source location or magnitude. [119] developed a transformer-based earthquake alert model, a multi-station approach that predicts final PGA values at multiple target stations using initial raw records at multiple input stations. [167, 168] demonstrated the power of Long Short-Term Memory (LSTM) networks, to improve the prediction process by capturing both short-term changes and longer-term trends in the seismic data, ensuring that the model continuously refines its predictions as more data becomes available. Graph Convolutional Networks (GCNs) have recently emerged as a powerful tool for modeling spatially distributed systems, including seismic networks. Unlike traditional machine learning models that treat input stations independently, GCNs leverage the spatial relationships between stations by representing them as nodes in a graph, where edges encode physical proximity or other meaningful connections. This structure allows the model to integrate information across neighboring stations, enabling more spatially coherent and accurate predictions. Moreover, GCNs are inherently robust to missing data and can generalize across varying station geometries, making them highly suitable for real-time applications such as earthquake early warning and rapid response systems. In the context of earthquake ground motion prediction, [17] demonstrated that GCNs can outperform conventional methods in both speed and accuracy, by using early waveform data to estimate shaking even at stations where the peak ground motion has not yet been recorded.

In this context, we start from the work by [17] as a departure point and illustrate how to go beyond the limitation of a fixed set of seismic stations that needs to be defined at training

time, using an updated GCN model for our task. Our implementation uses a dynamic masking mechanism, first introduced by [177] in the context of image recognition with incomplete inputs. Instead of propagating all node attributes, the Masked GCN selectively propagates only a subset, controlled by an event-based mask assigned to each node. A binary mask tensor dynamically determines which stations contribute to the prediction, based on whether the P-wave arrives within a 10-second window from the first triggered station. This selective inclusion ensures that only early, informative waveform data are used, thereby reducing noise and improving prediction accuracy. The masking mechanism also allows the model to adapt to changes in network geometry—stations can be added, removed, or deactivated without retraining—thus overcoming the fixed-graph limitations of standard GCNs and enhancing the model’s robustness and scalability for earthquake early warning applications. A more detailed explanation of the masking strategy is provided in Section 3.2.

Our work articulates along the following main steps:

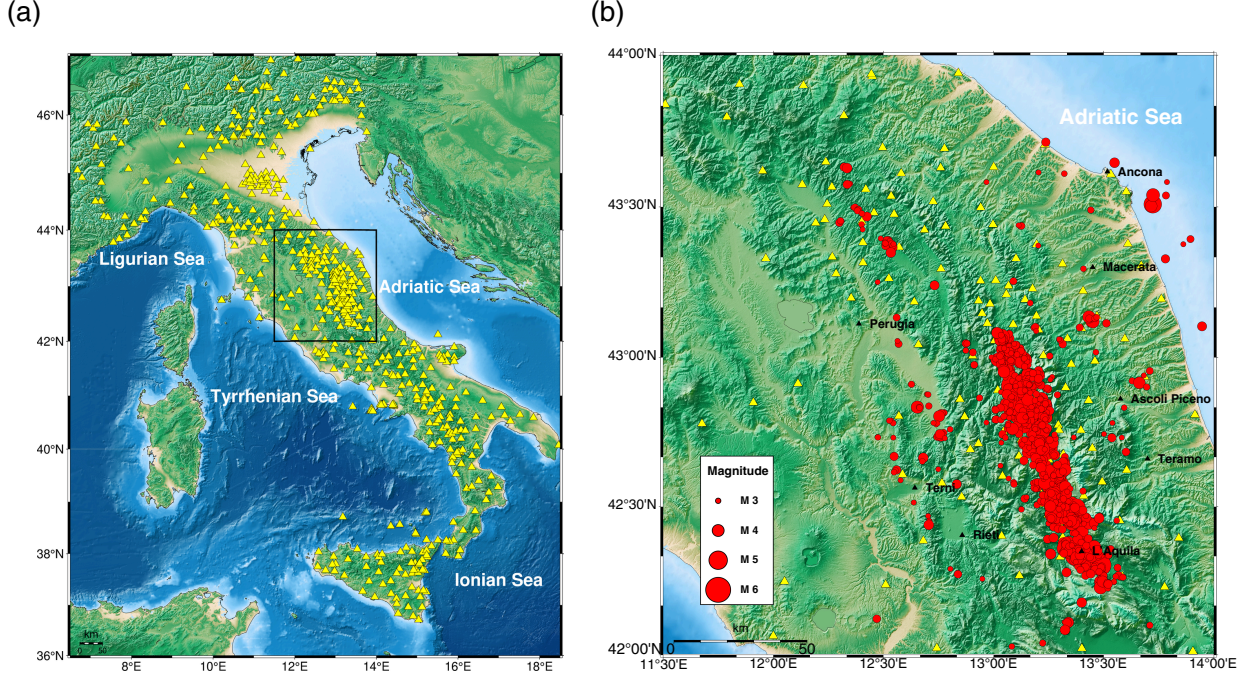
1. We build the dataset using a multi-step, pre-processing approach suitable for real-time applications to identify high-quality waveforms and improve the predictive performance of the model.
2. We present a novel architecture utilizing convolutional and graph convolutional layers to perform multivariate regression on time series originating from graph-structured data.
3. We evaluate our updated model thoroughly on the pre-processed dataset featuring 565 stations, evidencing the generality and potential of the proposed GCN-based architecture in this task. We discuss our results in detail and perform a Masked GCN to prioritize stations where the P wave arrives within the first 10 seconds.
4. We systematically evaluate the capabilities of our model through comprehensive experiments, incorporating additional information (e.g., station distances to a reference station) into the final fully connected layer of our proposed workflow.
5. Finally, we compare our results with predictions generated using the Ground Motion Model (GMM) by [16], calibrated for Italy, which requires an earthquake location and magnitude as input.

## 3.2 Data

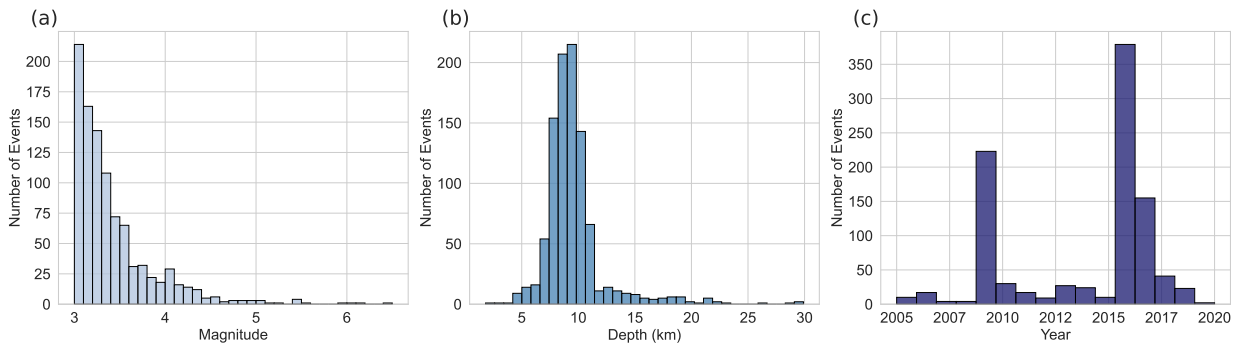
The input data consist of three-component earthquake waveform traces recorded in Italy primarily by the Italian National Seismic Network [101, 106]. The target data comprise intensity measures (IMs) associated with each recording: peak ground acceleration (PGA), peak ground velocity (PGV), and spectral acceleration (SA) at periods of 0.3, 1, and 3 seconds (SA(0.3), SA(1.0), and SA(3.0)). In our approach, the waveform inputs serve as the foundational data for the GCN model, designed to capture complex spatial and temporal patterns within seismic signals. Through this model, we aim to provide accurate predictions of ground motion intensities at stations (locations) further away from the epicentre of the earthquake.

The waveform data were downloaded from the INSTANCE dataset [104] and include both HN\* and HL\* (acceleration) and HH\* and EH\* (velocity) channels, where  $*$   $\in$  {E,N,Z}. All the waveform traces have a length of 120 s, are sampled at 100 Hz, and are provided in ground motion physical

units after deconvolution of the instrument transfer functions. We selected earthquakes in the same area of central Italy (bounded by latitude  $[42^\circ, 43.75^\circ]$  and longitude  $[12.3^\circ, 14^\circ]$ ) used by [17, 77, 78]. The earthquakes occurred between January 2005 and January 2020. Using these criteria, 975 earthquakes with magnitude  $M \geq 3$  and depth  $< 30$  km have been used (Figs 3.1, 3.2).



**Figure 3.1:** Spatial distribution of (a) the 565 stations (yellow triangles) together with (b) the 975 earthquakes (red dots) used in this study. We have selected earthquakes with magnitude  $M \geq 3$  and depth  $< 30$  km occurred in the study area bounded by latitude  $[42^\circ, 43.75^\circ]$  and longitude  $[12.3^\circ, 14^\circ]$  between January 2005 and January 2020.



**Figure 3.2:** Distribution of earthquake (a) magnitudes, (b) depths and (c) years for the selected events.

### 3.2.1 Quality Control and Filtering

To ensure the selection of high-quality waveform records, we applied a set of selection criteria to the metadata provided in INSTANCE, focusing on both accelerometric and velocimetric data characteristics:

- For acceleration data: we retained only traces with PGA values exceeding  $10^{-3}ms^{-2}$ . This threshold filters out records with extremely low accelerations, minimizing noise-dominated

traces.

- For Velocimetric Data Acquired by Broadband Sensors with 24-bit Data Loggers: we included only traces that featured maximum values within  $\pm 7,000,000$ , 17% below full scale, to reduce the risk of clipping and account for possible amplification or preprocessing issues. Additionally, traces with derived acceleration exceeding  $100 \text{ cm.s}^{-2}$  were excluded to filter out implausible or distorted records. These conservative thresholds enhance data reliability and improve model stability by limiting the influence of noisy inputs.
- For Co-located Accelerometric and Velocimetric Data, and for lower values of ground motions selected as explained in the previous steps, we included only the velocimetric data. Colocated instruments allowed us to do a further layer of quality control excluding large outliers, which we defined as data where the PGA ratio between the accelerometric and velocimetric data fell outside the range  $[0.5, 2]$ .
- For Data from non colocated Instruments: to determine which non colocated instrument data should be prioritized, we used the [16] GMM, comparing observed PGA values against predictions. We calculated the difference (i.e.  $\log_{10} \frac{PGA_{obs}}{PGA_{pred}}$ ) and selected the instrument data with the observed value closest to this prediction. Specifically, given that Bindi’s standard deviation is approximately  $0.35 \log_{10} PGA$ , we excluded data for which  $\left| \log_{10} \frac{PGA_{obs}}{PGA_{pred}} \right| > 0.7$ , as they were likely inconsistent or of lower quality. This filtering approach enhanced the dataset’s reliability and alignment with expected ground motion.

Following the above mentioned criteria, a subset of 565 stations has been selected (Fig 3.1). The dominant portion of the data ( $\sim 99\%$ ) has been acquired by the Italian National Seismic Network (IV code) and by the MedNet (MN code), both operated by INGV [37, 106]. The remaining records included in the dataset come from the Emersito Seismic Network (XO code), which was deployed in Amatrice and surrounding areas (Central Italy) to conduct field surveys shortly after the mainshock event on August 24th, 2016 (Mw 6.0).

### 3.2.2 Preparation for Model Training

The data preparation and processing for the training phase followed the criteria outlined in [77, 78]. Based on their findings, a 10-s window offers a good trade-off between prediction accuracy and timeliness, which we also adopted. Specifically, to align waveforms, all records were trimmed to a 10-s window starting from a reference time defined as 1 s before the P-wave arrival at the first recording station for each earthquake. Velocity traces were then differentiated to obtain acceleration.

Since recordings are not available across the entire national network for each earthquake, waveform data were entirely missing for many stations. These waveforms were filled with zeros following the approach of [77, 78]. This solution required overcoming technical challenges to adapt to a sparse graph with null values for numerous stations, marking a substantial advancement over the previous work by [17], which was limited to a fixed network of only 39 stations. When the waveforms (i.e. the model input) for a station were missing, the missing intensity measures (i.e. the model output) were estimated using the USGS ShakeMap software with the latest configuration for Italy [105] to ensure complete output data (target variables), following [77, 78]. Although using ShakeMap estimates introduces additional assumptions into the model, we consider it the most effective method

to handle missing data, allowing the model to generate approximations for a site even in the absence of input data—an important feature, for instance, in EEW applications. As a result, the target values are composed of 11% observed data and 89% ShakeMap-calculated data (a consequence of INSTANCE having 21 3-C waveforms per earthquake on average, most of them from stations closest to the epicenter). ShakeMap is a widely used software system that generates interpolated ground motion values by combining recorded data and/or macroseismic information, GMM-based estimates, and seismological parameters such as event location, magnitude, and finite fault models (when available). As such, it may introduce systematic biases—particularly in areas with sparse station coverage, near-fault regions, or complex site conditions. These biases can propagate into the model during training, potentially affecting its ability to generalize to purely observational data. Nevertheless, the results presented in Section 4 — and in particular in Section 4.2 — confirm that using ShakeMap-derived data as targets for the majority of the training set does not negatively affect the performance of the proposed model.

Following [77], the waveform data are normalized by the input maximum (i.e. the largest amplitude observed across all stations within the time window), and this maximum is saved as the normalization value which is later inserted into the final fully connected layer of our proposed Masked GCN model.

### 3.3 Method and Training

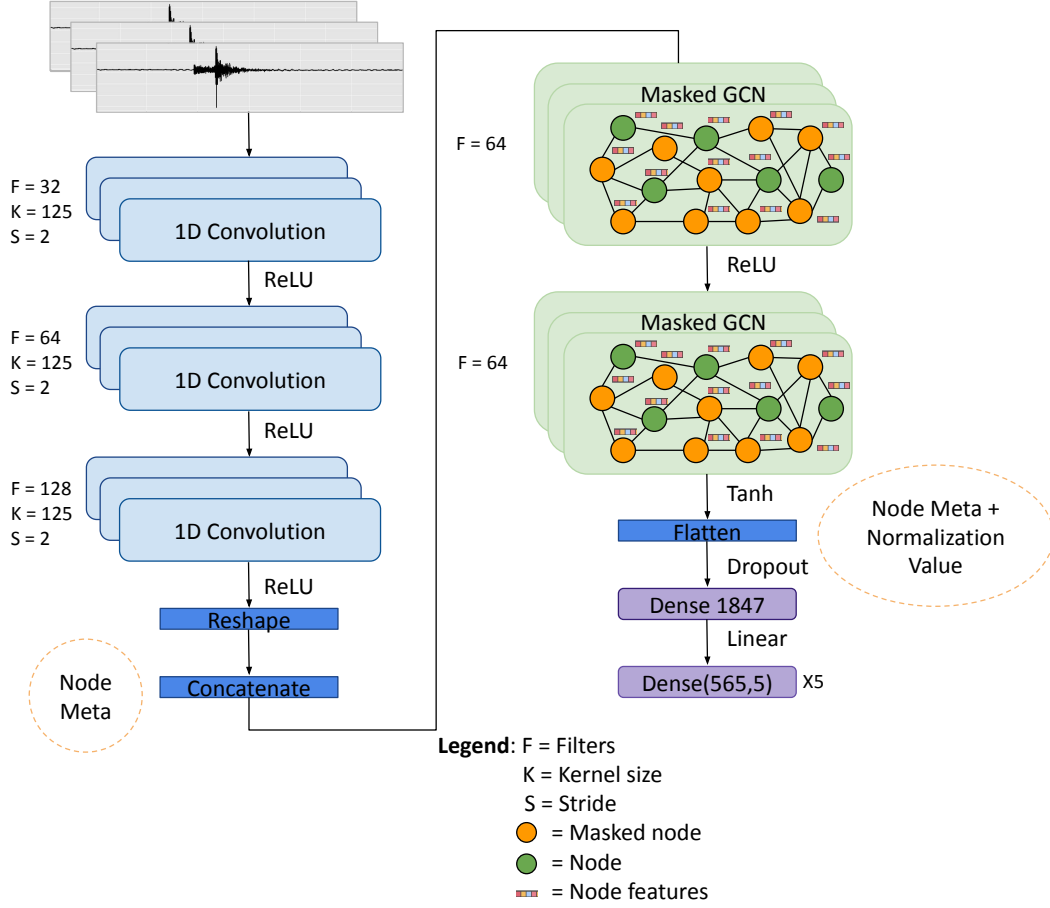
#### 3.3.1 Model Proposed

The Masked GCN model used here is a neural network architecture that combines CNNs and GCNs to capture both spatial and temporal information. Specifically designed for seismic signal analysis across a network of stations, the model predicts earthquake IMs (PGA, PGV, SA(0.3), SA(1.0) and SA(3.0)) at recording stations using multistation waveforms.

The architecture is based on [17], with several modifications, the most relevant being the introduction of a deeper network (i.e., an increased number of layers) and the implementation of the dynamic masking mechanism within the GCN component (Fig 3.3). Specifically, the model uses three primary components: convolutional layers, graph convolutional layers, and fully connected (FC) layers. The convolutional layers, designed to process 1D wave input data, consist of three layers with increasing output channels (32, 64, and 128). Each convolutional layer uses a kernel size of (1, 125) and a stride of (1, 2) to extract temporal features, followed by a ReLU activation function. The output from the convolutional layers is then reshaped for subsequent graph processing. The graph convolutional layers leverage spatial relationships across stations, with the first GCN layer projecting the input features into a 64-dimensional space, with the second layer further refining the learned representation.

In particular, spatial information is provided to the model through an adjacency matrix, which represents a graph structure based on interstation distances. The adjacency matrix was computed following the methodology described in [17]. All pairwise geodesic distances between stations are computed, then normalized and inverted so that closer station pairs receive higher edge weights. To control graph sparsity, a threshold of 0.3 is applied to remove weak (i.e., distant) connections. This means that all edges with a weight below this threshold are set to zero in the adjacency matrix, directly affecting the final graph topology. The resulting adjacency matrix is fixed for our network





**Figure 3.3:** The architecture of the proposed model for IM prediction. Boxes shaded grey represent inputs, shaded light blue represent convolutional layers, shaded blue represent operators, shaded green represent graph convolutional layers and shaded violet represent fully connected layers.

configuration, ensuring a consistent spatial structure while allowing event-specific features to vary at the node level, as explained in detail in Section 3.2. The output of the GCN layers is passed through a “tanh” activation function to introduce non-linearity and bound the values within  $[-1,1]$ . After the GCN layers, the network output is flattened and combined with additional information—including the maximum amplitude value—in a fully connected layer. We show that adding this knowledge in the neural network improves the results (see Section 4.1), consistent with the findings of [78]. In this work, we consider the maximum amplitude value from the vertical component waveforms recorded across all stations within the time window. In parallel, we verified that similar results were obtained when using the maximum amplitude across all three components (vertical, north–south, and east–west), which is the solution to be adopted in the operational phase. Finally, the last fully connected layer produces arrays of size  $(565,5)$ , where 565 is the number of stations, and 5 is the number of predicted IMs per station (PGA, PGV, SA(0.3), SA(1.0) and SA(3.0)). The base-10 logarithm has been applied to all the IMs (i.e.  $\log_{10} IM$ ).

The dataset comprises 975 seismic events, each recorded across up to 565 stations (but 21 stations on average), resulting in a total of 35,078 waveform instances. The data were randomly split into a training set (80%) and a test set (20%). Model training was conducted using a 5-fold cross-validation approach, where  $1/5$  of the training set was used for validation in rotation.

The model was trained for 100 epochs. ReduceLROnPlateau learning rate scheduler, starting at 0.0001, was employed to dynamically adjust the learning rate during training. The reduction is applied by a factor of 0.5 if the validation loss does not improve for three consecutive epochs. The mean squared error (MSE) loss function was used for model optimization, with the RMSProp optimizer. To mitigate overfitting, an additional dropout layer with a rate of 0.4 was applied before concatenating the maximum amplitude. Training was performed on an NVIDIA RTX 4090 GPU, utilizing approximately 9.2 GB of memory, and required a total of 1 hour and 12 minutes to complete across all 5 folds. In an operational setting, during testing, the model predicts each feature in 1 second, leading to a total of 5 seconds for all 5 features across 195 traces, excluding dataset loading time. Additionally, in a real-time deployment, the system would need to integrate a model such as PhaseNet [185] or STA/LTA method for detecting the first P-wave arrival. Then, our pipeline incurs a cost of 1.06 seconds for data preprocessing and 0.7 seconds for the prediction for a single event, resulting in a total computation time of approximately 1.76 seconds per event.

### 3.3.2 Mask Method:

The Masked GCN model utilizes a dynamic masking mechanism to determine which stations contribute to the model’s predictions, enabling flexible input data configurations and enhancing real-time adaptability for seismic monitoring. The mask is implemented as a tensor, where each value indicates whether the input data from a given station should be included (1) or excluded (0) [177]. This mask is applied to the waveform data prior to their input into the model and during loss computation. This mask is designed to include only waveforms where the P-wave arrives within the first 10 seconds starting from the P arrival at the first recording station. During the training phase, we used the manually picked P-wave arrival times from the INSTANCE metadata for each station. This approach minimizes the inclusion of noisy or irrelevant signals, allowing the model to focus on meaningful seismic data. Conversely, in the testing phase, we computed the travel times assuming a P-wave velocity of  $6.5 \text{ kms}^{-1}$  based on the P arrival at the closest station provided in the INSTANCE metadata for each earthquake. A more detailed explanation of the mask’s functionality is provided in Figure S1. By restricting the input to this 10-second window, the mask increases the likelihood of capturing the P-wave arrival, and often both the P- and S-waves of the stations close to the hypocenter, at each station within this timeframe. This targeted selection strategy not only enhances the quality of input data by prioritizing critical early-wave information but also optimizes the model’s responsiveness in real-time scenarios. In practical terms, this procedure would be based on the station with the first trigger which should be a good approximation of the station closest to the epicenter (and the epicenter itself in the regions with moderate to high network density), especially for earthquakes with  $M \geq 3$  (used in this study).

The masking process is dynamic and operator adjustable, allowing the system to adapt to network changes, such as the addition or deactivation of stations. If a station goes offline or becomes unreliable, its mask value can be set to [0], effectively excluding it from the model’s predictions without requiring structural modifications or retraining. This flexible masking strategy overcomes a common limitation of graph convolutional networks, which typically struggle to handle changes in node structures. By selectively masking inactive nodes, the model maintains a consistent graph structure, ensuring seamless adaptation to variations in the operational network while preserving the reliability and relevance of its predictions.

## 3.4 Results

### 3.4.1 Evaluating the Impact of Additional Knowledge

In the first part of this study, we explored the additional knowledge needed to make reliable predictions of the maximum values of the IMs. To assess the effectiveness of our model, we compared the training performed using the (i) coordinates of the first station that records the P wave, (ii) maximum amplitude information and (iii) interstation distances (i.e., the distance matrix) as additional metadata inserted into the last fully connected layer. These three options were added in varying combinations to create eight different configurations, allowing us to evaluate the individual and combined impact of each metadata type on model performance. More specifically, in the first configuration, neither the coordinates of the first P-wave station nor the interstation distances were incorporated, while the normalized amplitude value was included. In the second configuration, all three additional features were included. In the third configuration, all metadata except the normalized amplitude value were integrated. The same approach was applied to the remaining configurations, ensuring a full exploration of all eight possible combinations. A complete representation of the eight configurations and the results of all the experiments is shown in Table S1.

For each experiment, the Masked GCN model results from all five-fold cross-validation test sets are averaged. The best results were achieved when all information, except the coordinates of the first P-wave station, was included (i.e., Experiment 6 for SA(0.3) and SA(1.0)) or when only the maximum amplitude information was integrated in the model (i.e., Experiment 1 for PGA, PGV and SA(3.0)). The mean squared errors (MSE) were: 0.188 for PGA, 0.199 for PGV, 0.181 for SA(0.3), 0.229 for SA(1.0) and 0.267 for SA(3.0). Remarkably, when the third feature is also added, model performance deteriorates (Table S1) compared to that observed in Experiments 1 and 6. This result might originate from the redundancy of information inserted in the model. That is, the knowledge conveyed by the coordinates of the first station that records the P wave is likely captured already by the adjacency matrix and the interstation distances. Finally, we evaluated the performance of the model’s best configuration with and without applying the aforementioned mask, which focuses on stations where the P-wave arrives within the first 10 seconds. The results indicate that the MSE of the Masked GCN model for the 5 IMs is lower than that of the model without the mask (i.e., 0.203 for PGA, 0.210 for PGV, 0.194 for SA(0.3), 0.243 for SA(1.0) and 0.278 for SA(3.0)). This outcome is expected, as the selection of good quality data recorded near the epicenter improves the accuracy of our predictions. It is worthwhile to stress here that the methodology does not avail of any explicit hypocenter location and magnitude input information. Overall, Experiment 1 was found to be the most effective configuration for all five IMs and was adopted for our model. Furthermore, we conducted an additional experiment using PhaseNet as the pretrained encoder, selected for its greater capacity and training on the STEAD dataset via SeisBench. Initializing the encoder with these weights improved performance by reducing MSE by 2.7% for PGA, 5.5% for PGV, and 4.1% for SA(0.3) compared with Experiment 1, and lowered GPU memory usage from 9.2 GB to 2.58 GB. We will further investigate this approach and evaluate alternative pretrained temporal encoders and techniques in future work.

### 3.4.2 Model performance

Considering the results of the Masked GCN model on each individual earthquake and each IM metric, Figure 3.5 displays scatter density plots comparing predicted versus target values of the test set. The analysis also includes ShakeMap estimates. In Figure 3.5 the x-axis represents the target values in on a base-10 logarithmic scale, while the y-axis shows the predicted values, also on a base-10 logarithmic scale and in the same units. A red dashed line indicates the ideal prediction line, where the predicted values match the target values. The scatter points are coded by color and size to represent data density, with smaller dots indicating low-density areas and larger dots highlighting high-density areas. The colorbar, ranging from dark to light colors, represents the number of data points. The points cluster around the ideal prediction line, indicating a general agreement between predicted and target values. However, the presence of a few outlier events in the test set can be attributed to the unbalanced dataset, which contains limited training examples for larger magnitudes, making it challenging for the model to generalize effectively at higher magnitudes. Furthermore, rather poor predictive performance is also observed in a few borderline earthquake cases located in areas with a sparse distribution of stations (e.g., offshore or near the coast). As a result, these events are characterized by only a limited number of stations providing earthquake P-wave signals (or even S-wave signals for the nearest ones) within the 10-second window.

### 3.4.3 Examples of event data prediction

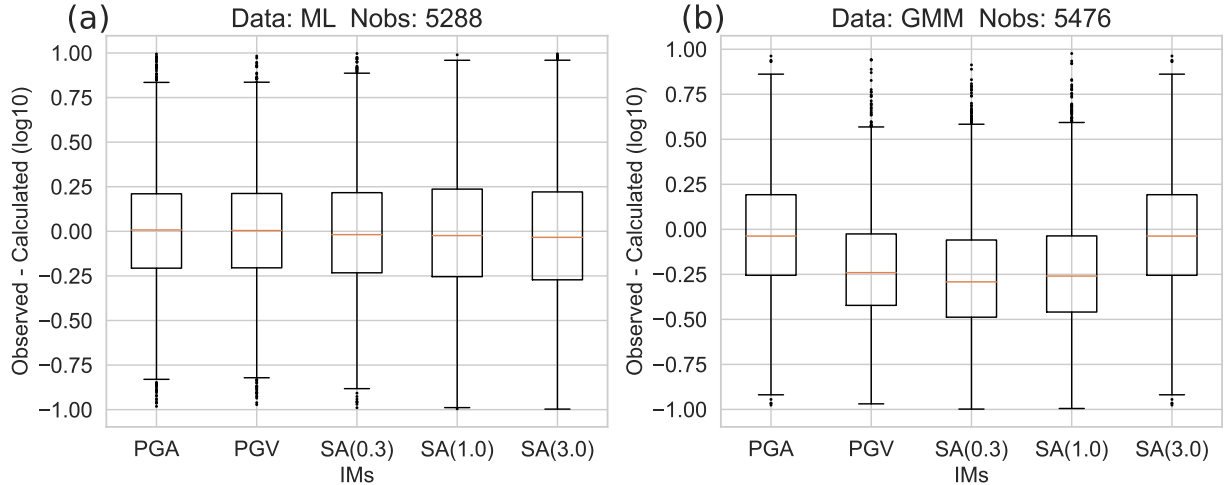
The panels in Figures 3.6, 3.7 and 3.8 display six sample events from the test set, presenting a comparison of real and predicted PGA values both spatially and as a function of distance. The maps and scatter plots in Figures 3.6, 3.7 and 3.8 clearly illustrate the performance of the predictive model, while the summary table provides a quantitative evaluation of the model’s accuracy. Specifically, the top row illustrates the spatial distribution of observed PGA (left), predicted PGA (center), and their differences (right). The maps highlight the station locations, with color gradients representing the magnitude of the values: from purple (low) to yellow (high) for observed and predicted values, and from blue (underpredictions) to red (overpredictions) for the differences. The epicenter is marked by a black star. At the bottom, the left scatter plot shows observed (blue) and predicted (red) PGA values as a function of epicentral distance. The center scatter plot illustrates the residuals (observed - predicted) against epicentral distance, with dashed lines indicating the median, the first and the third quartile residuals, respectively. The table (bottom right) summarizes key metrics, including the range of observed and predicted PGA values, the mean absolute error (MAE) and the mean squared error (MSE).

The six events shown include the 26 August 2016 M 4.8, the 15 September 2016 M 3.7, the 26 October 2016 M 3.1, the 14 December 2016 M 3.8, the 6 April 2009 M 6.1 and the 13 June 2013 M 3.8. They have been selected because they are representative of the overall results. The relatively low MAE and MSE values for the first four earthquakes (Figs 3.6 and 3.7) demonstrate a strong agreement between observed and predicted values, representing the accuracy achieved in 96.4% of the cases within the test set. In contrast, the last two earthquakes are representative of the remaining 3.6% (the outliers) of the testing events, where predictive performance is significantly reduced. This lower accuracy can be primarily attributed to: (i) the limited availability of earthquake waveform data for larger magnitudes (Fig 3.8a) and (ii) poor station coverage—especially the

uneven distribution in different directions surrounding the epicenter—which leads to incomplete or unbalanced waveform data and, consequently, to inaccurate ground motion predictions, potentially confusing the model during inference (Fig 3.8b).

### 3.4.4 Benchmarking the Model Against Baselines

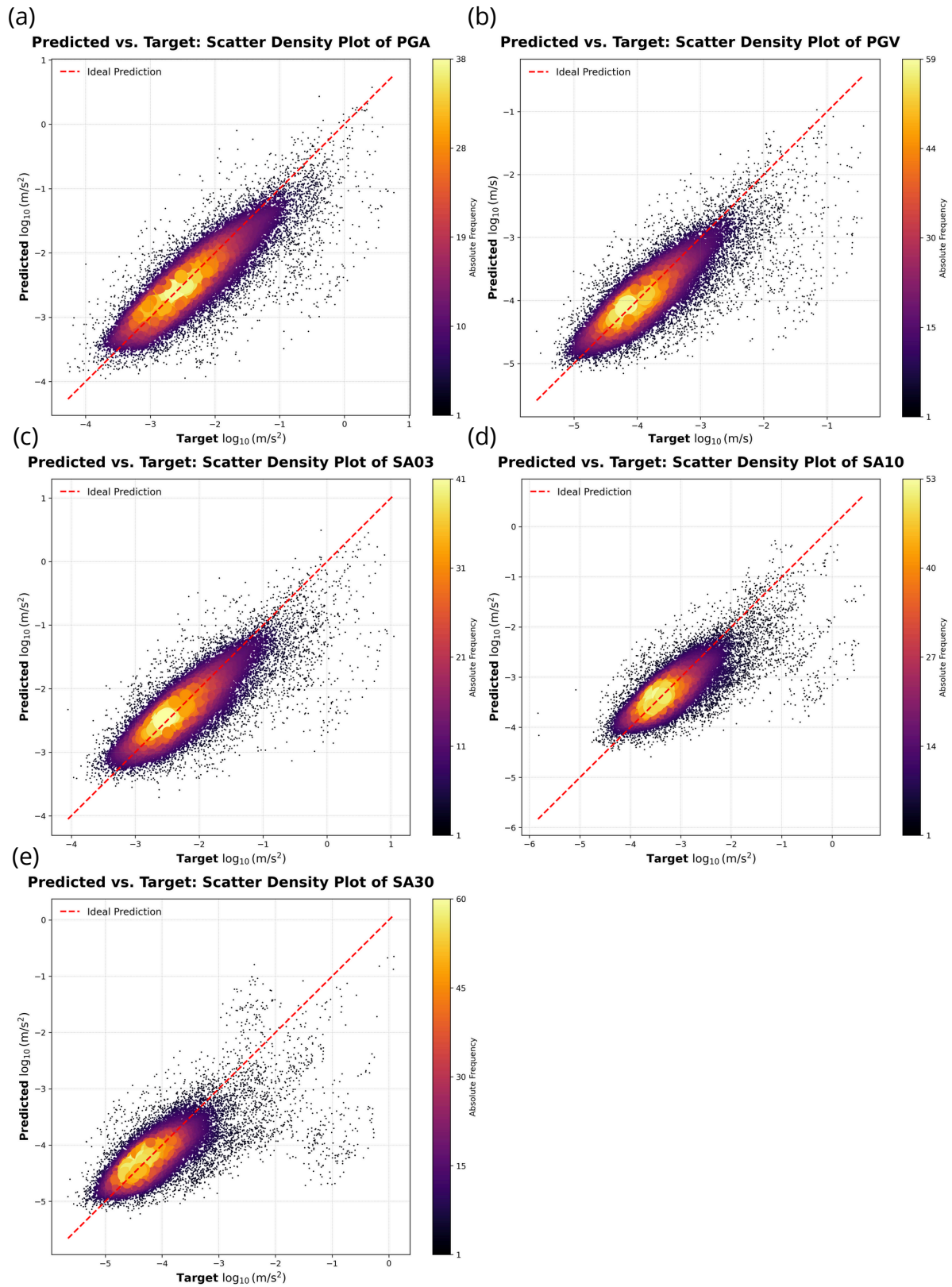
To evaluate the performance of the Masked GCN model against a baseline case, we compared our test set results with predictions obtained using the GMM by [16], calibrated specifically for Italy. For this comparison, we excluded the ShakeMap predictions used when no observed data are available (as it relies on the [16] GMM itself). For the GMM predictions, we used the INSTANCE metadata [104] and we applied the necessary corrections based on the EC8 site classes of the stations. No between-event correction [9], however, has been applied when predicting the ground motion using the GMM by [16]. The residuals between the base-10 logarithms of observed and predicted values (i.e.  $\log_{10} \frac{IM_{obs}}{IM_{pred}}$ ) have been calculated together with their mean, median and standard deviation (SD) for the results obtained with (i) the Masked GCN model and (ii) the GMM by [16] (Table S2 and Fig 3.4). Large residual values  $|\log_{10} \frac{IM_{obs}}{IM_{pred}}| > 1$  were removed resulting in 94.21% of the data retained for the GCN predictions and 97.56% for the GMM estimates. Our analysis shows that the median values of  $\log_{10} \frac{IM_{obs}}{IM_{pred}}$ , which reflect model bias, are significantly reduced in the Masked GCN model, particularly for PGV, SA(0.3) and SA(1.0). In terms of standard deviation, the values are similar across both models and are consistent with those reported by [16], who observed a standard deviation between 0.34 and 0.38 for  $\log_{10} \frac{IM_{obs}}{IM_{pred}}$ .



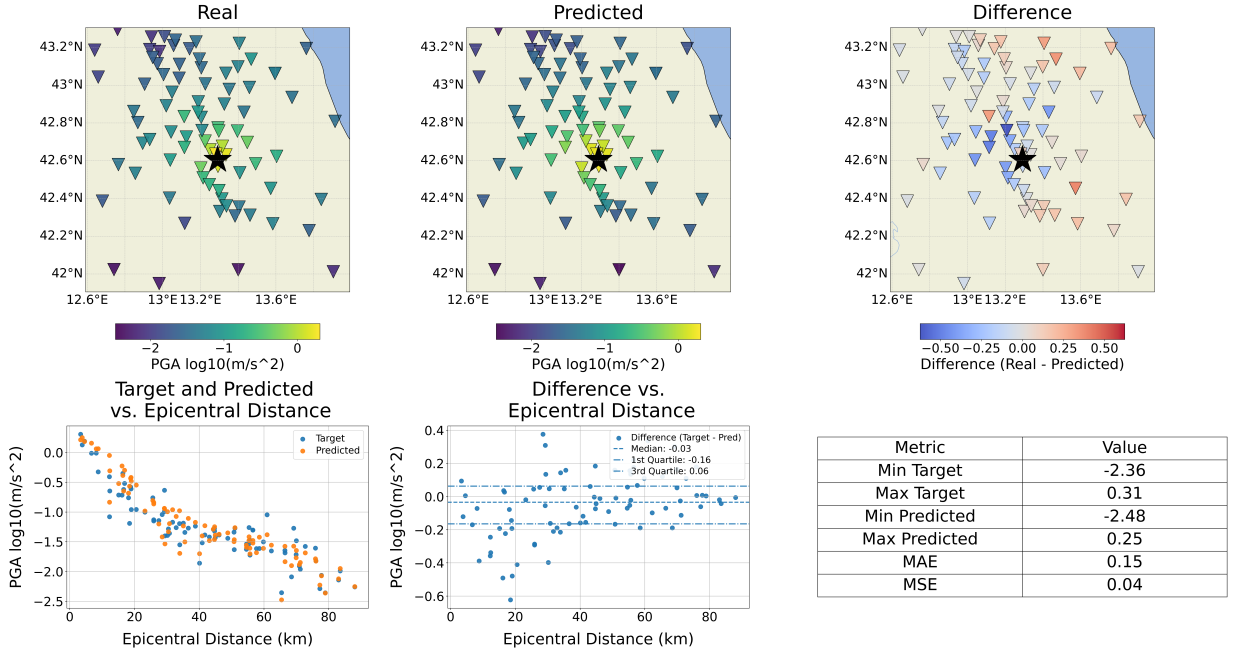
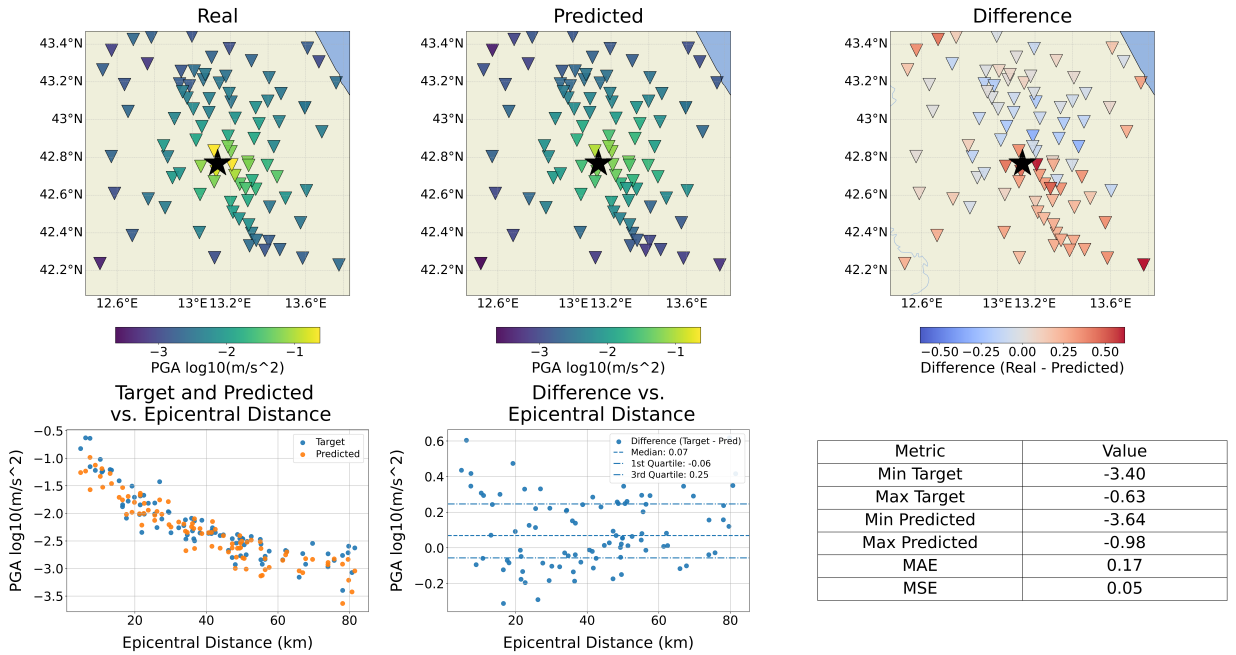
**Figure 3.4:** Boxplots for the residuals  $[\log_{10} \frac{IM_{obs}}{IM_{pred}}]$  of the test set (195 events): (a) masked GCN model results for the observed IMs and (b) GMM results.

We also compared our work with the models proposed by [77] and [17] (Table S3). These baseline models were not given the exact same input data as our proposed Masked TISER-GCN; instead, they were provided with different time series per station for different earthquakes in the same area. Overall, when evaluating the individual performance of the models, our Masked TISER-GCN performs comparably to the baselines. In terms of MSE, our model shows an improvement of 6% on PGA, 5.5% on PGV, and 5% on SA(0.3) compared to the TISER-GCN model proposed by [17]. When compared to the CNN model from [77], the improvement is even more pronounced,

achieving 17% on PGA, 31% on PGV, and 33% on SA(0.3).



**Figure 3.5:** Observed versus predicted scatterplot of the testing dataset for: (a) PGA, (b) PGV, (c) SA(0.3), (d) SA(1.0) and (e) SA(3.0). Scatter points are both color and size-coded, with black indicating low-density areas and yellow representing high-density areas. The colorbar shows the total number of data points in each area.

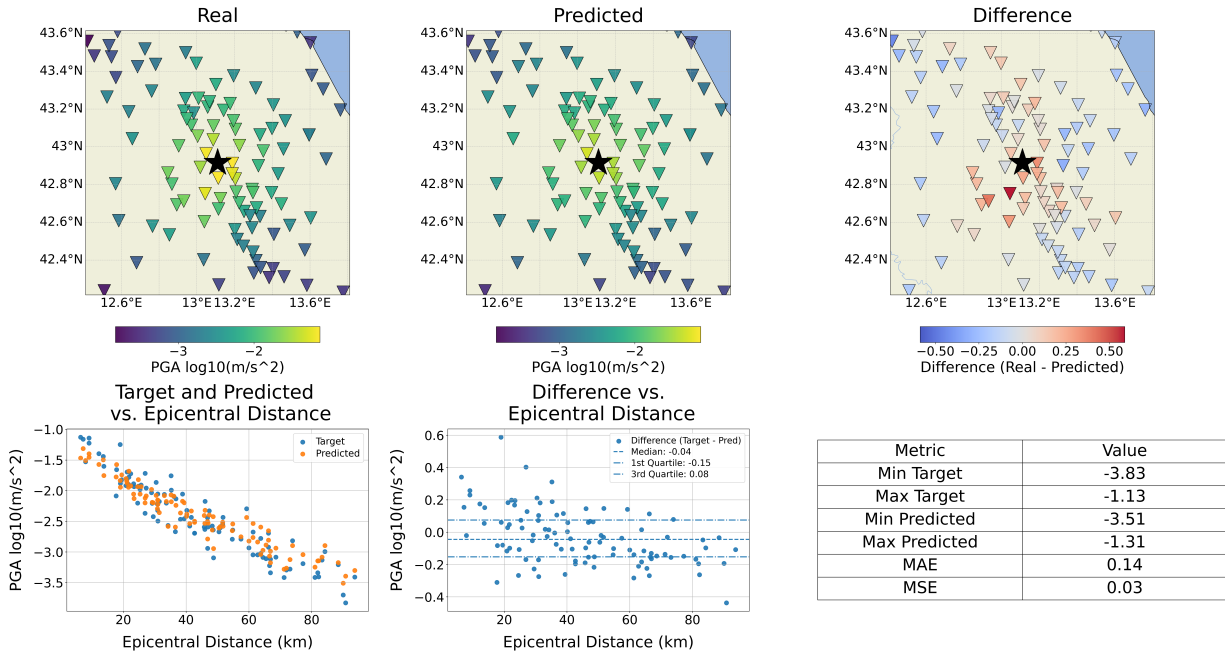
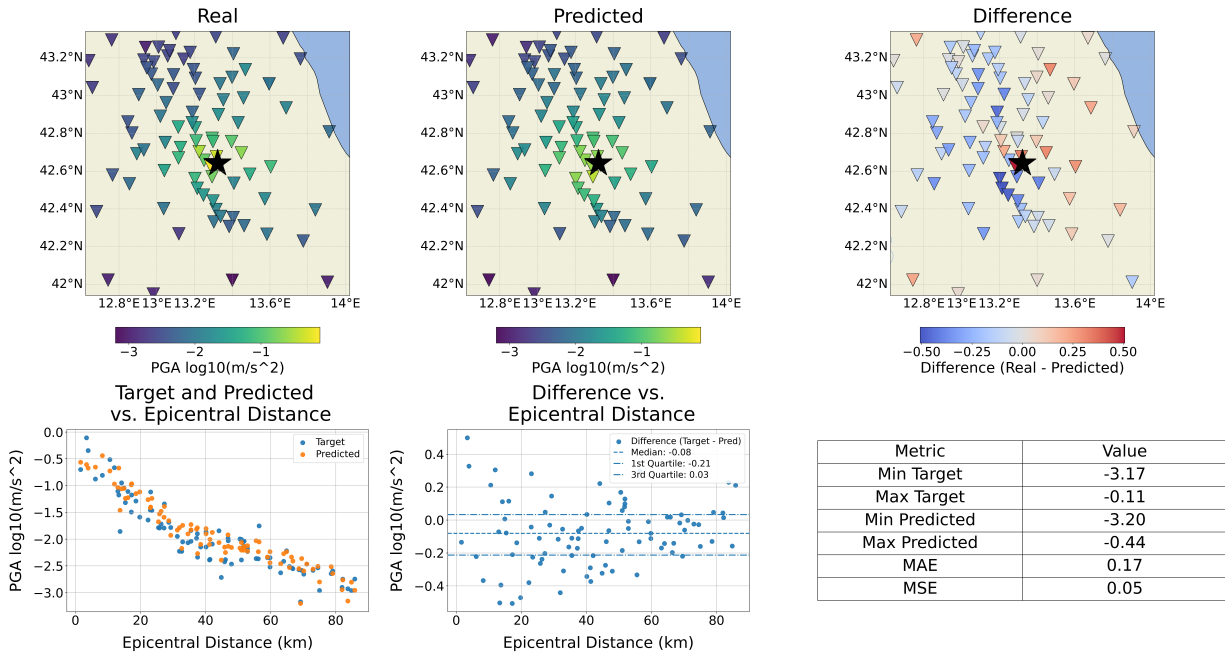
**a) Comparison of Real and Predicted — Event: 7224451 — Magnitude 4.8****b) Comparison of Real and Predicted — Event: 8034701 — Magnitude 3.7**

**Figure 3.6:** Our model results for (a) the 26 August 2016 M 4.8 and (b) the 15 September 2016 M 3.7. The top row presents spatial distributions of observed PGA (left), predicted PGA (center), and their differences (right) across the stations, with color gradients representing the respective values. The bottom row includes scatter plots of PGA values as a function of epicentral distance: observed vs. predicted values (left) and their differences (right). Key performance metrics, such as MAE, MSE, and the range of observed and predicted PGA values, are summarized in the accompanying table (bottom right).

### 3.4.5 Effect of Window length on Model Performance

Sections 4.1, 4.2, 4.3 and 4.4 present results based on a 10-second input window length. In the final part of this study, we investigated how our model performs when the window length is reduced, as

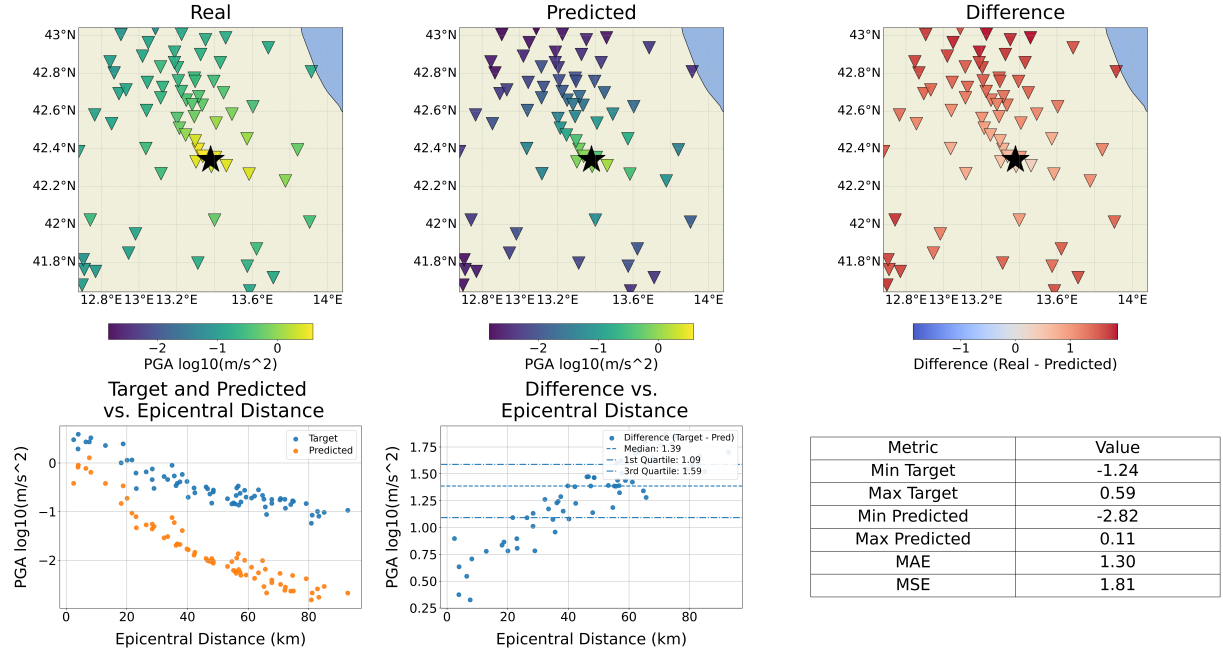
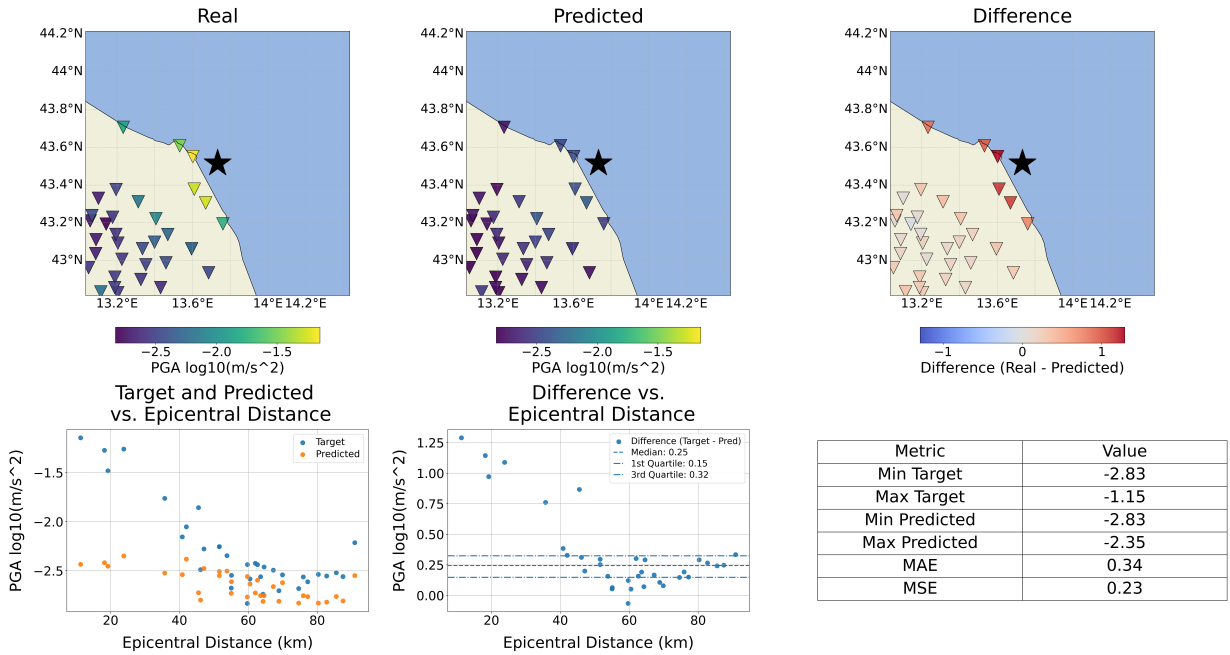


**a) Comparison of Real and Predicted — Event: 8673931 — Magnitude 3.1****b) Comparison of Real and Predicted — Event: 11510961 — Magnitude 3.8**

**Figure 3.7:** Our model results for (a) the 26 October 2016 M 3.1 and (b) the 14 December 2016 M 3.8. Refer to the caption of Figure 3.6 for further details.

shorter windows could lead to earlier predictions, a critical aspect for integrating this methodology into EEW systems. However, using a smaller input window introduces greater challenges, as fewer stations will have received sufficient information to predict IMs at more distant locations. To mitigate this, we allowed data from as many stations as possible to be considered. Specifically, we included all stations that recorded the earthquake within 5 seconds, regardless of their distance from the first recording station. Our findings indicate that the input window can be reduced by half while still keeping comparable performance (Fig 3.9). Specifically, For PGA and PGV, the MSE of the



**a) Comparison of Real and Predicted — Event: 1895389 — Magnitude 6.1****b) Comparison of Real and Predicted — Event: 2131571 — Magnitude 3.8**

**Figure 3.8:** Our model results for (a) the 6 April 2009 M 6.1 and (b) the 13 June 2013 M 3.8. Refer to the caption of Figure 3.6 for further details.

residuals between the base-10 logarithms of observed and predicted values  $[\log_{10} \frac{IM_{obs}}{IM_{pred}}]$  is 0.205. For SA(0.3), the MSE increased to 0.198, while for SA(1.0) and SA(3.0), the misfit increased slightly to 0.236 and 0.269, respectively. Overall, these results emphasize the effectiveness of the GCN layers and our implementation. Given that GCN layers are designed to facilitate node feature sharing during the convolution process, they enable the transfer of a substantial amount of information between nodes, even when the input is reduced by half, supporting the feasibility of our approach for real-time applications such as EEW.

### 3.5 Discussion

In this study, we demonstrated that our Masked GCN model can accurately predict earthquake IMs at recording stations using only waveforms captured within a 10-second time window starting from the earthquake’s first P-wave arrival times. This approach enables accurate prediction of IMs at distant stations, even before they record the earthquake signal or its maximum values, based solely on the initial recordings at stations near the epicenter. To achieve this, we used 3C station waveform data and the spatial pattern of the waveforms across the recording network, without requiring any prior knowledge of the earthquake’s location or magnitude. Our approach is conceptually similar to GMMs, but instead of relying on earthquake source (location and magnitude) and site parameters, it uses direct ground motion observations from the station network, specifically the waveform ground motion patterns.

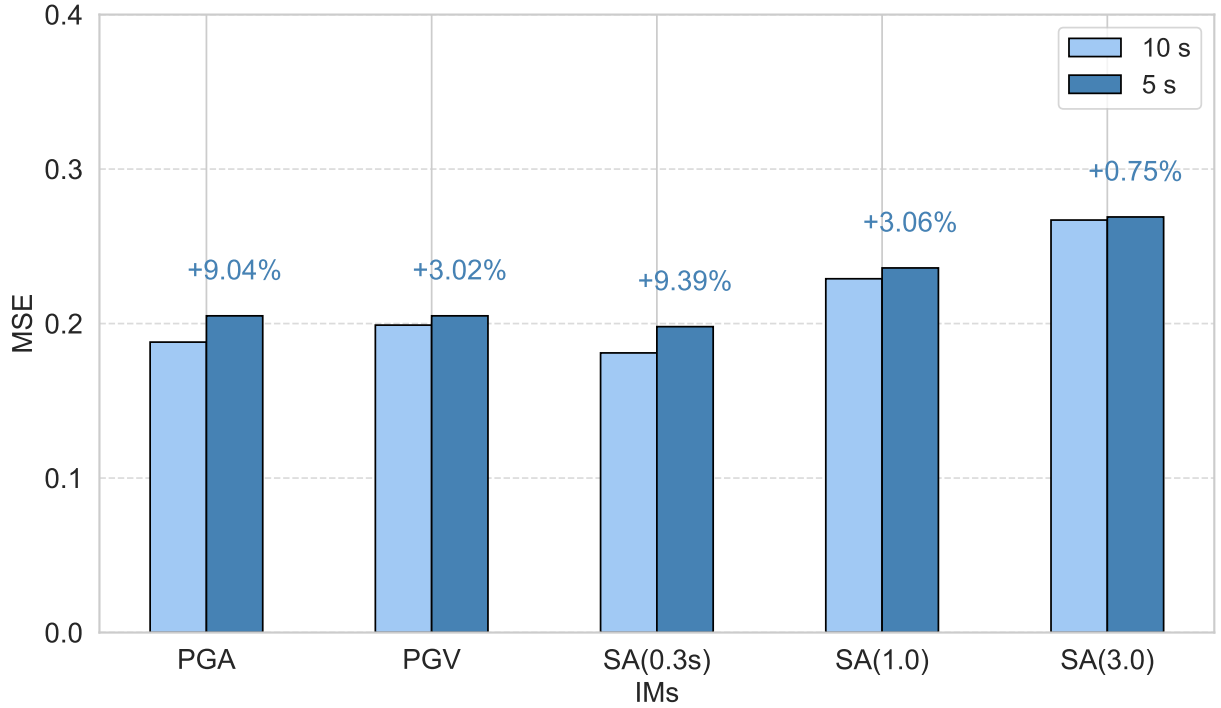
The target outputs—PGA, PGV and SA at 0.3, 1, and 3-second periods—are predicted as a regression problem for all 565 stations using the Masked GCN model. This marks a significant advancement over the original TISER-GCN [17], which was constrained by a fixed network geometry with only 39 stations. The model generally performs consistently well across all IM types. As expected, its performance improves when additional knowledge is integrated into the model and when stations receiving the P-wave within the first 10 seconds are prioritized.

Due to the Gutenberg-Richter magnitude distribution, our dataset is significantly imbalanced, with a much higher frequency of smaller-magnitude events compared to larger ones. This imbalance presents challenges in predicting the IMs for larger events, as illustrated in Figure 3.8a. One potential solution involves applying data augmentation techniques, such as those proposed by [154], by introducing additional training data for larger magnitudes through random transformations of existing events. Alternatively, synthetic seismograms could be generated for hypothetical larger events at the same stations. Another approach to address limited data is transfer learning [152], where a pre-trained model from a larger dataset is fine-tuned on a smaller, domain specific dataset. As shown by [78], a pre-trained network designed for other seismological applications can be effectively adapted for tasks such as IM prediction or magnitude estimation. Similarly, the network developed in this study could be utilized as a pre-trained model for IM prediction in other regions.

To validate the methodology, we used processed waveforms with preliminary data cleaning to exclude stations affected by missing or erroneous data due to equipment failures or transmission issues. Following [77], missing data were replaced with zeros, and ShakeMap predicted target values were adopted to ensure the availability of target data during training. Despite these limitations, the Masked GCN model demonstrated robust accuracy in predicting IMs, suggesting its ability to learn meaningful spatial relationships between stations, conditioned on the input data. This implies that the model can exploit the spatial configuration of the seismic network and adapt its predictions based on the available waveform signals, even if it does not explicitly model the physical process of seismic wave propagation.

To evaluate the predictive performance of the Masked GCN model, we compared it against GMMs proposed by [16], which are employed in ShakeMap configurations for Italy [105]. As shown in Figure 3.4, the Masked GCN model outperforms the GMMs in terms of IM residuals. This is likely due to (i) the independence of the Masked GCN model from earthquake magnitude uncertainties (eliminating the need for between event correction terms), (ii) the relatively uniform source-receiver

geometry and wave paths within the study area, and (iii) the model’s ability to capture and adjust for local site effects.



**Figure 3.9:** MSE at 10s and 5s input window lengths for each IM. Our model achieves approximately the same MSE values even with half the input, demonstrating the effectiveness of GCNs in processing spatial information.

Notably, the earthquakes in our dataset are spatially concentrated, reducing variability at individual stations caused by differing wave paths. This likely contributed to the strong performance of our model. Nevertheless, we are confident that by incorporating a larger dataset with a broader spatial distribution of epicenters, the Masked GCN model can learn site and path specific characteristics for each station and zone, provided there is sufficient station coverage in these areas. Future studies will address these aspects by expanding the geographic scope and dataset size. In this context, a more accurate selection of P-wave velocity could be necessary to refine the travel-time estimates, considering a larger area that may require the adoption of different velocity models, depending on the depth and specific geology.

Our results (Figs 3.5, 3.6 and 3.7) confirm that the proposed methodology can predict ground motion effectively. Combined with the translational invariance properties of neural networks, this suggests that the approach could be implemented in real-time systems. In such a setup, continuously streamed data could be used to predict IMs while leveraging the dynamic masking mechanism to flexibly select which stations contribute to the model’s predictions, allowing for adaptable input configurations.

### 3.6 Conclusions

A Masked GCN model was employed to predict IMs, including PGA, PGV, and SA at 0.3 s, 1 s, and 3 s, using multistation 3C waveforms within a 10-second time window starting at the first

P-wave arrival. The dataset consists of waveforms recorded by 565 stations from 975 earthquakes ( $M \geq 3.0$ , depth  $< 30$  km) in central Italy.

The IM predictions were achieved without prior knowledge of earthquake location, distance, and magnitude. The model performance improved when additional information—such as the coordinates of the first-arrival station and the maximum amplitude value—was incorporated. To optimize the performance further, a dynamic masking mechanism was introduced to manage which stations contributed to the predictions, enabling flexible input configurations.

Our findings show consistent model performance across all IM types. A comparison between the Masked GCN model and the [16] GMM, calibrated for earthquakes in Italy, revealed that the Masked GCN exhibited no prediction bias while maintaining a similar residual variance.

Model performance degraded only in two scenarios: (i) for large-magnitude events due to the limited number of such earthquakes available for training, and (ii) in regions with sparse station coverage (e.g., offshore or near the coast). Our model achieves comparable performance while using only half of the input window length (5 s).

Overall, the results suggest that the proposed approach could be integrated into EEW systems, leveraging the Masked GCN model’s ability to prioritize critical early-wave information through dynamic masking. Although this work focused on a geographically concentrated dataset, the methodology could be extended to larger areas with more widespread seismicity, provided sufficient training data is available. Overall, we showed that the proposed Masked GCN could achieve ground motion predictions as reliably as those in [17] without being restricted to a fixed station geometry.

## Chapter 4

# LLM4Geo

In this Chapter, we introduce a novel approach for rapid earthquake source characterization using Large Language Models (LLMs). Determining earthquake magnitude and location with fast and automated methods is a cornerstone of *Earthquake Early Warning Systems (EEWS)*, where every second is crucial to issue timely alerts and reduce potential damage.

We explore the potential of *TinyLlama*, a lightweight yet powerful LLM, and adapt it to seismology through fine-tuning on a small subset of earthquake data using the *LoRA* (Low-Rank Adaptation) technique. This strategy allows retraining only 0.1% of the model parameters, making the approach computationally efficient while retaining strong predictive capabilities.

Our method takes as input station coordinates, P-wave arrival times, and peak ground velocities within a narrow window of 0.2 s around the P-wave onset. From these, it directly estimates earthquake magnitude, epicenter, hypocenter, and origin time. To ensure near real-time applicability, we restrict the analysis to stations where the P-wave arrives within 5 seconds of the first triggered station, enabling rapid and scalable computation.

Compared to standard automatic procedures for rapid parameter estimation, the proposed model achieves similar accuracy in magnitude prediction and superior performance in epicentral, hypocentral, and origin time estimation, with an average improvement exceeding 26%. Moreover, we validate the adaptability of the model by extending the analysis to both P- and S-waves and by modifying the input station and source configurations, observing stable and robust performance across diverse conditions.

These results demonstrate that LLMs can be effectively repurposed for seismological tasks, paving the way for a new generation of data-driven approaches to earthquake early warning and real-time seismic analysis.

### 4.1 Introduction

Determining earthquake source characteristics, such as magnitude and location, is a critical first step in nearly every aspect of earthquake science. Earthquake catalogs are crucial for improving hazard assessment, enhancing our understanding of Earth’s structure and improving earthquake early warning systems (EEWS). Interest in EEWS has grown dramatically in recent years, driven by advancements in communication systems and improvements in automated methods for earthquake signal detection/identification [7]. Modern EEWS provide real-time information such as location

and hazard level to issue warnings before significant shaking. While there are a variety of EEWS methods, three main approaches exist: Front detection [5, 22, 46], in which seismic sensors are placed between an area of interest and a seismic zone, and a warning is triggered whenever ground shaking greater than a certain threshold is measured; On-site approaches [20, 64, 120], using single station data to provide warning in a small geographical area; and Regional approaches [6, 23, 79, 141, 173], using data from a network of stations, potentially distributed over a large area.

In recent years, earthquake magnitude estimation, as well as epicentral and hypocentral estimation, have been significantly enhanced through methods that combine advanced signal processing with high-density sensor networks [137]. Traditionally, seismic wave analysis relies on amplitude and duration, but this tends to saturate for large-magnitude events. Therefore, spectral analyses and empirical calibration of magnitude scales have been developed [2, 31, 53].

Standard methods for estimating location typically focus on determining the hypocenter that minimizes the difference between observed and predicted wave arrival times. These methods can be subdivided in absolute and relative location methods. The iterative techniques, such as the Geiger method introduced in 1912 [54], as well as grid search algorithms [130], graphical methods [107], or more advanced techniques like NonLinLoc [98], are absolute location methods. Relative location methods allow for much higher relative precision but they may suffer of bias in absolute location owing to inaccuracies of the velocity model. Early works have been improved steadily [38, 50, 56, 70, 127, 139] to arrive at the current, widely used method of double difference (DD) [166].

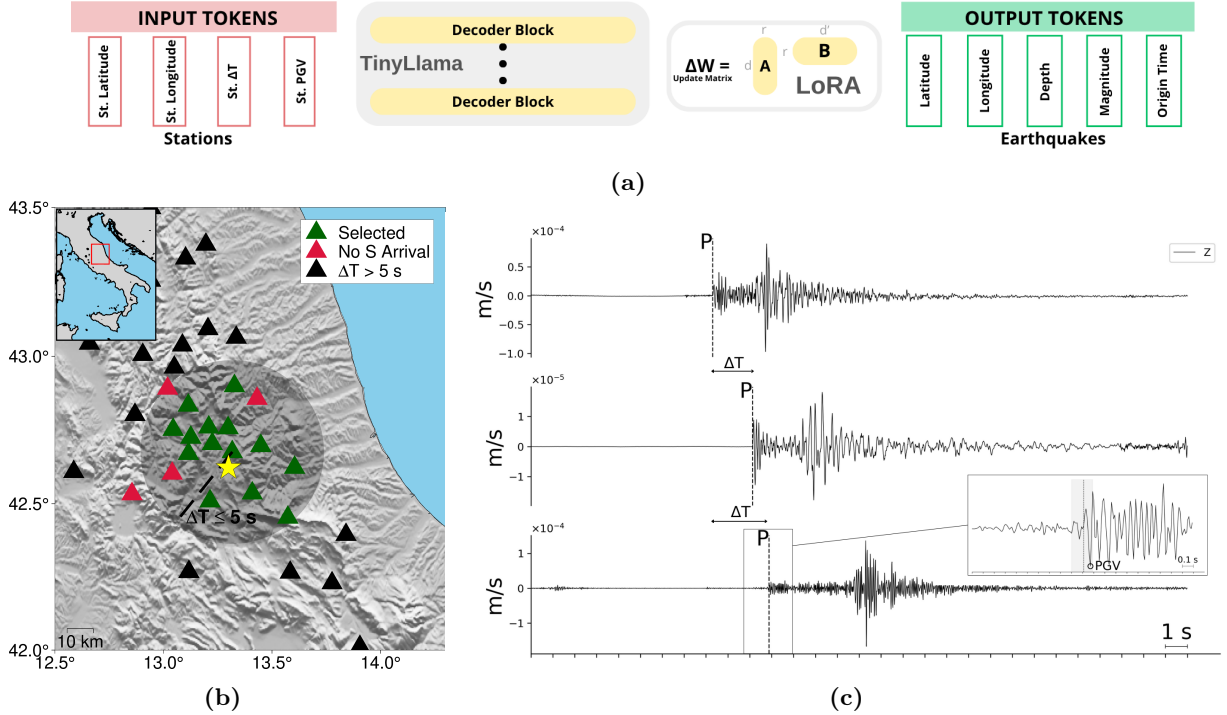
Machine learning (ML) has revolutionized these tasks, with neural networks such as CNNs and RNNs capable of processing raw seismic data to provide highly accurate predictions. For example, DCNN-M [184], a deep convolutional neural network, has been proposed to estimate magnitude using 12 parameters derived from the initial 3 s of the P-wave. These parameters, related to frequency, amplitude, and energy, provide interpretable inputs, and the model outperforms traditional methods based on characteristic period  $\tau_c$  and peak displacement amplitude  $P_d$  [72, 80, 86]. Transfer learning has allowed models trained in one region to be applied elsewhere, facilitating broader applicability [42] [78]. Deep learning further enhances these capabilities: Mousavi and Beroza [111] demonstrated that CNNs can estimate earthquake magnitude directly from single station waveforms, and attention-based networks, which incorporate additional station metadata, have achieved higher precision [118, 150]. EQGraphNet[172], a deep graph network model with residual connections, leverages the relational structure of seismic parameters represented as graphs to enhance single-station earthquake magnitude estimation; the graph structure is used also by [33]. Similarly, ConvNetQuake [126], is a CNN for detection and location using 3C waveforms from a single station. However, the use in real-time applications of ML models for magnitude and hypocenter estimation is limited because they often rely on large time windows [110, 112, 183].

Here, we show how Large Language Models can be used for earthquake location and magnitude estimation. We use TinyLlama [180], which is a nimble, open-source version of LLaMA [156] (Large Language Model Meta AI) –the family of open-source LLMs developed by Meta and Facebook AI. We adapt TinyLlama for seismic applications and demonstrate how a pre-trained LLM can serve as a backbone for earthquake location and magnitude estimation. Using an LLM instead of a deep learning approach is motivated by several factors. First, ML and DL methods require large volumes of labeled data for training and often struggle to generalize across different contexts. Second, by

leveraging an LLM, we can bypass the typical multi-step process, in which location is first estimated and then used to compute magnitude. Instead, we directly infer both parameters by appropriately formatting the model input and output. Lastly, LLMs have recently achieved the state of the art in a wide range of text-related tasks, demonstrating extraordinary capabilities in capturing non-linear and long-term relationships between input tokens. This has prompted growing interest in their potential use for numerical applications, like time series forecasting. The excellent results in this domain - shown by models such as PromptCast [176], Time-LLM [71], LLMTime [58], Chronos [8]- combined with the convenience of using pre-trained models, motivated our choice to adopt this architecture for seismology.

## 4.2 An LLM for earthquakes

We use the LLM TinyLlama [180], which is a decoder-only model trained on up to 3 trillion tokens. Fine-tuning is done using seismic data from INSTANCE [104] and the LoRA [67] technique to freeze part of the pre-trained model weights and reduce time and computational resources (Fig. 4.1a). The model is designed to be integrated into a regional EEWS, using data from several stations and



**Figure 4.1:** LLM model structure and workflow. a) Model summary showing 4 input values (station latitude and longitude,  $\Delta T$ , and PGV), the decoder block followed by fine-tuning using the Low-Rank Adaptation (LoRA) method of freezing part of the pre-trained model weights, and 5 output values (earthquake latitude, longitude, depth, magnitude and origin time). b) Map showing an earthquake (yellow star) and local seismic stations (triangles, red stations were not used because s-waves were not detected). c) Seismograms from 3 stations showing vertical component of ground velocity and time delay between p-wave arrivals  $\Delta T$ . Zoom shows time window for measurement of peak ground velocity (PGV).

relying on the first few seconds of the seismic waveforms recorded by surface seismometers. We use only the stations for which the P-arrival time is within 5 sec of the arrival for the closest station. The input to our model (Fig. 4.1) includes station latitudes and longitudes, the P-arrival time difference relative to the closest station  $\Delta T$ , and the peak ground velocity (PGV) on the vertical component, measured within a  $\sim 0.2$  s window centered on the P-wave (see zoom of 4.1c). Unlike other deep learning models (e.g. [112], [30]), we use only a small segment of the seismogram and a single parameter extracted from the ground motion (PGV). Our technique does not rely on a seismic velocity model or S-wave information - data typically used for earthquake location [81, 98]. Instead, our LLM uses only the input data to estimate the earthquake’s latitude, longitude, depth, magnitude and origin time (Fig. 4.1).

### 4.2.1 INSTANCE dataset

INSTANCE is an open-access, benchmark dataset containing waveforms and earthquake metadata of earthquakes in Italy and neighboring regions designed for machine learning studies [104]. It



includes over 1.1 million 3-component records from over 54 thousand earthquakes recorded by 600+ seismic stations in the time period January 2005 to January 2021. Table 4.1 shows the distribution of earthquakes by magnitude for INSTANCE. Because of the Gutenberg-Richter frequency magnitude relation, there are many more small events than large events. In particular, around 94% of the earthquakes have a magnitude smaller than 3 and there are only a few earthquakes larger than magnitude 5.

#### 4.2.2 Data selection

We culled the dataset based on a few criteria, selecting only events recorded from velocity seismograms and only those with INGV bulletin S-wave arrival times. This step is justified by our final verification study (see Discussion section) where we compare results for P-wave data with those for S-wave data. Motivated by applications to EEWS, where rapid estimation of earthquake location and magnitude is critical, we use only data from stations where the P-wave arrival occurred within 5 seconds of the first detection. In multi-station approaches, there is an inherent trade-off between prediction latency and the number of stations used: including more stations generally improves accuracy, but at the expense of extending latency and delaying early warning. By limiting our input to stations with P-wave arrivals within 5 seconds, we focus on local stations typically located within 40 km of the epicenter. Figure 4.1b shows an example of the station selection. Our dataset consists only of events with a sufficient number of waveforms for accurate predictions. Thus, we selected only events that after the selection steps described above still have at least 5 waveforms recorded. For all records remaining after the selection phases, we computed the PGV using in of m/s. After this process we had 30,879 earthquakes, and 281,591 seismic traces with magnitudes ranging from 0 to 6.5 (Table 4.1).

**Table 4.1:** Earthquakes used for the LLM. Here, intervals are denoted [a, b), where a is included and b is excluded.

| All events in the INSTANCE Dataset |        |        |        |        |        |        |        |
|------------------------------------|--------|--------|--------|--------|--------|--------|--------|
| Magnitude Range                    | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) |
| Number of Events                   | 4462   | 15249  | 30845  | 3106   | 315    | 28     | 3      |
| Our Final Dataset                  |        |        |        |        |        |        |        |
| Magnitude Range                    | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) |
| Number of Events                   | 2562   | 9654   | 16941  | 1537   | 162    | 21     | 2      |

#### 4.2.3 Data formatting

LLMs work with formatted token strings. TinyLlama allows up to 2048 tokens, and numbers are tokenized digit by digit, meaning each digit represents a separate token. Thus, we multiplied each PGV (measured in m/s) by  $10^5$  to limit leading zeros. To convert all real numbers into integers, we multiplied latitudes, longitudes, and PGV values by  $10^4$ , all other parameters by  $10^2$  - effectively removing decimal places and avoiding floating-point representations. For each event, we then concatenated all tokenized input data from all stations. Due to constraints on the maximum input length, we did not prepend any explanatory text before the parameters. Although typical LLM fine-tuning includes a natural-language instruction to specify the task (e.g. PromptCast [176]) we deliberately omitted such prompts. Instead, we evaluate whether the model knows, or can learn,

from leveraging previous knowledge about token relationships without explaining the problem. Input strings are formatted as sequences of  $(lat, lon, PGV, \Delta T)$  *new line*, where the *new line* character is used to split each station. The output token string contains the predicted earthquake parameters in the form:  $lat, lon, depth, magnitude, origin\ time$  (Figure 4.1). An example of input and output of the model is shown in C.0.1 .

#### 4.2.4 Training

We split the dataset into training, validation and testing sets using an 80% – 10% – 10% ratio, ensuring that the distribution and range of earthquake magnitudes remained consistent across all three subsets. Lastly, each pair of input-output for the events in the training and validation sets was formatted to align with the TinyLlama structure. Because we use the chat version [155], the input and output were concatenated as strings using the special tokens  $< s >$ ,  $< |user| >$ ,  $< |assistant| >$ , and  $< /s >$  to delimit the input and output segments, as well as to mark the start and end of the sequences. Test set events were formatted similarly, but without the output part.

The model was trained for 30 epochs using 2 NVIDIA GeForce RTX 4090 GPUs, with 24 GB VRAM each. We used a small learning rate of  $4 * 10^{-4}$  with a cosine scheduler, a batch size for device of 4 and the Adam optimizer. All the other hyperparameters are given in Table 4.2. We

**Table 4.2:** Training Parameters

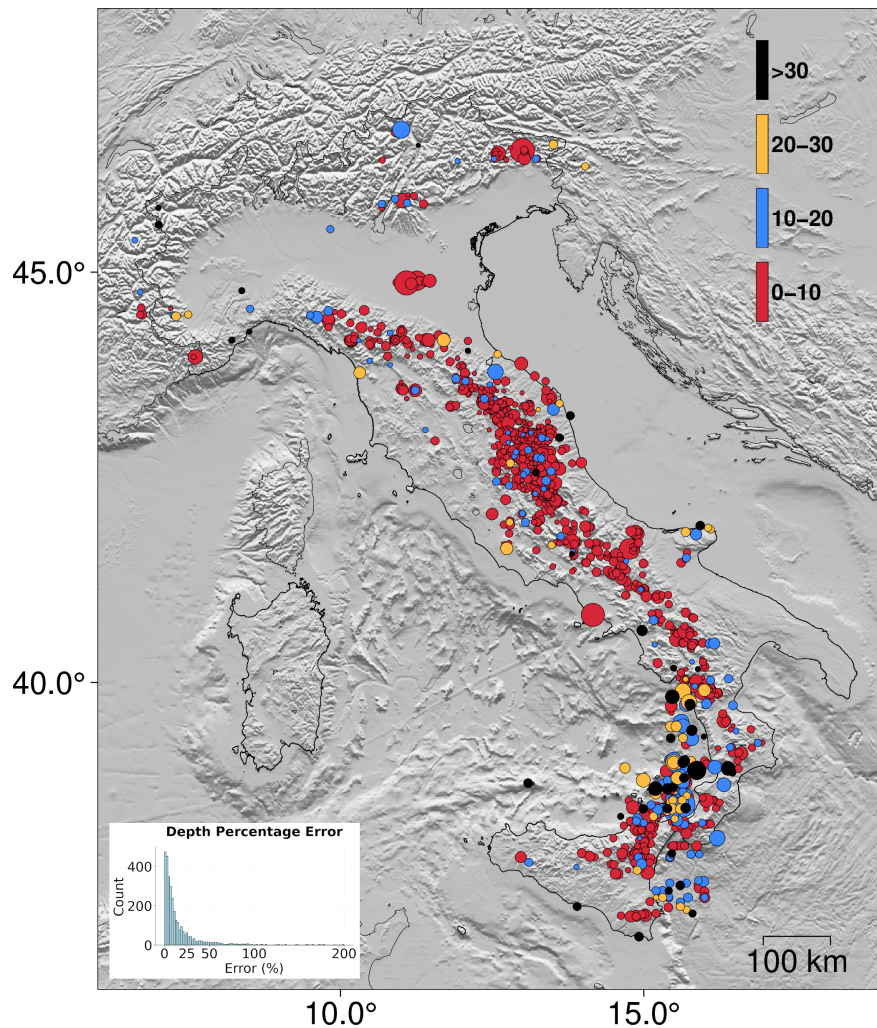
| Parameter                  | Value  |
|----------------------------|--------|
| Epochs                     | 30     |
| Batch Size for Device      | 4      |
| Gradient Accumulation Step | 1      |
| Learning Rate              | 0.0004 |
| Learning Rate Scheduler    | cosine |
| Weight Decay               | 0.001  |
| Maximum Gradient Norm      | 0.3    |

used the LoRA fine-tuning technique with a rank of  $r = 8$ ,  $\alpha = 16$  and a dropout rate of 0.4. This approach allowed us to train only 1.1 million parameters out of the original 1,101 million, resulting in a  $\sim 99\%$  reduction in trainable parameters and a substantial gain in computational efficiency.

## 4.3 Results

### 4.3.1 Test set performance

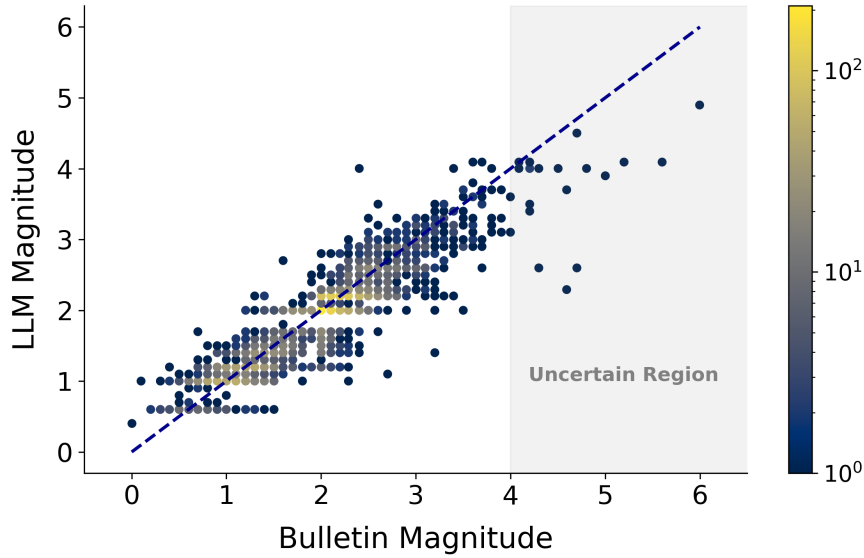
We assessed the model performance on 3,094 test events by comparing each estimated parameter with the corresponding values from the INGV bulletin provided in INSTANCE. Figure 4.2 summarizes the epicentral errors: symbol size is proportional to event estimated magnitude and color indicates the binned location error relative to the bulletin values assumed as ground-truth.



**Figure 4.2:** Map of earthquake epicenters determined by the LLM. Symbol size is proportional to estimated magnitude and color is binned by the epicentral location error, with error ranges (in kilometers) specified in the legend. Note that most event locations are within 10 km of the INGV bulletin values, which are considered our ground-truth values. Inset shows depth error distribution, and indicates that depth errors are small for most events.

Note that the majority of test events fall within the smallest error bin (0 – 10 km). There are a few events with large location errors (black dots) mostly located in the Tyrrhenian Sea offshore the Calabrian coast, where both the low station density and the large azimuthal gap contribute to a significant degradation in location accuracy. The inset to Figure 4.2 shows absolute depth error as a percentage and highlights that, for most events, the depth error remains small. Overall our model shows promising results with a mean error of 4 km for the epicenter, 6.8 km for the hypocenter, a Mean Absolute Error (MAE) of 0.6 s for the earthquake origin time and MAE of 4.3 km for depth.

Regarding earthquake magnitude the model achieves a MAE of 0.2 and a Root Mean Squared Error (RMSE) of 0.28. Figure 4.3 shows a comparison of bulletin and LLM magnitudes. As expected, larger errors are associated with the largest magnitude events. Specifically for events with  $M \geq 4$ , a trend of underestimation can be observed. This is consistent with the unbalanced nature of the dataset, which follows a typical Gutenberg-Richter distribution where the majority of events are small. Consequently, the LLM does not see enough larger events to recognize them efficiently.



**Figure 4.3:** Comparison of INGV bulletin magnitudes with those estimated by the LLM. Symbol color is based on the number of events at each location (scale bar at right). The gray region above  $M 4$  highlights the range where there are few events and estimates.

Additional plots of the overall error distributions can be found in the Supplementary Materials (C.1 C.2).

#### 4.3.2 Toward EEWS: comparison of LLM and rapid automated results

To evaluate the applicability of our LLM model in a real-time scenario, we compared its performance with that of the standard INGV rapid, automated method deployed for seismic surveillance. INGV’s automatic location and magnitude estimates are determined by the Earthworm software [44]. INGV has configured Earthworm to provide earthquake locations and magnitudes at 3 successive times labeled as ‘0’, ‘1’ and ‘2’. The ‘0’ solution is generated as soon as  $\geq 8$  seismic phase arrivals are associated with an event. This means that a solution is typically available about 30 seconds after the origin time. The INGV ‘1’ solution is provided when events have at least 30 phases identified, which typically occurs 1 minute after the earthquake, and the final automatic solution, ‘2’, is available when the solution has been stable for the last 60 s; that is, when no additional phases have been associated within 60 s. This implies a waiting time of at most 5 to 6 minutes after the initial time for larger size earthquakes.

For a fair comparison, we used our model, which is trained with bulletin data, with automatic phases picked in real time. Specifically, we selected the automatic phases used by the INGV for the 3 different solutions, keeping only those within five seconds of the first detected wave, thus implying

the use of a smaller number of stations in our model. Table 4.3 shows our LLM and the INGV performances compared against the bulletin values, which are assumed to be ground-truth. Using automatic seismic phases, our LLM performs better on average than all three of the rapid INGV solutions. The LLM error estimates decrease when comparing methods 0, 1, and 2, as expected given that solution 0 involves fewer data than solutions 1 and 2. Note that INGV solution 2 requires the longest latency but can have fewer data from seismic phase picks than solution 1 and larger average error because many more events are identified. In contrast, the LLM has smaller errors even though it uses fewer data. Our location estimates have an average epicentral error of 6.5 km whereas the INGV rapid estimates are on average 8.5 km. On average, our LLM '0' solutions are

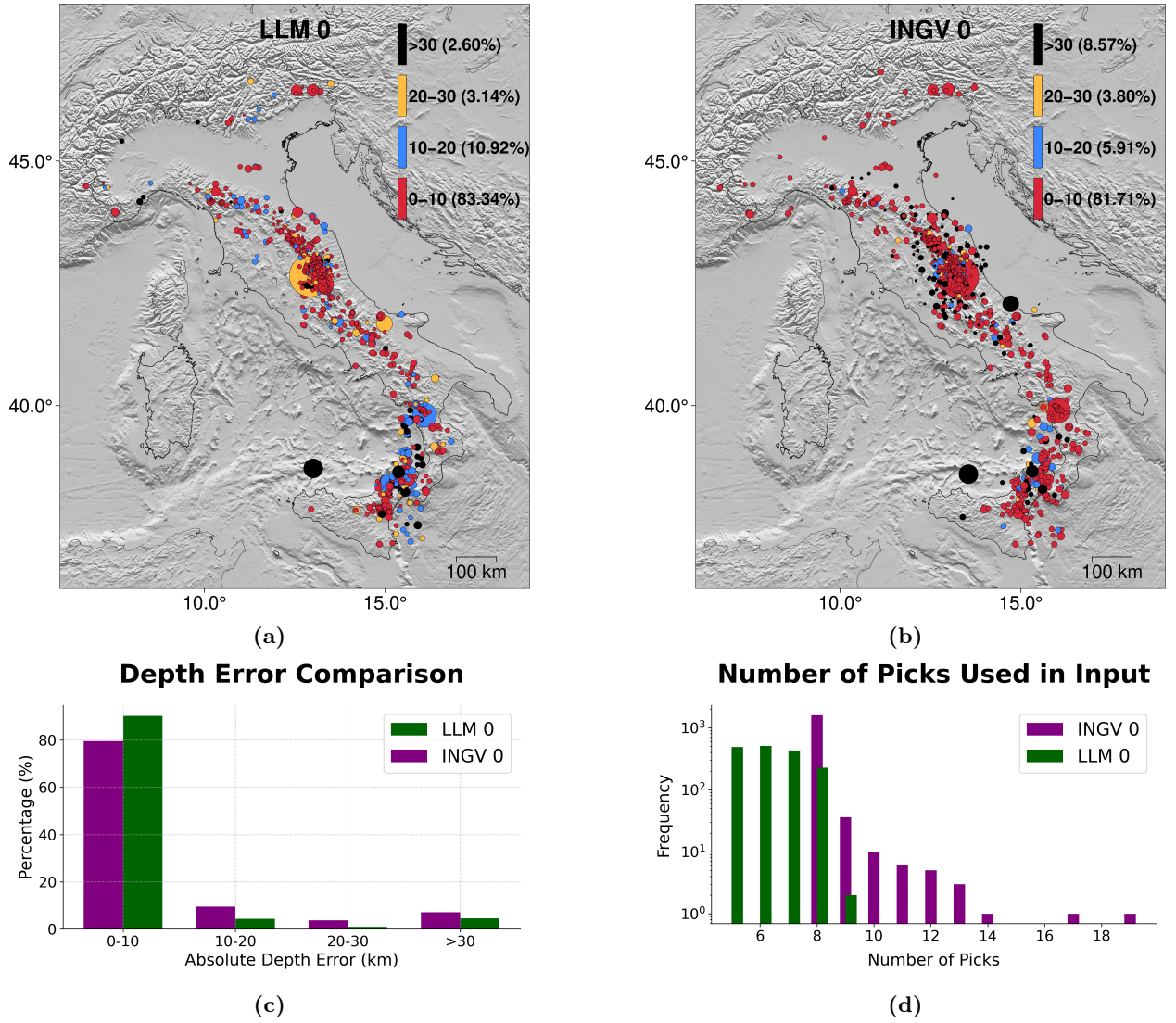
|                             | Fine-Tuned LLM |      |      | INGV  |       |       |
|-----------------------------|----------------|------|------|-------|-------|-------|
|                             | 0              | 1    | 2    | 0     | 1     | 2     |
| Number of Samples           | 1657           | 889  | 2241 | 1657  | 889   | 2241  |
| Average Number of Picks     | 6.2            | 10.8 | 9.2  | 8.1   | 34.8  | 30.4  |
| Mean Epicentral Error (km)  | 6.26           | 5.61 | 5.29 | 8.62  | 9.84  | 10.68 |
| Mean Hypocentral Error (km) | 11.05          | 9.51 | 8.62 | 15.05 | 15.41 | 16.43 |
| MAE Magnitude               | 0.30           | 0.32 | 0.25 | 0.25  | 0.14  | 0.17  |
| MSE Magnitude               | 0.18           | 0.22 | 0.13 | 0.15  | 0.08  | 0.15  |
| MAE First P Travel Time (s) | 1.10           | 1.14 | 0.90 | 1.76  | 1.89  | 1.85  |

**Table 4.3:** Comparison of results from the LLM and INGV automated methods. In both cases the INGV methods 0, 1, and 2 are used. *Fine-Tuned LLM* columns show our model performance using data recovered using INGV methods 0, 1, 2. *INGV* columns show performances for the same three methods. Note that the LLM model performs better on average for epicentral and hypocentral location and that magnitude estimates are similar.

better for epicentral and hypocentral location and slightly worse than the INGV model '0' solutions for the magnitude estimation; however the differences are not significant. For example, the INGV '0' epicentral error is 8.6 km and ours are 6.3 whereas the INGV magnitudes have a MAE of 0.25 whereas those from our model have a MAE of 0.30. The differences are small but increase for each solution, with INGV solution '2' results having larger location errors than our LLM solution '2'.

Given the good results obtained in the location comparison (Table 4.3) we further analyze the spatial distribution of the differences between our epicenter estimates and the INGV '0' solution in Figure 4.4. We begin with this comparison because even though the INGV solutions use more seismic phases than our method, the INGV '0' solution is the fastest available and therefore the most appropriate reference in terms of timeliness for EEWS purposes. Figure 4.4 shows that the largest errors are in the northernmost part of Italy, which is expected because there are fewer events and the LLM has less training. Conversely, in central Italy, the INGV map shows a significant number of very large errors that are not present in the LLM results, indicating that the LLM is more accurate there. Both maps show the largest errors in the sea between Sicily and Calabria, where low station density and large azimuthal gaps negatively impact location accuracy (Figure 2). Focusing on the percentage of events that fall within each epicentral error interval, we can see that for the LLM there is a decreasing trend with only 2.6% of the analyzed earthquakes having an epicentral error greater than 30 km, whereas for the INGV results this percentage is around 8%. Figure 4.4 also compares the depth error, showing that the LLM performs better.

Given the importance of waiting time for EEWS, it would be valuable to perform an experiment comparing the solution times for the LLM and the '0' rapid INGV method. However, conducting



**Figure 4.4:** Comparison of earthquake epicenter location error for our LLM (a) and the INGV '0' rapid solution (b). The LLM uses only a subset of the seismic phases used for the INGV '0' solution. Thus, these data differ from data shown in Fig. 4.2, both in terms of pick timing and the set of involved stations. Symbol size scales with the INGV bulletin magnitude, while color indicates the epicentral error range. Both plots display the percentage of events falling within each error interval. c) Depth error comparison. d) Comparison of the number of seismic phase picks used in input by the LLM and the INGV automatic method for the '0' estimates. Note that more of the LLM solutions have a depth error < 10 km (Panel c) and that the LLM solution '0' is generally obtained using fewer seismic phases (Panel d).

a real-time timing experiment was not possible. So, to get an idea, we compared the number of seismic phase picks used as a proxy for the waiting time. Figure 4.4d shows that the LLM uses fewer input data than the INGV rapid solutions. This suggests that LLM solution requires less time, because a higher number of picks in input implies a higher number of stations, thus a higher waiting time for a solution. Overall, this comparison indicates that our LLM model has promise for reducing waiting time for EEWS. Additional analysis of model performance can be found in the Supplementary Materials.



## 4.4 Discussion

We propose a new method for rapid estimation of earthquake parameters, utilizing a recently developed LLM. Intended for real-time implementation, our approach uses data from only stations within 5 seconds of the first detected wave. We fine-tuned the model using bulletin data from INSTANCE, with a sample size of around 30 thousand events, using the Lora technique to reduce the amount of resources required.

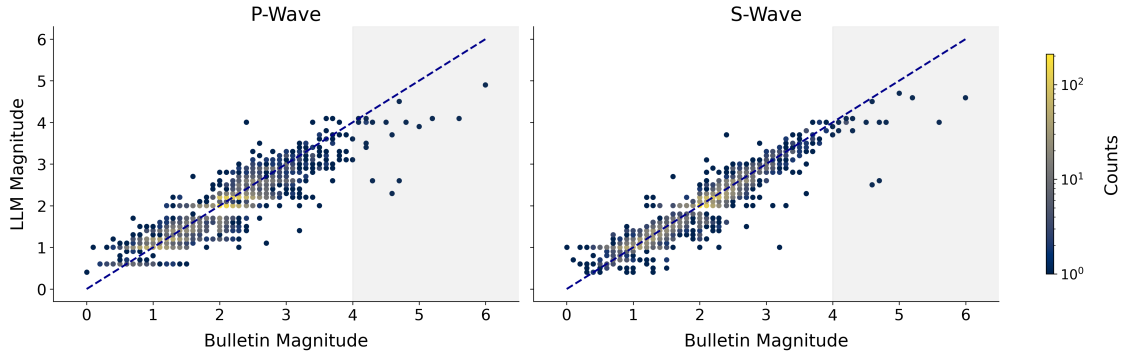
When tested with bulletin data, our model achieves promising performance for magnitude and location estimation, with small magnitude errors and average location errors of 4 km, 6.8 km and 0.6 s for the epicentre, hypocentre and origin time, respectively. We further compared the LLM with the automated procedures used by INGV. The LLM-based model outperforms INGV’s fastest solution for magnitude and location (see Fig. 4.3), with an average epicentral error of 6.26 km compared to 8.62 km for INGV. Figure 4.4d highlights how the proposed model performs better with a smaller number of input stations, demonstrating its potential in real-time scenarios where fast and efficient estimates are crucial. However, large errors in epicenter estimation can be seen in both Fig. 4.2 and Fig. 4.4a. We further analyzed the potential causes of these failures by investigating factors correlated with epicentral errors. The results indicate that epicentral error is mainly correlated with the number of input stations, and thus also with the azimuthal gap and the event’s depth (see C.3).

Our LLM method predicts earthquake magnitude accurately using only basic station information and the P-wave PGV, achieving an overall MAE of 0.2 and showing a trend of underestimation for high-magnitude events. The results are promising for EEWS but include the uncertainty of whether the final rupture size and magnitude, can be predicted from the early stages of rupture nucleation.

According to the cascade model of earthquake nucleation, events of all size begin as small events and the final size is determined by other factors [103, 109, 140]. Conversely, some studies suggest that P-wave characteristics directly reflect the earthquake’s final size [34, 45, 123], supporting the possibility of rapid magnitude estimation in EEWS. Here, we aim to contribute to this debate by testing whether, given the same data and training characteristics, there is a substantial difference between the use of P-waves alone and the use of S-waves. We trained, validated, and tested the same LLM on identical events, but instead of using P-wave PGV, we computed PGV around the S-wave arrival. Specifically, we used a time window, starting 0.5 s before the S-arrival and ending 1.5 s after, and used the average PGV for the two horizontal components. Figure 4.5 compares the LLM’s performance against ground truth (the INGV Bulletin) for P-wave and S-wave inputs. The overall comparison (Table 4.4) shows that the MAE and RMSE are smaller for the models based on S-wave data. The S-wave LLM magnitudes are slightly better, with a more concentrated distribution around the 0 error line (Fig. 4.5), and tend to reduce the underestimation for higher magnitude events. Nonetheless, even when centered on the S-wave arrival, the model still tends to underestimate large magnitude events. This indicates that while P-waves lack information

**Table 4.4:** Magnitude Estimation Results

| Metric         | PGV around P-wave | PGV around S-wave |
|----------------|-------------------|-------------------|
| MAE Magnitude  | 0.2               | 0.14              |
| RMSE Magnitude | 0.28              | 0.22              |



**Figure 4.5:** Comparison of magnitudes predicted by the LLM using PGV from P-waves and S-waves. PGV for P-waves is computed on the vertical component within a 0.2 s window centered on the P-wave arrival, while PGV for S-waves is calculated as the average of the two horizontal components within a 2.0 s window around the S-wave arrival

about the final stages of the earthquake rupture they provide a reasonably accurate estimate of the magnitude that is useful for EEWs. Additionally, this improvement in performance when using S-waves comes with an increase in waiting time for parameter estimation. When using only P-waves, the waiting time is 5s from the time of the first detection plus the computational time. In the case of S waves, we must add the additional time for the arrival of the S-waves themselves and 1.5s for the interval of the last station. Therefore, in the context of real-time monitoring, we show that with our model it is not worthwhile waiting for S-waves, as P-waves alone provide an accurate and fast estimation of the magnitude.

In conclusion, our model shows promising results in leveraging LLMs to estimate earthquake parameters through automated procedures and typical seismic data. While some outliers were observed - particularly in hypocenter estimates and for high-magnitude events, the model perform as well as traditional automatic approaches like those used by INGV and elsewhere at many other monitoring centers. In addition, the use of a small time window (less than 1 second) for the analysis of seismic waveforms allows fast and efficient estimates, a valuable property for early warning scenarios. Furthermore, while there are different studies on the application of LLMs to time series tasks [8, 58, 71, 176], to our knowledge, this is one of the first attempts to apply them to earthquake science highlighting the potential of LLMs in areas beyond traditional text-based tasks. Overall, our study confirms the ability of LLMs to provide accurate estimates of seismic parameters, paving the way for future applications of LLMs in seismology and earthquake early warning systems.



## Chapter 5

# SeismicAE: autoencoder for seismic waveforms embedding

In this Chapter we transfer an audio autoencoder and the learning process, and finetune it with seismic data. Then, we evaluate the method with critical seismology-specific downstream tasks, using it as a *Foundation Model (FM)* for *EQ* analysis even in few-shot training scenarios where traditional methods struggle. Our model accurately reconstructs seismic waveforms while reducing dimensionality by 99 % via quantizer and codebook optimized for audio. The model also predicts ground motion parameters precisely, which is crucial for *EQ* early warning tasks. Our model outperforms state-of-the-art techniques and provides a robust framework for developing *FMs* for *EQs*, underscoring the potential of transfer learning from audio to seismology. In this Chapter we move from a single task approach, to a unified framework for several tasks.

### 5.1 Overview and Related works

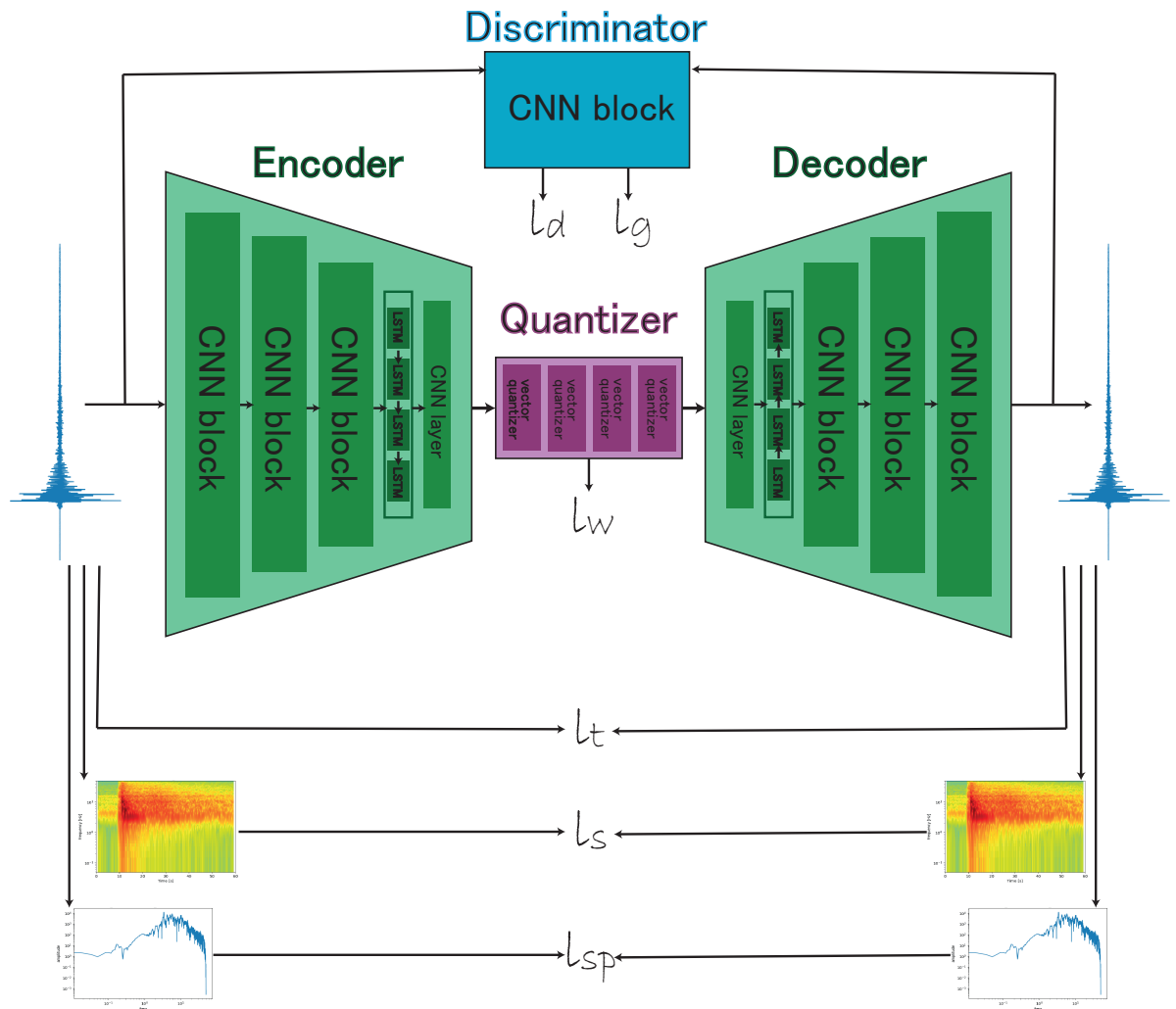
As previously explained, in the past few years *ML* and *DL* models applied to seismic waveforms have produced remarkable advances in seismology. Many of the applied methods are task-dependent [25, 47, 57, 138] and use shallow architecture techniques such as random forest [3, 160] or gradient boosting [73, 74]. Also *DL* architectures such as MLP, *Convolutional Neural Network (CNN)* [36, 65], *TFs* [90, 118], GNN [162] and others [113] are advancing rapidly in seismology.

Here we apply a self-supervised pre-training AutoEncoder AE model with the goal of using it for several downstream tasks in seismology. The adopted model architecture is EnCodec [43], a neural audio compression model with a convolutional-based encoder-decoder architecture. The model accepts audio traces to create a latent representation through the encoder network. The quantization layer produces a compressed representation via vector quantization. This compressed latent representation is then passed through a decoder network which reconstructs the original signal. We use the pre-trained version of the model, which was trained using multiple audio domains that include speech, noisy speech, music and general audio. Here, we present the first study to transfer the EnCodec model capabilities from general audio sources to raw seismic waveforms.

Seismic traces are acoustic signals, with certain important differences from audio, which require adjustments in training, fine-tuning and modeling aspects. To address the challenge of transitioning from audio to seismic waveforms, we improve the training procedure and fine-tune the pre-trained

model using seismic traces to obtain an Auto-Encoder for seismological applications (*Seismic Waveform Auto-Encoder (SeismicAE)*), which uses Acoustic Emissions. We demonstrate that pre-trained models can be used in conjunction with a few convolutional adaptation layers to learn downstream tasks. We also show that it is possible to perform tasks with only small datasets, which is often a limitation in seismology, especially when the datasets are highly unbalanced (e.g. in a year, we have  $\sim 1$  *EQ* with magnitude greater than 8, and  $\sim 1\text{M}$  *EQs* with magnitude greater than 2, following the Gutenberg-Richter Law; building a balanced catalog with a lot of very strong *EQ* is challenging). Our proposed *SeismicAE* represents a transferred *FM* since it serves seismic representation learning, as well as reconstruction and downstream tasks, with groundbreaking potential for future work.

## 5.2 Data and Methods



**Figure 5.1:** *SeismicAE* model architecture showing the encoder, decoder, and quantizer, with the six losses defined as the time series reconstruction ( $l_t$ ), mel-spectrogram ( $l_s$ ), spectra ( $l_{sp}$ ), quantizer ( $l_w$ ), the perceptual loss from the discriminator ( $l_d$  and  $l_g$ ).

### 5.2.1 Data

To develop *SeismicAE* we started with two well-established datasets in seismology: *STEAD* [116] and *INSTANCE* [104]. *STEAD* [116] is a large, benchmark dataset, with 1,200,000 *EQ* waveforms containing three-component seismic signals for magnitude from M0.5 to M7.9, along with noise examples. Waveforms labeled as noise do not contain *EQ*s, but do contain information about Earth, such as oceanic waves, tides, weather and human activities, all of which are applicable to active research areas. *STEAD* includes metadata related to the waveforms and epicentral distances, defined as the horizontal distance between the station and the source, less than 350 km. The data is a global compilation of historical records collected between January 1984 and August 2018.

The *INSTANCE* benchmark dataset [104] comprises  $\sim 1,300,000$  local three-component waveforms containing  $\sim 450,000$  *EQ*s (average of 21 events per station, while *STEAD* generally has each *EQ* recorded only by one station) that occurred in Italy with magnitudes ranging between M0.0 and M6.5 from January 2005 to January 2020. Metadata related to the waveforms are provided. Noise traces are also available in *INSTANCE*.

We select a small subsample of about 6,000 seismic waveforms from both *STEAD* and *INSTANCE* for testing. We keep %10 of the datasets for validation,  $\sim 200,000$  traces and finetune the model using the remaining  $\sim 2,000,000$  traces. The dataset is normalized in a trace-specific fashion. In this way we ensure that each datasample (seismic waveform) is in a range  $[-1,1]$  and that each component maintains its proportional amplitude between channels (i.e. for each *EQ* recording, the channel with the biggest amplitude will range up to 1 and the others will be smaller). This also ensures the adaptability of the model on a variety of use cases for different tasks and datasets.

### 5.2.2 Methods

#### Model Details

The architecture of *SeismicAE* is the same as EnCodec [43] with the training procedure adapted for seismology (Figure 5.1). In the model, 1 channel of the time-domain seismic record is input to a 1D convolutional-based encoder with convolution blocks, followed by a 2-layer *Long Short-Term Memory* (*LSTM*) to learn the temporal component, and a final *CNN* layer. The compressed quantized latent representation is decoded in the upsampling network, which mirrors the encoder to recover the original signal.

The training strategy combines a loss balancer to stabilize training, a reconstruction loss term, a perceptual loss term using a discriminator network, and the Residual Vector Quantization loss term. The reconstruction loss term aims to minimize the L1 loss between the target signal and output over the time domain ( $l_t$  in Figure 5.1), and to minimize the L1 and L2 distances over the mel-spectrogram between the target signal and model generated output in the frequency domain ( $l_s$  in Figure 5.1).

The perceptual loss term is based on a multi-scale *STFT* discriminator (i.e., *STFT* to evaluate the quality of generated data). The discriminator (Figure 5.1) contains sub-networks with 2D convolutional layers that focus on the multi-scaled complex-valued *STFT* using the concatenated

real and imaginary numbers. There is an adversarial loss for the generator ( $l_g$  in Figure 5.1) and a combined loss for the discriminators over the dimensions ( $l_d$  in Figure 5.1) that is trained to minimize the hinge-loss adversarial loss function. A Residual Vector Quantization loss term compares output of the encoder and the quantized value. The L2 loss is computed between the current residual and the nearest entry in the corresponding codebook ( $l_w$  in Figure 5.1). The pretrained codebook [43] is optimized for audio and is applied without any modification.

### Challenges in transitioning from audio to seismology

Moving from audio signals to seismic waveforms is challenging even though the data contain multiple similarities. Key differences include the fact that audio is vibrational p-wave energy caused by oscillation of air molecules while seismic wavefs are generated by energy release and acceleration due to shearing motion of an *EQ* fault. Elastic waves in Earth propagate through rock, soil, and fluids of varying properties. Other difference include the recording instruments: microphones or transducers for audio waveforms and mechanical seismometers for seismic data. Additionally, *EQs* generate at least 2 main waves (P and S, i.e. transverse and longitudinal waves) while audio has only longitudinal. The frequencies of interest are also different: the audible frequency ranges for sound generated by humans typically range from about 85 Hz to 255 Hz. While in seismology the phenomena of interest are between 0.01-50 Hz. Lower frequencies (0.01 - 1 Hz) seismic waves are often associated with volcanic tremor and long-period events; intermediate frequencies (0.1 - 1 Hz) are studied to understand the rupture process and the source mechanism, also at long distances; high frequencies (1 - 50 Hz) are important to study the rupture process within the fault zone during an *EQ* ground shaking and site effects. The *SNR* in audio depends on several factors and we report the averages for a few scenarios . During normal speech the *SNR* varies from -2.5 dB in an office meeting [100] to 4.0 dB at a restaurant [100] and up to 15 dB in classrooms [170]. For high quality recordings such as audiobooks, the average *SNR* is 32 dB in the case of LibriVox [10]. For *EQs* the *SNR* depends on the distance and magntiude of the event. In the near-field (within about 10 km of the epicenter) the *SNR* can exceed 30 dB for strong events [1], for regional *EQs* at distance within a few hundred kilometers it often ranges from 5 to 20 dB [19], and for teleseismic *EQs* at distances greater than 1000 km the *SNR* generally range from 0 to 10 dB [134]. For smaller events the *SNR* can be even lower and inhibit detection in noisy environments [75, 76].

### Fine-Tuning Strategy

For *SeismicAE* we start with the pretrained model from Meta [43], trained on audio, and fine-tune it on seismic data. The first challenge is that seismic data consists of 3 channels (N/S, E/W, Vertical) whereas the pre-trained model is designed for a single input channel. For this reason we fine-tuned the model one channel at a time.

The difference in sampling rate is another challenge. Seismic waveforms are typically sampled at 100 Hz, however, the pre-trained Meta model uses audio at 24 kHz. To adapt, we use the original seismic recording rate but keep the number of data points consistent: the length of all our seismic data traces is 6000 data points, 60 seconds, compared to the audio which is about 2 milliseconds. In practice, the difference does not affect the training procedure or the results and the content of the signal is well understood by the model.

The training strategy is modified by including an additional loss term for the spectra, which aims to minimize the L1 loss between the target spectra and output spectra ( $l_{sp}$  in Figure 5.1). This change was included after inspecting a preliminary result where the model performed poorly at the higher frequencies. Including this additional loss term greatly improved the performance. To evaluate the results and select the best model we use the total validation loss as the summation of  $l_t$ ,  $l_s$  and  $l_{sp}$ .

### 5.3 Results

To evaluate the effectiveness of *SeismicAE*, we conduct several tests. Initially, we assessed the reconstructed traces, which was the primary objective during both training and fine-tuning. Additionally, to measure the quality of the model’s representations, we select two downstream tasks: one for classification and one for regression.

For the downstream tasks, we take the last layer of the encoder as a representation, thus reducing the dimensionality of the input trace by  $\sim 60\%$ . Using the quantizer layer as a representation for downstream tasks is a common practice. However, in this case, it’s not the ideal solution since the codebooks are optimally chosen for the audio task from the original work, and codebooks are not trainable, making it hard to achieve the best functioning on seismology without modifications. By testing the quantizer we discovered that it cannot effectively represent our input signal, as further shown in 5.3.1. However allowed to perform fine-tuning and to effectively reconstruct the input traces.

#### 5.3.1 Reconstruction

| Train              | Test     | With Quantizer |        |        |         | Without Quantizer |        |        |         |
|--------------------|----------|----------------|--------|--------|---------|-------------------|--------|--------|---------|
|                    |          | MSE            | RMSE   | MAE    | R2      | MSE               | RMSE   | MAE    | R2      |
| Stead and Instance | Stead    | 0.0178         | 0.1062 | 0.0690 | -0.5252 | 0.0099            | 0.0776 | 0.0492 | 0.1342  |
|                    | Instance | 0.0120         | 0.0905 | 0.0498 | 0.1161  | 0.0087            | 0.0758 | 0.0391 | 0.3396  |
|                    | Norcia   | 0.0155         | 0.0966 | 0.0618 | -0.3417 | 0.0168            | 0.0960 | 0.0614 | -0.4024 |
| Stead              | Stead    | 0.0179         | 0.1058 | 0.0692 | -0.5163 | 0.0106            | 0.0802 | 0.0512 | 0.0819  |
|                    | Instance | 0.0125         | 0.0917 | 0.0505 | 0.0784  | 0.0090            | 0.0766 | 0.0399 | 0.3220  |
|                    | Norcia   | 0.0148         | 0.0959 | 0.0620 | -0.3081 | 0.0173            | 0.0952 | 0.0615 | -0.3705 |

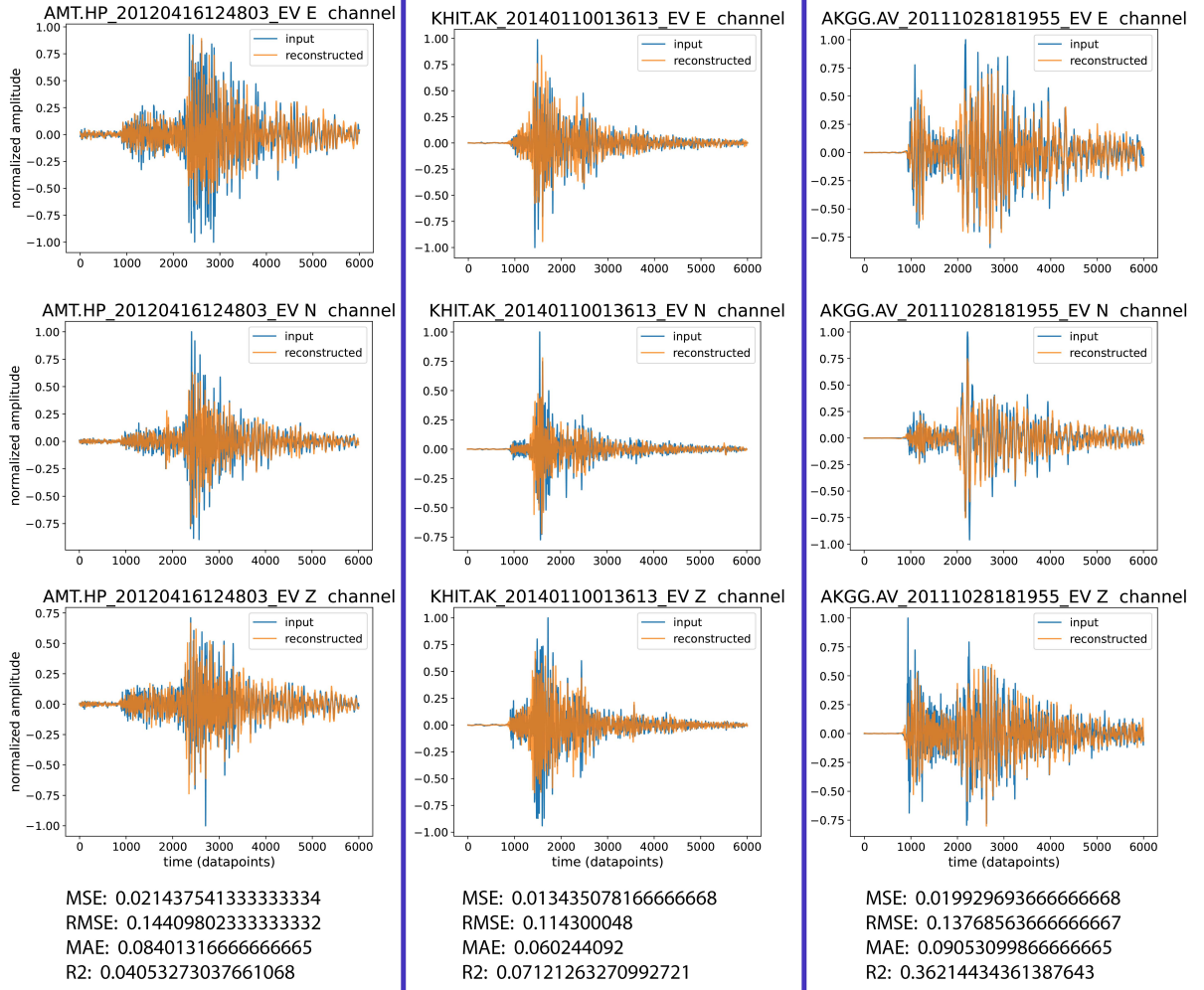
**Table 5.1:** Performance metrics for models trained on *STEAD* and *INSTANCE* datasets, including *STEAD* noise, and on *STEAD* only (without noise). We evaluate the models on the reconstruction task, showing the results with (99 % compression) and without (60 % compression) the quantizer. Metrics used: **MSE** (Mean Squared Error) measures the average squared prediction error. **RMSE** (Root Mean Squared Error) indicates the standard deviation of prediction errors. **MAE** (Mean Absolute Error) measures average prediction accuracy. **R2** (R-squared) indicates the proportion of variance explained by the model, ranging from -1 to 1, with higher values indicating better fit. The Table compares performance across *STEAD*, *INSTANCE*, and Norcia datasets.

One original motivation behind the EnCodec architecture [43] was audio compression, based on the rapid growth of internet traffic. EnCodec’s goal is to perform signal compression to minimizing the

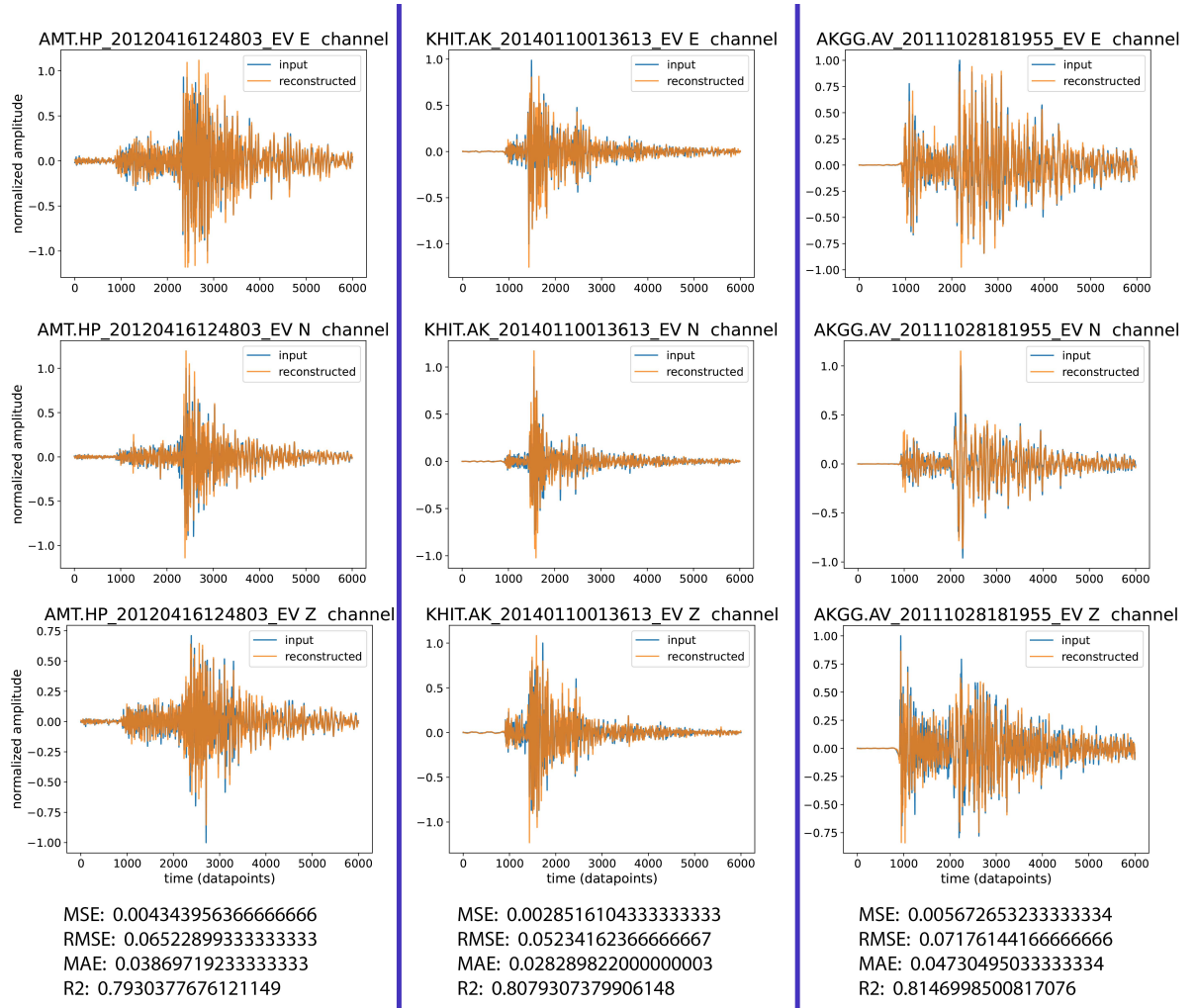
bit-rate of a sample while also minimizing loss and distortion according to a given metric correlated with human perception. Our goal for fine-tuning *SeismicAE* is not exactly the same; however a primary objective is similar: that of using a generic approach that minimizes data size while retaining essential information. Our representation is intended for use across multiple tasks, drawing inspiration from the concept of *FMs*. However, since the goal of the model during training and fine-tuning is trace reconstruction, we also evaluate the latent representation by reconstructing the original data from the compressed form.

In Table 5.1 we report metrics for reconstruction using a set of combinations for dataset used in training and evaluation. We also evaluate the model using the dataset from the M6.5 2016 Norcia *EQ* of central Italy. This dataset was adopted for the downstream task discussed in 5.3.2 and contains seismic traces recorded from September to December 2016 by the NRCA seismic station, for a total of  $\sim 7000$  seismic events. As explained in 5.2.2, the quantization is our bottleneck, because the codebook, specifically designed for audio and not for seismology, is not trainable. We support this statement showing the reconstruction of the encoder decoder, also without passing through the quantizer.

In Figure 5.2 we provide three examples of reconstruction for a qualitative evaluation, using the model as is, with the quantizer. In Figure 5.3 we do the same, excluding the quantizer during the reconstruction procedure.



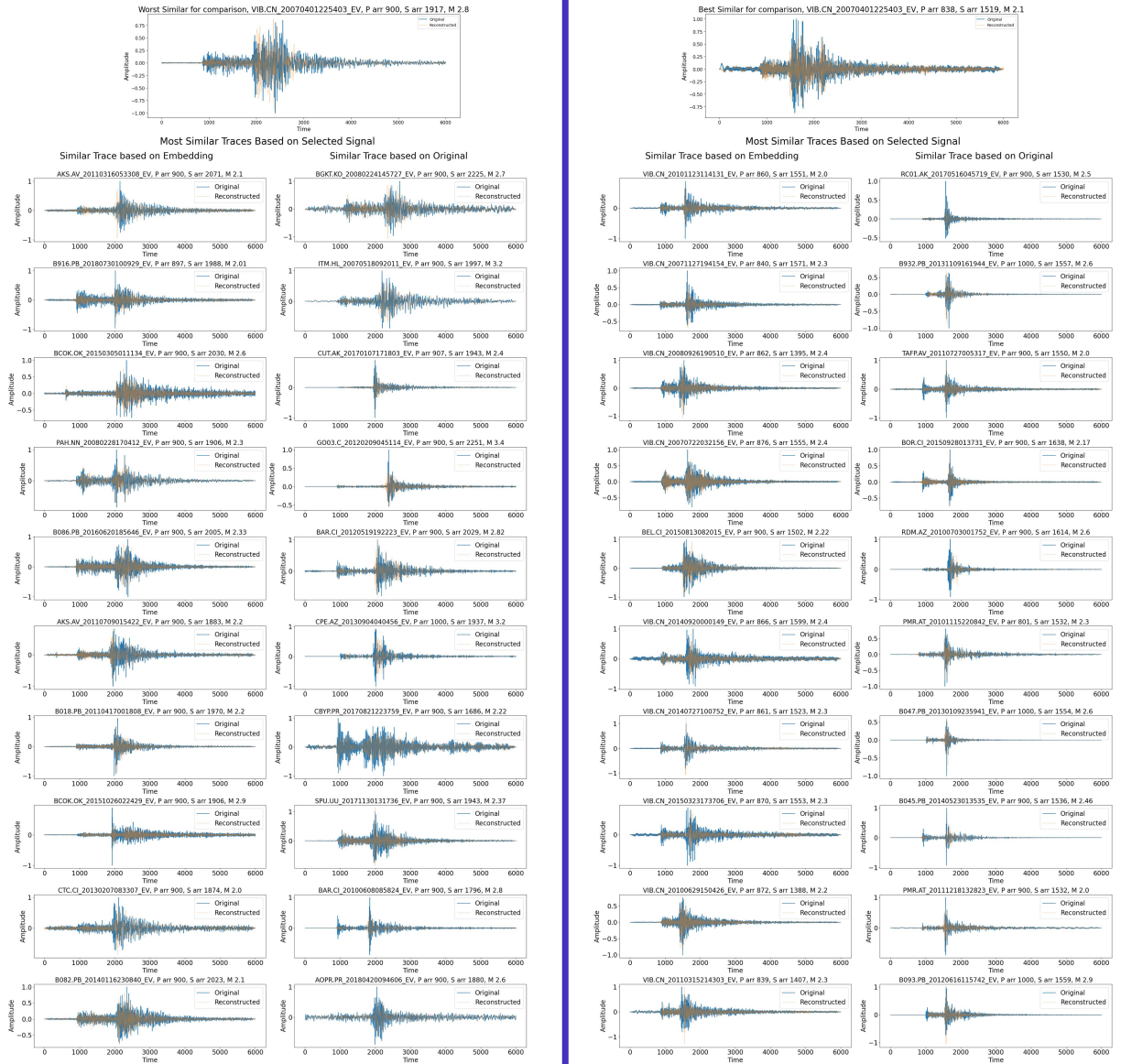
**Figure 5.2:** Three examples of reconstruction for seismic traces from *STEAD*. The overall shape and major features of the waveforms are well-preserved in the reconstructions. Some high-frequency details are smoothed out in the reconstructed versions. The metrics (MSE, RMSE, MAE, R2) provide quantitative reconstruction quality measures.



**Figure 5.3:** Three examples of reconstruction for seismic traces from *STEAD*, excluding the quantization. The reconstructions (orange) match the original traces (blue) more closely than in Figure 2. High-frequency details are better preserved without the quantizer. The metrics confirm that the overall fit is better.



## Retrieval



**Figure 5.4:** 2 examples of similar trace retrieved based on embedded and original seismic waveform. We display the best and worst examples based on difference between the selected P arrival, S arrival and Magnitude, versus the other detected similar traces. Only channel E is presented. The "Best example" shows a very close match between the target trace and the retrieved similar trace, preserving important seismic features like P and S wave arrivals and overall envelope shape. The "Worst example" still shows similarity in overall shape, but with more noticeable differences in timing and peaks.

| Feature        | Raw signal | Embedded signal |
|----------------|------------|-----------------|
| P arrival (ms) | 44.9416    | <b>19.2546</b>  |
| S arrival (ms) | 116.6149   | <b>74.9390</b>  |
| Magnitude      | 0.5820     | <b>0.4256</b>   |

**Table 5.2:** For each trace of the test dataset, we make a comparison with all the others, as explained in section 3.1 Retrieval. Here we compute the absolute difference value between features of interest (P arrival, S arrival, Magnitude) of each target trace vs the 10 most similar traces. The lower the better.

To gauge *SeismicAE*'s efficacy in representing traces with the embedding intended for downstream tasks (the last layer of the encoder), we conduct a trace retrieval evaluation. This evaluation method relies on the similarity of embeddings between a target trace from the test dataset, and the most analogous one, again within the test dataset. We compare the traces based on Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) to measure similarity, defined as

$$SI - SDR = 10 \log_{10} \left( \frac{\sum_{t=1}^T (\alpha s_t)^2}{\sum_{t=1}^T (s_t - \hat{s}_t)^2} \right) \quad (5.1)$$

Where  $s_t$  represents the target signal to be compared, at time  $t$ ;  $\hat{s}_t$  represents the signal to be compared at time  $t$ ;  $\alpha$  is a scaling factor that minimizes the energy;  $T$  is the total number of time points or samples in the signals.

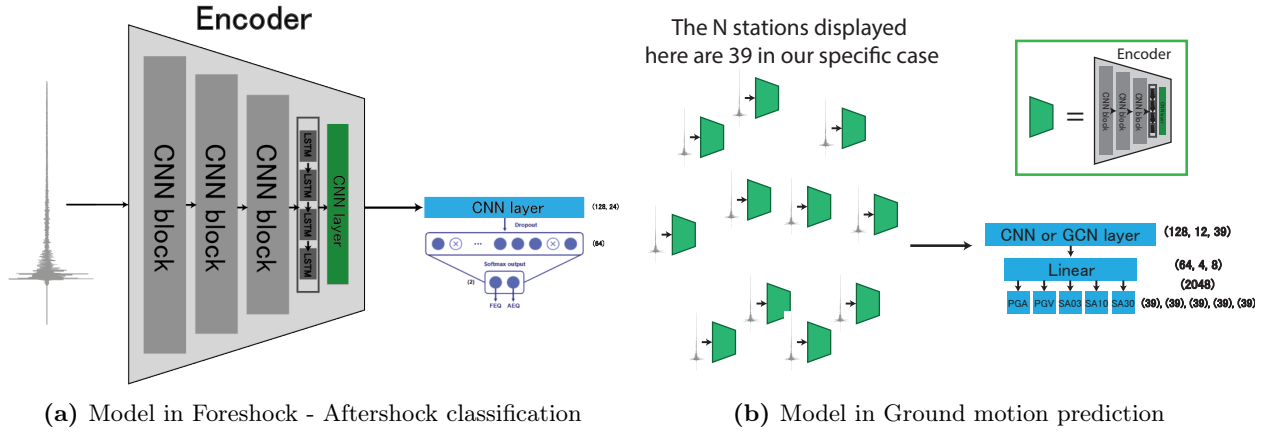
We compute SI-SDR between the embedded version of the target signal and all the other embedded traces in the test set to evaluate the embedding capability of keeping important information. For comparison, we also compute SI-SDR between the raw original target signal and all the other raw original traces in the test set. This approach enables us to assess the model's proficiency in accurately reconstructing traces based on their embedded versions. In Table 5.2 we report quantitative comparisons in terms of P and S wave arrivals times as well as Magnitude. Those are three important features of a seismic trace, and this is of high interest to be able to properly encode them in the embedded version of the signal. P waves (Primary waves) are the fastest seismic waves and are the first to reach the seismic station during an *EQ*, causing longitudinal ground motion in the direction of waves propagation (longitudinal oscillations); S waves (Secondary waves) are slower than P waves and arrive after them, causing the ground to move orthogonal to the direction of wave propagation (transverse oscillations).

In Table 5.2 we report the absolute values of the difference between the features of the target trace compared with the features of the retrieved 10 most similar traces, both in terms of embedding and raw signal. Table 5.2 highlights the capability of *SeismicAE* to produce an embedding which retains valuable information while not getting distracted by unnecessary ones. Two qualitative examples are in Figure 5.4, namely the best and the worst, based on our evaluation in Table 5.2.

### 5.3.2 Foreshock - Aftershock classification

| Classes | Training 70%        |                     | Training 2%   |                     |
|---------|---------------------|---------------------|---------------|---------------------|
|         | 7 CNN               | enc+1 CNN           | 7 CNN         | enc+1 CNN           |
| 2       | <b>99.19 ± 0.12</b> | 96.85 ± 1.15        | 82.41 ± 15.26 | 96.07 ± 0.26        |
| 4       | 68.21 ± 4.85        | 73.43 ± 3.44        | 47.46 ± 14.63 | <b>73.50 ± 0.98</b> |
| 8       | 44.69 ± 1.76        | <b>53.86 ± 3.19</b> | 19.82 ± 7.87  | 48.27 ± 1.66        |
| 9       | 57.94 ± 4.42        | <b>61.80 ± 1.53</b> | 20.33 ± 9.40  | 50.17 ± 1.88        |

**Table 5.3:** Performance results (ratio of correctly classified events to total events) for the testing phase of the *DL* model trained to distinguish foreshock, aftershock, and *Time To Failure (TTF)*. This is to compare the models trained on 70% of the dataset (5076 samples) vs the few shot case, i.e. 2% of the dataset (128 samples). Comparing the model from that is a 7 layers *CNN* (7*CNN*) vs the model made of the embedding concatenated with a 1 layer *CNN* (Enc+1*CNN*). Our results are averaged over 3 runs in the standard case (training on 70% of the dataset), and averaged over 10 runs in the case of few shot training.



**Figure 5.5:** Models used for our downstream tasks, combining the embedding coming from the autoencoder, concatenated with a final set of layers.

This downstream task builds upon the approach presented in [89], where *DL* is used to analyze raw seismic waveforms as probing signals to track the evolution of fault properties during the 2016 Central Italy seismic cycle. Here, we extend that work by showing that acoustic pre-training enables effective few-shot learning, a setting in which the original model from [89] fails to generalize.

We are interested in few-shot learning because, typically, a mainshock is preceded by very few foreshocks, leading to a highly unbalanced dataset. However, this study case differs from that scenario.

We focus here only on the case of NRCA station for classification in foreshocks (2 months before the Norcia *EQ*) and aftershocks (2 months after the Norcia *EQ*), i.e. binary classification; or *TTF* classification, dividing the dataset in 4-8-9 classes based on the time before and after the mainshock. We use the embedding coming from *SeismicAE* fine-tuning on *INSTANCE* + *STEAD* datasets to encode the *EQs* in the catalog, coming from 4 months during the seismic sequence (September to December 2016). We use this embedding in input to a *CNN* layer and the classification layer, as displayed in Figure 5.5a.

We notice that in the training set of *SeismicAE* there are a negligible number of events that are the same as in our Norcia catalog, since *INSTANCE* Dataset also contains seismic traces from the central Italy seismic sequence in 2016. This number is around  $\sim 1\%$  of the recordings used during the training, so we decided to keep them.

The 70% of the dataset is used for training, following a temporal split strategy as described in [89]. Performance metrics (mean  $\pm$  standard deviation over 3 runs) are reported in Table 5.3.

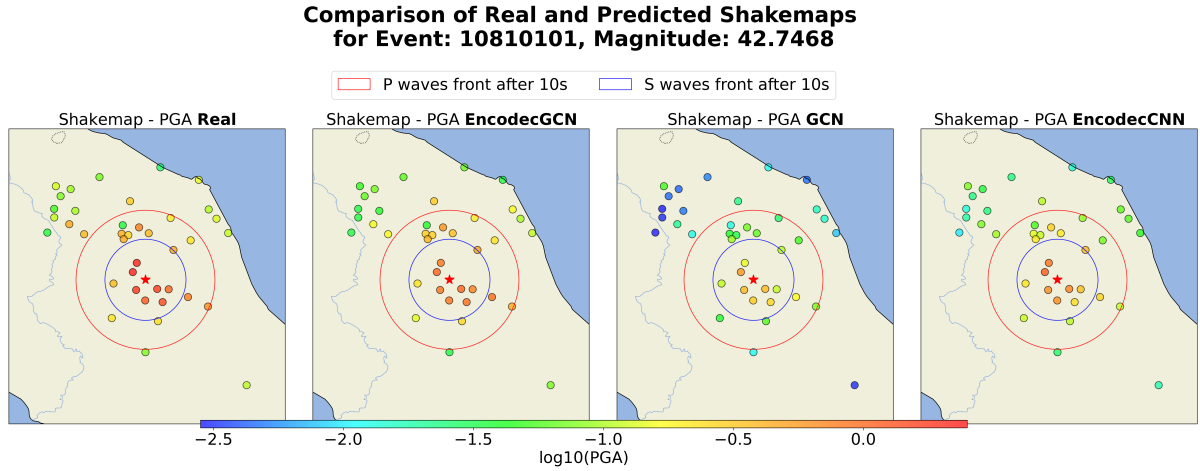
We also evaluate the model in a few shot framework, using the 2 % of the data in training (128 training examples, having 5076 training examples in the standard case). We show that using the encoded version of the traces, the model can learn also in the case of a small dataset available for training, as reported in Table 5.3. In this case, the split in train-val-test is random because of a small amount of data, which has to be representative enough. The small number of data, is also responsible for a big variability, that's why in this case are mean  $\pm$  std of 10 runs. In all the cases, the use of embedding improves the results considerably. In the case of 2 classes, the performance is saturated, and our model made by encoder + 1 *CNN* works as well as the 7 *CNN* model. The

improvement is even more significant in the case of few-shot learning, where the pretraining is essential, especially in the case of 8 and 9 classes.

### 5.3.3 Ground motion prediction

| Model   | Training 65% |              |              |              |              | Training 8%  |              |              |              |              |
|---------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|         | PGA          | PGV          | PSA03        | PSA1         | PSA3         | PGA          | PGV          | PSA03        | PSA1         | PSA3         |
| GCN     | 0.20         | 0.21         | 0.19         | 0.20         | 0.21         | 0.331        | 0.311        | 0.328        | 0.347        | 0.389        |
| Enc+CNN | <b>0.112</b> | <b>0.099</b> | <b>0.105</b> | <b>0.094</b> | <b>0.107</b> | 0.247        | 0.218        | 0.239        | 0.236        | 0.266        |
| Enc+GCN | 0.126        | 0.103        | 0.115        | 0.099        | 0.109        | <b>0.244</b> | <b>0.209</b> | <b>0.233</b> | <b>0.226</b> | <b>0.257</b> |

**Table 5.4:** Results measured in MSE. Our results are averaged over 10 runs. STD is around  $10^{-2}$ . In the standard case we have 586 examples in training, while in the few shot case we have 80 examples in training. *Graph Convolutional Network (GCN)* results refer to the original work [17]. Enc+*CNN* (or *GCN*) is made of the embedding concatenated with a *CNN* (or *GCN*).



**Figure 5.6:** Peak Ground Acceleration (PGA) values across various stations in the Central Italy (CI) network. The plots compare real PGA values and those predicted by different models: **Left:** Real PGA values. **Center Left:** Predictions from the *GCN* model enhanced with our *SeismicAE* for encoding the traces. **Center Right Left:** Predictions from the original *GCN* model. **Right:** Predictions from the *CNN* model combined with *SeismicAE* for trace encoding. The circles represent the wave fronts after 10 seconds for the P wave (red) and the S wave (blue). If a trace contains the arrival of the S wave, it implicitly means that it also contains the P wave within 10 seconds.

*EQ* early warning systems are a hot topic in research because they aim to provide a (near) real-time information on events and their associated impact, i.e. detect *EQs* in a few seconds, and warn surrounding location about the ground motion at selected target points. Rapid assessment of *EQ* generated ground motions is fundamental to provide information crucial to disaster response and public information.

This work builds upon the research conducted by [17, 77]. Their study aims to predict the ground shaking intensity at a predefined set of seismic station, from multivariate time series sensor seismic data. Data come from the Central Italy seismic cycle in 2016, the task involves predicting peak ground motion values based on the seismic event recorded by all the seismic stations in the region (for a total of 39 stations). In this context,  $L$  represents a symmetrically normalized Laplacian

matrix  $L \in \mathbb{R}^{N \times N}$  where  $N$  is the number of graph nodes. The node feature matrix  $Z \in \mathbb{R}^{2 \times N}$  contains the latitude and longitude of each node. The input time series  $X \in \mathbb{R}^{E \times N \times T \times C}$  comprises  $E$  EQs,  $N$  stations,  $T$  the time series length, and  $C$  channels. The objective is to predict  $Y \in \mathbb{R}^{5 \times N}$  representing five target parameters for ground motion of each event  $E$ : PGV (Peak Ground Velocity), PGA (Peak Ground Acceleration), SA 1 (Spectral Acceleration at 1 second), SA 0.3 (Spectral Acceleration at 0.3 second), and SA 3 (Spectral Acceleration at 3 second) for each node. Details about the meaning of these seismological quantities can be retrieved in [77]. These measures are used in seismic engineering to design and evaluate the *EQ* resistance of structures (how structures will behave during an *EQ*), ensuring safety and minimizing damage. The inputs are waveforms up until only a few seconds after origin time when only a few nearby stations have recorded the *EQ*, informing more distant stations; the complexity of the task increases as the input length  $T$  is progressively reduced.

In this context, we decided to test the efficacy of our *SeismicAE* model. By encoding the inputs, we aimed to determine whether this approach could improve the model’s overall performance. For this downstream task, we used the model fine-tuned on *STEAD*, since the dataset employed by the authors is a subset of *INSTANCE*. To ensure fairness, we had to exclude *INSTANCE* from the main training set, as including it would not have been fair since the pretrained model have learned in training phase how to optimally encode the seismic waveforms the authors provide in the catalog [77]. We test two variation of the model in this case, displayed in Figure 5.5b. One is made by the embedding from the fine-tuned model, plus one *CNN* layer and the regression (Linear) layer. The second is made again by the embedding, plus two *GCN* layers and the regression (Linear) layer. The results are in Table 5.4. We can improve the performance of the authors [17] by a big range and there is not a big difference in performance between our variations of the models. This means that encoding the signal is what really makes the difference. Also in this case we perform the training using both the same train-val-test split as the original work, and developing a few shot training. Even if for this experiment also the original training has not much data-samples, the results are in line with expectation, having that the results using the embedding from *SeismicAE* can always improve the performances of the original model from [17].

This is also visible in the qualitative example in Figure 5.6 where the real PGA values (left) serve as the benchmark for evaluating the accuracy of the models. The original *GCN* model (center right) is effective, but shows some deviations from the real values. Our *CNN* model with encoded signal in input (right) appears to closely match the real PGA values, suggesting that encoding signal significantly improves the model’s predictive capabilities. The *GCN* with encoded signal in input (center left) also demonstrates improved performance over the original *GCN*. These results highlight the importance of advanced encoding techniques in improving seismic prediction models.

## 5.4 Discussion

In this Chapter we develop *SeismicAE*, an autoencoder model that, starting from a model trained with audio traces, is capable of creating robust representation of a seismic waveforms. Through appropriate fine-tuning, *SeismicAE* demonstrates strong performance in trace reconstruction and shows significant improvements, with minor training, in downstream tasks such as fault state classification and ground motion prediction, even in few-shot learning scenarios.

The success of *SeismicAE* in adapting audio processing techniques to seismic data represents a significant step forward in applying *ML* to geoscience. By retaining essential features of seismic waveforms while reducing dimensionality, our model offers a powerful tool for various seismological applications, potentially serving as a *FM* for *EQ* analysis.

We focused on three primary applications of our model: (a) achieving good trace reconstruction both with and without the quantizer, (b) improving both the classification of the Norcia fault's state, and (c) the regression of the ground motion during the 2016 seismic sequence in Central Italy (also when having few shots in training).

Using our embedded representation of the seismic traces, we depict here other future impact tasks. Magnitude estimation (regression) is challenging because of our normalization approach, which brings all the seismic events in the range  $[-1,1]$ . One trick could be to include the absolute amplitude as a feature, since it is the most important information for magnitude estimation.

Another interesting goal is noise vs *EQs* classification or *EQs* vs other sources classification that can be applied as a detection method in real time. The latter requires knowledge about other sources (e.g. quarry blasts, explosions, human activities, volcanic sources, ...), which is not present in the datasets used for our fine-tuning. However, this can be easily trained with an appropriate dataset.

Tasks related to specific time sections of the traces are critical since the embedded representation doesn't explicitly keep time information, which is then reconstructed in the decoder. One example is P and S wave picking. Picking these waves is a difficult task performed by seismologists or automated *ML* algorithms [185]. Even though we have shown in 5.3.1 that the P and S information is preserved, it is crucial to detect of the arrival times with extreme precision. Another example is first motion polarity. After identifying P and S wave arrivals, seismologists determine the direction of motion of the first arrival (i.e., positive or negative). This information allows them to identify the source mechanism (orientation of the source fault and its kinematics). This can also be done with *ML* algorithms too [136].

These capabilities could contribute to more effective early warning systems and better understanding of seismic processes.

As *ML* continues to revolutionize geoscience, models like *SeismicAE* demonstrate the power of interdisciplinary approaches. By bridging the gap between audio processing and seismology, which share similarities and differences, we've created a versatile tool as a general encoding procedure for seismology. This work opens new possibilities in the field of *FMs*, specifically developed for *EQ* study, potentially leading to breakthroughs in our ability to monitor, understand, and predict seismic events.

## Chapter 6

# Conclusion

### Conclusions

In this thesis, we discussed *DL* crossroads with seismology and proposed techniques that improve data quality, accelerate inference, and improve predictive performance throughout the seismic monitoring pipeline.

The emergence of *CDiffSD* provides a significant advancement in data processing: by learning non-Gaussian degradation processes from real station noise, our cold diffusion denoiser retains onsets and generates a higher cross-correlation and signal-to-noise ratio, resulting in more trustworthy picking and downstream processing.

We proposed also a masked *GCN* for fast shaking prediction in Italy with 10 s three-component (Three-Component (seismic waveform) (3C)) data at a national scale. By applying masking method, the network is able to target a multi-dimensional spatial property of the growing wavefront while producing continuous error reduction for the three main intensity measures (Peak Ground Acceleration (PGA), Peak Ground Velocity (PGV), Spectral Acceleration (SA)) and reducing bias with respect to the country-specific ground-motion model.

To improve latencies further in the fields of operations, we also designed LLM4GEO, a domain-specific *Large Language Model (LLM)* that estimates epicenter, hypocenter, origin time, and magnitude in seconds after initial detection. On INSTANCE trained and evaluated LLM4GEO reduces early-stage location errors and reduces the number of picks, which in turn points back to tangible gains on *Earthquake Early Warning (EEW)* and rapid situational awareness.

In addition to these features, SEISMICAE communicates our representation learning from audio domain to seismic waveform input to enable compact embeddings which result in better few-shot classification and errors reduction in ground-motion regression and ultimately leads to a reusable *FM* for seismology.

Collectively, these contributions show how *DL* from end-to-end can improve the quality and timely quality of seismic information, from denoising through shaking prediction.

In addition to a single model, the project develops a prototype *AI agent* that integrates our masked GCN, LLM4GEO, and PHASENET into a collaborative, pipeline that achieves representation learning, rapid source characterization, and near real time ground-motion estimation. This modular platform could be compatible with other monitoring solutions as well, offering results that can be used to evaluate seismic risk in actual operations.

## 6.1 Final Remarks

Taken together, our findings show that AI offers concrete benefits for this field, improving data quality, accelerating source characterization, and reducing predictive error across operational datasets. At the same time, because seismic monitoring and early warning have direct safety implications, AI should be integrated as a complement to, not a replacement for classical, physics-based methods, traditional signal processing, and established operational procedures. Promising directions include physics-based neural networks [131, 132, 181], and AI agents tightly integrated with classical physical models within control systems that enforce hard physical constraints. Embedding these models inside conventional pipelines (for denoising, rapid ground-motion inference, and early parameter estimation) can enhance robustness through cross-checks, calibration, and fail-safes, while preserving transparency and expert oversight. In this hybrid paradigm, AI acts as a force multiplier: it strengthens the existing toolkit and improves responsiveness without compromising reliability. An open question that remains from this work is strategic: For seismological tasks, is it preferable to develop a single, general “foundation” model Chapter 5 capable of addressing multiple tasks, or to rely on expert Chapter 7, task-specific models trained for each objective and calibrated to specific domains and geographic regions? Answering this will likely depend on the availability of data, the transferability between tectonic settings, the interpretability needs, and the tolerance to operational risk.

## 6.2 Code and Data Availability

### Chapter 2 – Seismic Denoising with CDiffSD

- Dataset: [doi.org/10.5281/zenodo.10978419](https://doi.org/10.5281/zenodo.10978419) [157]
- Code: <https://github.com/DanieleTrappolini/CDiffSD>

### Chapter 3 – Masked GCN for Ground Motion Prediction

- Code: <https://github.com/DanieleTrappolini/MaskedGCN>
- Dataset: INSTANCEDataset, maintained by INGV

### Chapter 4 – LLM4Geoscience: Earthquake Source Estimation with LLMs

- Code: <https://github.com/AuroraBassani/LLM4Geoscience-1>
- Dataset: INSTANCEDataset



## Chapter 5 – Seismic AE (Acoustic Emission)

- Code and data will be made available upon publication of the corresponding paper.

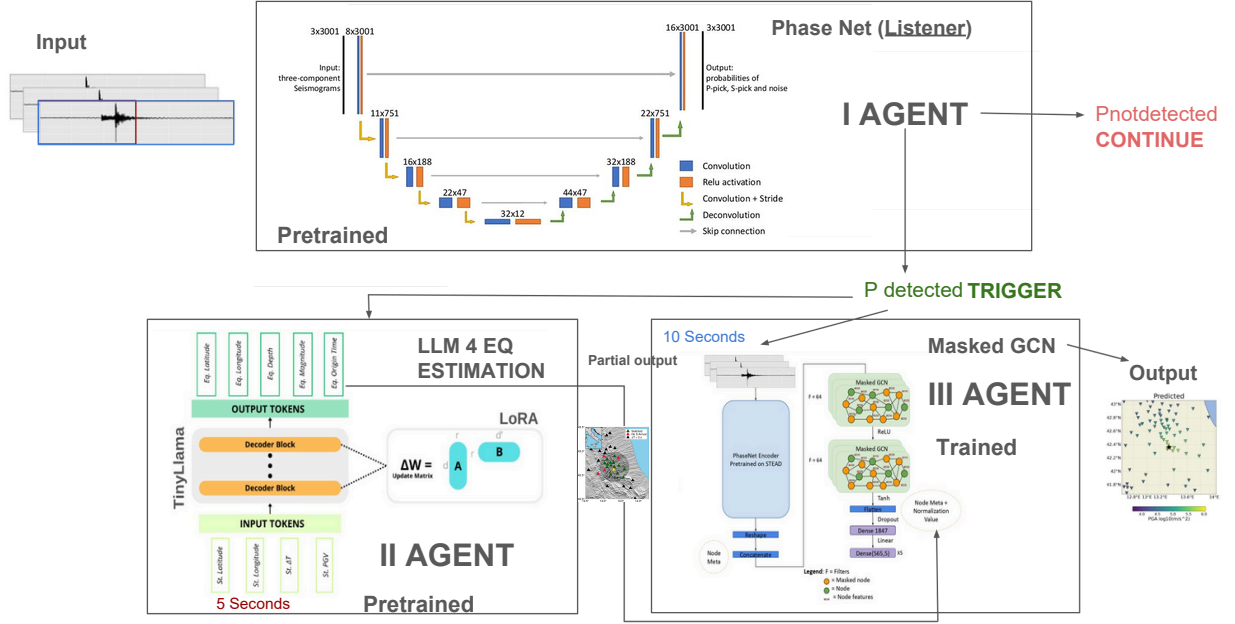
## Chapter 7

# Future Works

This thesis has explored many novel prospects at the convergence of seismology and AI. One of the most exciting future opportunities lies in the development of a modular agent-based pipeline for Earthquake Early Warning (EEW) that embeds advanced models of EEW into an organized, scalable, and extensible pipeline. Central to this system is the training of several expert models Fig. 7.1, each optimized for certain tasks, phase picking, magnitude, location estimation, and ground motion prediction, without being based on a single monolithic architecture. At present, communication between the modules is managed through a manual orchestration layer, but in the future, an intelligent orchestration will be implemented, for example, with the aid of a framework such as Rivet AI, LangGraph [169], LangChain [102], or Postman Flows [84]. Three models that have been established in this framework are:

- **PhaseNet** [185], used as a real-time phase picker and lightweight event trigger: it estimates per-station  $P/S$  arrival probabilities and picks to (i) gate the downstream processing by verifying the presence of an event, and (ii) pass phase-derived metadata to the other modules (LLM4Geoscience and the Masked GCN).
- **LLM4Geoscience** (Chapter 4), which is a fine-tuned decoder-only large language model for predicting earthquake source characteristics (magnitude, latitude, longitude, depth) using early features determined from the waveform;
- **Masked Graph Convolutional Network (GCN)** (Chapter 3), applied to estimate shaking parameters (PGA, PGV, SA), distributed across a network of stations; integrated spatial and edge information to capture coastal and low-coverage areas.

Trained and validated on the INSTANCE dataset, the integrated pipeline enhances the Masked GCN's performance under complex conditions (e.g., offshore epicenters and sparse networks), yielding more reliable shaking estimates. The architecture was formulated for modular, easy-to-interpret, large-scale, and scalable capabilities, which were in line with scientific principles and could be applied practically. Our ultimate desire is to extend this system to be an accurate device in the industry for actual risk management and emergency response with specific reference to the INGV's Seismic Room. With its modular setup and fast inference abilities, it could achieve almost real-time earthquake impact estimation, assisting humans in a disaster.



**Figure 7.1:** Agentic EEW architecture (*Seismo Agents*). *PhaseNet* listens in for P-wave detection and activates *LLM4Geoscience* for speedy prediction of source parameters (latitude, longitude, depth, magnitude). Subsequently, the *Masked GCN* can predict shaking metrics (PGA/PGV/SA) throughout the station network. The pipeline works on discrete time windows (e.g., 5–10 s) and utilizes pretrained and task-specific models to provide modular orchestration and near real-time risk management support.

## 7.1 Future development directions

A few forward-looking improvements are predicted:

- **Generalization to other regions**, experiment and retrain the expert models on international data (e.g., Japan, California) in order to adapt to different seismic regimes;
- **Real-time streaming and deployment**, connecting the pipeline with live seismic data to assess latency, fault tolerance, and the potential to be integrated with alert systems;
- **Uncertainty estimation**, including estimate of confidence intervals and probabilistic values for epicentral parameters and ground motion predictions;
- **Multi-modal data integration**, GNSS, crowd data, and other environmental sensors are integrated to enhance the prediction performance in regions with poor instrumentation;
- **Phase Association (ongoing)**, a fourth agent plugged in immediately after *PhaseNet* to associate picks across stations and disambiguate overlapping events; this stage is designed to overcome the current filter on larger magnitudes ( $M \geq 3$ ) and enable operation at lower magnitudes by clustering and linking phase arrivals into coherent event hypotheses. We plan to evaluate/deploy association models such as *GaMMA* [186] and *PyOcto* [117].
- **Advanced orchestration layer**, a replacement of existing manual orchestration by a dynamic reactive agent orchestration, based on modern frameworks (Rivet, LangGraph) in order to enable inter-agent communication, task delegation, and fault recovery.

Eventually, this effort will produce a modular-operational-ready EEW pipeline that combines specialized expert models with an elastic orchestration layer. The aim is to create a scientifically grounded and workable framework for increasing the effectiveness and agility of seismic monitoring tools, facilitating early warning systems, and deploying seismic hazard mitigation techniques.

# Bibliography

- [1] R. E. Abercrombie. Earthquake source scaling relationships from -1 to 5 ml using seismograms recorded at 2.5-km depth. *Journal of Geophysical Research: Solid Earth*, 100(B12):24015–24036, 1995.
- [2] R. E. Abercrombie, A. Baltay, S. Chu, T. Taira, D. Bindi, O. S. Boyd, X. Chen, E. S. Cochran, E. Devin, D. Dreger, et al. Overview of the scec/usgs community stress drop validation study using the 2019 ridgecrest earthquake sequence. *Bulletin of the Seismological Society of America*, 115(3):734–759, 2025.
- [3] F. Aden-Antoniów, W. B. Frank, and L. Seydoux. An adaptable random forest model for the declustering of earthquake catalogs. *Journal of Geophysical Research: Solid Earth*, 127:e2021JB023254, 2022.
- [4] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [5] H. Alcik, O. Ozel, N. Apaydin, and M. Erdik. A study on warning algorithms for istanbul earthquake early warning system. *Geophysical Research Letters*, 36(5), 2009.
- [6] R. M. Allen, H. Brown, M. Hellweg, O. Khainovski, P. Lombard, and D. Neuhauser. Real-time earthquake detection and hazard assessment by elarms across california. *Geophysical Research Letters*, 36(5), 2009.
- [7] R. M. Allen, P. Gasparini, O. Kamigaichi, and M. Bose. The status of earthquake early warning around the world: An introductory overview. *Seismological Research Letters*, 80(5):682–693, 2009.
- [8] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.
- [9] L. A. Atik, N. Abrahamson, J. J. Bommer, F. Scherbaum, F. Cotton, and N. Kuehn. The variability of ground-motion prediction models and its components. *Seismological Research Letters*, 81(5):794–801, 2010.
- [10] E. Bakhturina, V. Lavrukhin, B. Ginsburg, and Y. Zhang. Hi-fi multi-speaker english tts dataset. *arXiv preprint arXiv:2104.01497*, 2021.

- [11] A. Bansal, E. Borgnia, H.-M. Chu, J. S. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, and T. Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022.
- [12] A. Bassani, D. Trappolini, G. Poggiali, E. Tinti, F. Galasso, A. Michelini, and C. Marone. Rapid estimation of earthquake location and magnitude using a large language model. *AGU25*.
- [13] L. K. Bear, G. L. Pavlis, and G. H. Bokelmann. Multi-wavelet analysis of three-component seismic arrays: application to measure effective anisotropy at pinon flats, california. *Bulletin of the Seismological Society of America*, 89(3):693–705, 1999.
- [14] M. Bekara and M. V. der Baan. Random and coherent noise attenuation by empirical mode decomposition. *Geophysics*, vol. 74, no. 5, pp. V89–V98, 2009, 2009.
- [15] X. Bie, S. Leglaive, X. Alameda-Pineda, and L. Girin. Unsupervised speech enhancement using dynamical variational autoencoders. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2993–3007, 2022.
- [16] D. Bindi, F. Pacor, L. Luzi, R. Puglia, M. Massa, G. Ameri, and R. Paolucci. Ground motion prediction equations derived from the italian strong motion database. *Bulletin of Earthquake Engineering*, 9:1899–1920, 2011.
- [17] S. Bloemheuvel, J. van den Hoogen, D. Jozinović, A. Michelini, and M. Atzmueller. Graph neural networks for multivariate time series regression with application to seismic data. *International Journal of Data Science and Analytics*, 16(3):317–332, 2023.
- [18] G. Bohm, A. Affatato, G. Brancatelli, E. Forlin, F. Meneghini, L. Baradello, P. Corubolo, et al. 3d p-wave and s-wave velocity model for the characterisation of the subsurface beneath the village of arquata del tronto (central italy). *BULLETIN OF GEOPHYSICS AND OCEANOGRAPHY*, 2024.
- [19] D. M. Boore and W. B. Joyner. Site amplifications for generic rock sites. *Bulletin of the Seismological Society of America*, 87(2):327–341, 1997.
- [20] M. Böse, E. Hauksson, K. Solanki, H. Kanamori, Y.-M. Wu, and T. Heaton. A new trigger criterion for improved real-time performance of onsite earthquake early warning in southern california. *Bulletin of the Seismological Society of America*, 99(2A):897–905, 2009.
- [21] M. Böse, T. Heaton, and E. Hauksson. Rapid estimation of earthquake source and ground-motion parameters for earthquake early warning using data from a single three-component broadband or strong-motion sensor. *Bulletin of the Seismological Society of America*, 102(2):738–750, 2012.
- [22] M. Böse, C. Ionescu, and F. Wenzel. Earthquake early warning for bucharest, romania: Novel and revised scaling relations. *Geophysical Research Letters*, 34(7), 2007.
- [23] M. Böse, F. Wenzel, and M. Erdik. Preseis: A neural network-based approach to earthquake early warning for finite faults. *Bulletin of the Seismological Society of America*, 98(1):366–382, 2008.

- 
- [24] P. Boué, P. Roux, M. Campillo, and B. de Cacqueray. Double beamforming processing in a seismic prospecting context. *Geophysics*, 78(3):V101–V108, 2013.
  - [25] B. Braeuer and K. Bauer. A new interpretation of seismic tomography in the southern dead sea basin using neural network clustering techniques. *Geophysical Research Letters*, 42:9772–9780, 2015.
  - [26] L. A. Brooks, J. Townend, P. Gerstoft, S. Bannister, and L. Carter. Fundamental and higher-mode rayleigh wave characteristics of ambient seismic noise in new zealand. *Geophysical Research Letters*, 36(23), 2009.
  - [27] G. Cabras, R. Carniel, and J. Wasserman. Signal enhancement with generalized ica applied to mt. etna volcano, italy. *Bollettino di Geofisica Teorica ed Applicata*, 51(1), 2010.
  - [28] R. Cao, S. Abdulatif, and B. Yang. Cmgan: Conformer-based metric gan for speech enhancement. *arXiv preprint arXiv:2203.15149*, 2022.
  - [29] S. Cao and X. Chen. The second-generation wavelet transform and its application in denoising of seismic data. *Applied geophysics*, 2:70–74, 2005.
  - [30] M. Chakraborty, D. Fenner, W. Li, J. Faber, K. Zhou, G. Rümpker, H. Stoecker, and N. Srivastava. Creime—a convolutional recurrent model for earthquake identification and magnitude estimation. *Journal of Geophysical Research: Solid Earth*, 127(7):e2022JB024595, 2022.
  - [31] D.-Y. Chen, Y.-M. Wu, and T.-L. Chin. An empirical evolutionary magnitude estimation for early warning of earthquakes. *Journal of Asian Earth Sciences*, 135:190–197, 2017.
  - [32] Y. Chen and J. Ma. Random noise attenuation by f-x empiricalmode decomposition predictive filtering. *Geophysics*, vol. 79, no. 3, pp. V81–V91, 2014, 2014.
  - [33] Z. Chen, Z. Wang, S. Wu, Y. Wang, and J. Gao. Maginfonet: Magnitude estimation using seismic information augmentation and graph transformer. *Earth and Space Science*, 9(12):e2022EA002580, 2022.
  - [34] S. Colombelli, G. Festa, and A. Zollo. Early rupture signals predict the final earthquake size. *Geophysical Journal International*, 223(1):692–706, 2020.
  - [35] P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.
  - [36] M. Corradini, I. W. McBrearty, D. T. Trugman, C. Satriano, P. A. Johnson, and P. Bernard. Investigating the influence of earthquake source complexity on back-projection images using convolutional neural networks. *Geophysical Journal International*, 229:1824–1839, 2022.
  - [37] P. Danecek, S. Pintore, S. Mazza, A. Mandiello, M. Fares, I. Carluccio, E. Della Bina, D. Franceschi, M. Moretti, V. Lauciani, et al. The italian node of the european integrated data archive. *Seismological Research Letters*, 92(3):1726–1737, 2021.
  - [38] N. Deichmann and M. Garcia-Fernandez. Rupture geometry from high-precision relative hypocentre locations of microearthquake clusters. *Geophysical Journal International*, 110(3):501–517, Sept. 1992.
-

- [39] B. Derras, P. Y. Bard, and F. Cotton. Towards fully data driven ground-motion prediction models for europe. *Bulletin of Earthquake Engineering*, 12(1):495–516, 2014.
- [40] C. Donahue, B. Li, and R. Prabhavalkar. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5024–5028. IEEE, 2018.
- [41] R. Durall, A. Ghanim, M. R. Fernandez, N. Ettrich, and J. Keuper. Deep diffusion models for seismic processing. *Computers & Geosciences*, 177:105377, 2023.
- [42] S. N. Dybing, W. L. Yeck, H. M. Cole, and D. Melgar. Rapid estimation of single-station earthquake magnitudes with machine learning on a global scale. *Bulletin of the Seismological Society of America*, 114(3):1523–1538, 2024.
- [43] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi. High fidelity neural audio compression, 2022.
- [44] Earthworm Community. Earthworm central: Seismology software and resources, 2025. Accessed: 2025-01-23.
- [45] W. Ellsworth and G. Beroza. Seismic evidence for an earthquake nucleation phase. *Science*, 268(5212):851–855, 1995.
- [46] J. M. Espinosa Aranda, A. Jimenez, G. Ibarrola, F. Alcantar, A. Aguilar, M. Inostroza, and S. Maldonado. Mexico city seismic alert system. *Seismological Research Letters*, 66:42–53, 1995.
- [47] A. M. Esposito, F. Giudicepietro, S. Scarpetta, L. D’Auria, M. Marinaro, and M. Martini. Automatic discrimination among landslide, explosion-quake, and microtremor seismic signals at stromboli volcano using neural networks. *Bulletin of the Seismological Society of America*, 96:1230–1240, 2006.
- [48] H. Fang, G. Carbajal, S. Wermter, and T. Gerkmann. Variational autoencoder for speech enhancement with a noise-aware encoder. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 676–680. IEEE, 2021.
- [49] S. F. Fornasari, V. Pazzi, and G. Costa. A machine-learning approach for the reconstruction of ground-shaking fields in real time. *Bulletin of the Seismological Society of America*, 112(5):2642–2652, 2022.
- [50] M.-J. Frémont and S. D. Malone. High precision relative locations of earthquakes at Mount St. Helens, Washington. *Journal of Geophysical Research: Solid Earth*, 92(B10):10223–10236, 1987. \_eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/JB092iB10p10223>.
- [51] S. Gaci. The use of wavelet-based denoising techniques to enhance the first-arrival picking on seismic traces. *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4558–4563, Aug. 2014, 2014.
- [52] A. H. Gandomi, A. H. Alavi, M. Mousavi, and S. M. Tabatabaei. A hybrid computational approach to derive new ground-motion prediction equations. *Engineering Applications of Artificial Intelligence*, 24(4):717–732, 2011.



- 
- [53] P. Gasperini, B. Lolli, and G. Vannucci. Empirical calibration of local magnitude data sets versus moment magnitude in Italy. *Bulletin of the Seismological Society of America*, 103(4):2227–2246, 2013.
  - [54] L. Geiger. Probability method for the determination of earthquake epicentres from the arrival time only. *Bull. St. Louis Univ.*, 8:60, 1912.
  - [55] S. J. Gibbons, F. Ringdal, and T. K  rna. Detection and characterization of seismic phases using continuous spectral estimation on incoherent and partially coherent arrays. *Geophysical Journal International*, 172(1):405–421, 2008.
  - [56] D. Gillard, A. M. Rubin, and P. Okubo. Highly concentrated seismicity caused by deformation of Kilauea’s deep magma system. *Nature*, 384(6607):343–346, Nov. 1996.
  - [57] L.-W. Gong, H. Zhang, S. Chen, and L.-J. Chen. Three-dimensional modeling of the Xichang crust in Sichuan, China by machine learning. *Applied Sciences*, 12(6):2955, 2021.
  - [58] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.
  - [59] J. Han and M. van der Baan. Microseismic and seismic denoising via ensemble empirical mode decomposition and adaptive thresholding. *Geophysics*, vol. 80, no. 6, pp. KS69–KS80, . doi: 10.1190/geo2014-0423.1, 2015.
  - [60] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
  - [61] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
  - [62] G. Hennenfent and F. J. Herrmann. Seismic denoising with nonuniformly sampled curvelets. *Comput. Sci. Eng.*, vol. 8, no. 3, p. 16, May 2006, 2006.
  - [63] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
  - [64] N.-C. Hsiao, Y.-M. Wu, T.-C. Shin, L. Zhao, and T.-L. Teng. Development of earthquake early warning system in Taiwan. *Geophysical Research Letters*, 36(5), 2009.
  - [65] T.-Y. Hsu and C.-W. Huang. Onsite early prediction of PGA using CNN with multi-scale and multi-domain P-waves as input. *Frontiers in Earth Science*, 9:626908, 2021.
  - [66] T.-Y. Hsu, S.-K. Huang, Y.-W. Chang, C.-H. Kuo, C.-M. Lin, T.-M. Chang, K.-L. Wen, and C.-H. Loh. Rapid on-site peak ground acceleration estimation based on support vector regression and P-wave features in Taiwan. *Soil Dynamics and Earthquake Engineering*, 49:210–217, 2013.
  - [67] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
-

- 
- [68] J. Hu, Y. Ding, H. Zhang, C. Jin, and Z. Wang. A real-time seismic intensity prediction framework based on interpretable ensemble learning. *Seismological Research Letters*, 94(3):1579–1602, 2023.
  - [69] A. G. Iaccarino, A. Cristofaro, M. Picozzi, D. Spallarossa, and D. Scafidi. Real-time prediction of distance and pga from p-wave features using gradient boosting regressor for on-site earthquake early warning applications. *Geophysical Journal International*, 236(1):675–687, 2024.
  - [70] A. Ito. High Resolution Relative Hypocenters of Similar Earthquakes by Cross-Spectral Analysis Method. *Journal of Physics of the Earth*, 33(4):279–294, 1985.
  - [71] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.
  - [72] X. Jin, H. Zhang, J. Li, Y. Wei, and Q. Ma. Earthquake magnitude estimation using the  $\tau$  c and p d method for earthquake early warning systems. *Earthquake Science*, 26:23–31, 2013.
  - [73] C. Johnson, C. Hulbert, B. Rouet-Leduc, and P. Johnson. Learning the low frequency earthquake daily intensity on the central san andreas fault, 11 2020.
  - [74] C. W. Johnson and P. A. Johnson. Seismic features predict ground motions during repeating caldera collapse sequence. *Geophysical Research Letters*, 51(11):e2024GL108288, 2024.
  - [75] C. W. Johnson, D. Kilb, A. Baltay, and F. Vernon. Peak ground velocity spatial variability revealed by dense seismic array in southern california. *Journal of Geophysical Research: Solid Earth*, 125(6):e2019JB019157, 2020. e2019JB019157 2019JB019157.
  - [76] C. W. Johnson, H. Meng, F. Vernon, and Y. Ben-Zion. Characteristics of ground motion generated by wind interaction with trees, structures, and other surface obstacles. *Journal of Geophysical Research: Solid Earth*, 124(8):8519–8539, 2019.
  - [77] D. Jozinović, A. Lomax, I. Štajduhar, and A. Michelini. Rapid prediction of earthquake ground shaking intensity using raw waveform data and a convolutional neural network. *Geophysical Journal International*, 222(2):1379–1389, 2020.
  - [78] D. Jozinović, A. Lomax, I. Štajduhar, and A. Michelini. Transfer learning: Improving neural network based prediction of earthquake ground shaking for an area with insufficient training data. *Geophysical Journal International*, 229(1):704–718, 2022.
  - [79] O. Kamigaichi, M. Saito, K. Doi, T. Matsumori, S. Tsukada, K. Takeda, T. Shimoyama, K. Nakamura, M. Kiyomoto, and Y. Watanabe. Earthquake early warning in japan: Warning the general public and future prospects. *Seismological Research Letters*, 80(5):717–726, 2009.
  - [80] H. Kanamori. Real-time seismology and earthquake damage mitigation. *Annu. Rev. Earth Planet. Sci.*, 33(1):195–214, 2005.
  - [81] E. Karasözen and B. Karasözen. Earthquake location methods. *GEM-International Journal on Geomathematics*, 11(1):13, 2020.
-

- [82] S. Karimpouli and P. Tahmasebi. Physics informed machine learning: Seismic wave equation. *Geoscience Frontiers*, 11(6):1993–2001, 2020.
- [83] H. Y. Kim, J. W. Yoon, S. J. Cheon, W. H. Kang, and N. S. Kim. A multi-resolution approach to gan-based speech enhancement. *Applied Sciences*, 11(2):721, 2021.
- [84] R. Krosnick. Postman flows: A visual programming tool for building api-powered apps and workflows. In *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 356–358. IEEE, 2024.
- [85] S. Lacaze, S. Philit, F. Pauget, and T. Wilson. Comprehensive method for a multi-2d seismic interpretation. In *SEG Technical Program Expanded Abstracts 2018*, pages 1524–1528. Society of Exploration Geophysicists, 2018.
- [86] M. Lancieri and A. Zollo. A bayesian approach to the real-time estimation of magnitude from the early p and s wave displacement peaks. *Journal of Geophysical Research: Solid Earth*, 113(B12), 2008.
- [87] R. B. Langley. The utm grid system. *GPS world*, 9(2):46–50, 1998.
- [88] L. Laurenti, C. W. Johnson, D. Trappolini, E. Tinti, F. Galasso, and C. Marone. Testing audio compression autoencoders for seismology: moving toward foundation models. *Authorea Preprints*, 2025.
- [89] L. Laurenti, G. Paoletti, E. Tinti, et al. Probing the evolution of fault properties during the seismic cycle with deep learning. *Research Square*, June 2024. preprint (Version 1).
- [90] L. Laurenti, E. Tinti, F. Galasso, L. Franco, and C. Marone. Deep learning for laboratory earthquake prediction and autoregressive forecasting of fault zone stress. *Earth and Planetary Science Letters*, 598:117825, 2022.
- [91] S. Leglaive, X. Alameda-Pineda, L. Girin, and R. Horaud. A recurrent variational autoencoder for speech enhancement. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 371–375. IEEE, 2020.
- [92] S. Leglaive, L. Girin, and R. Horaud. A variance modeling framework based on variational autoencoders for speech enhancement. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2018.
- [93] R. Li, L. B. Allal, Y. Zi, N. Muennighoff, D. Kocetkov, C. Mou, M. Marone, C. Akiki, J. Li, J. Chim, et al. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*, 2023.
- [94] J. S. Lim. *Two-dimensional signal and image processing*. Prentice-Hall, Inc., 1990.
- [95] T. Liu, J. Münchmeyer, L. Laurenti, C. Marone, M. V. de Hoop, and I. Dokmanić. Seislm: a foundation model for seismic waveforms. *arXiv preprint arXiv:2410.15765*, 2024.
- [96] W. Liu, S. Cao, and Y. Chen. Seismic time-frequency analysis via empirical wavelet transform. *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 28–32, Jan. 2016, 2016.

- [97] Y. Liu, Y. Li, H. Lin, and H. Ma. An amplitude-preserved time–frequency peak filtering based on empirical mode decomposition for seismic random noise reduction. *IEEE Geoscience and Remote Sensing Letters*, 11(5):896–900, 2013.
- [98] A. Lomax, A. Michelini, A. Curtis, R. Meyers, et al. Earthquake location, direct, global-search methods. *Encyclopedia of complexity and systems science*, 5:2449–2473, 2009.
- [99] Y.-J. Lu, Z.-Q. Wang, S. Watanabe, A. Richard, C. Yu, and Y. Tsao. Conditional diffusion probabilistic model for speech enhancement. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7402–7406. IEEE, 2022.
- [100] N. Mansour, M. Marschall, T. May, A. Westermann, and T. Dau. A method for realistic, conversational signal-to-noise ratio estimation. *The Journal of the Acoustical Society of America*, 149(3):1559–1566, 2021.
- [101] L. Margheriti, C. Nostro, O. Cocina, M. Castellano, M. Moretti, V. Lauciani, M. Quintiliani, A. Bono, F. M. Mele, S. Pintore, et al. Seismic surveillance and earthquake monitoring in Italy. *Seismological Research Letters*, 92(3):1659–1671, 2021.
- [102] V. Mavroudis. Langchain. 2024.
- [103] M.-A. Meier, T. Heaton, and J. Clinton. Evidence for universal earthquake rupture initiation behavior. *Geophysical Research Letters*, 43(15):7991–7996, 2016.
- [104] A. Michelini, S. Cianetti, S. Gaviano, C. Giunchi, D. Jozinović, and V. Lauciani. Instance—the Italian seismic dataset for machine learning. *Earth System Science Data*, 13(12):5509–5544, 2021.
- [105] A. Michelini, L. Faenza, G. Lanzano, V. Lauciani, D. Jozinović, R. Puglia, and L. Luzi. The new shakemap in Italy: Progress and advances in the last 10 yr. *Seismological Research Letters*, 91(1):317–333, 2020.
- [106] A. Michelini, L. Margheriti, M. Cattaneo, G. Cecere, G. D’Anna, A. Delladio, M. Moretti, S. Pintore, A. Amato, A. Basili, et al. The Italian national seismic network and the earthquake and tsunami monitoring and surveillance systems. *Advances in Geosciences*, 43:31–38, 2016.
- [107] J. Milne. *Earthquakes and other earth movements*, volume 56. K. Paul, Trench, 1886.
- [108] A. Moni, C. J. Bean, I. Lokmer, and S. Rickard. Source separation on seismic data: Application in a geophysical setting. *IEEE Signal Processing Magazine*, 29(3):16–28, 2012.
- [109] J. Mori and H. Kanamori. Initial rupture of earthquakes in the 1995 ridgecrest, California sequence. *Geophysical Research Letters*, 23(18):2437–2440, 1996.
- [110] S. M. Mousavi and G. C. Beroza. Bayesian-deep-learning estimation of earthquake location from single-station observations. *arXiv preprint arXiv:1912.01144*, 2019.
- [111] S. M. Mousavi and G. C. Beroza. A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*, 47(1):e2019GL085976, 2020.

- [112] S. M. Mousavi and G. C. Beroza. A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*, 47(1):e2019GL085976, 2020.
- [113] S. M. Mousavi and G. C. Beroza. Machine learning in earthquake seismology. *Annual Review of Earth and Planetary Sciences*, 51(1):105–129, 2023.
- [114] S. M. Mousavi and C. A. Langston. Adaptive noise estimation and suppression for improving microseismic event detection. *Appl. Geophys.*, vol. 132, pp. 116–124, Sep. 2016. doi: 10.1016/j.jappgeo.2016.06.008, 2016.
- [115] S. M. Mousavi and C. A. Langston. Automatic noise-removal/signalremoval based on general cross-validation thresholding in synchrosqueezed domain and its application on earthquake data. *Geophysics*, vol. 82, no. 4, pp. V211–V227, 2017. doi: 10.1190/geo20160433.1, 2017.
- [116] S. M. Mousavi, Y. Sheng, W. Zhu, and G. C. Beroza. Stanford earthquake dataset (stead): A global data set of seismic signals for ai. *IEEE Access*, 7:179464–179476, 2019.
- [117] J. Münchmeyer. Pyocto: A high-throughput seismic phase associator. *arXiv preprint arXiv:2310.11157*, 2023.
- [118] J. Münchmeyer, D. Bindi, U. Leser, and F. Tilmann. Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network. *Geophysical Journal International*, 226(2):1086–1104, 2021.
- [119] J. Münchmeyer, D. Bindi, U. Leser, and F. Tilmann. The transformer earthquake alerting model: A new versatile approach to earthquake early warning. *Geophysical Journal International*, 225(1):646–656, 2021.
- [120] Y. Nakamura. On the urgent earthquake detection and alarm system (uredas). In *Proc. of the 9th world conference on earthquake engineering*, volume 7, pages 673–678, 1988.
- [121] R. Neelamani, A. I. Baumstein, D. G. Gillard, M. T. Hadidi, and W. L. Soroka. Coherent and random noise attenuation using the curvelet transform. *The Leading Edge*, 27(2):240–248, 2008.
- [122] A. Novoselov, P. Balazs, and G. Bokelmann. Sedenoss: Separating and denoising seismic signals with dual-path recurrent neural network architecture. *Journal of Geophysical Research: Solid Earth*, 127(3):e2021JB023183, 2022.
- [123] E. L. Olson and R. M. Allen. The deterministic nature of earthquake rupture. *Nature*, 438(7065):212–215, 2005.
- [124] R. Otake, J. Kurima, H. Goto, and S. Sawada. Deep learning model for spatial interpolation of real-time seismic intensity. *Seismological Research Letters*, 91(6):3433–3443, 2020.
- [125] S. Pascual, A. Bonafonte, and J. Serrà. Segan: Speech enhancement generative adversarial network. in *Proc. Interspeech*, 2017, 2017.
- [126] T. Perol, M. Gharbi, and M. Denolle. Convolutional neural network for earthquake detection and location. *Science Advances*, 4(2):e1700578, 2018.

- [127] G. Poupinet, W. L. Ellsworth, and J. Frechet. Monitoring velocity variations in the crust using earthquake doublets: An application to the Calaveras Fault, California. *Journal of Geophysical Research: Solid Earth*, 89(B7):5719–5731, July 1984. Publisher: John Wiley & Sons, Ltd.
- [128] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askeel, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [129] M. Rasht-Behesht, C. Huber, K. Shukla, and G. E. Karniadakis. Physics-informed neural networks (pinns) for wave propagation and full waveform inversions. *Journal of Geophysical Research: Solid Earth*, 127(5):e2021JB023120, 2022.
- [130] H. F. Reid. The california earthquake of april 18, 1906. *Report of the state earthquake investigation commission*, 2:16–18, 1910.
- [131] P. Ren, C. Rao, S. Chen, J.-X. Wang, H. Sun, and Y. Liu. Seismicnet: Physics-informed neural networks for seismic wave modeling in semi-infinite domain. *Computer Physics Communications*, 295:109010, 2024.
- [132] Y. Ren, X. Xu, S. Yang, L. Nie, and Y. Chen. A physics-based neural-network way to perform seismic full waveform inversion. *IEEE Access*, 8:112266–112277, 2020.
- [133] J. Richter, S. Welker, J.-M. Lemerrier, B. Lay, and T. Gerkmann. Speech enhancement and dereverberation with diffusion-based generative models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [134] A. J. Rodgers and W. R. Walter. Comparison of snr for teleseismic p and regional pn waves. *Bulletin of the Seismological Society of America*, 92(6):2501–2516, 2002.
- [135] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.
- [136] Z. E. Ross, M.-A. Meier, and E. Hauksson. P wave arrival picking and first-motion polarity determination with deep learning. *Journal of Geophysical Research: Solid Earth*, 123(6):5120–5129, 2018.
- [137] Z. E. Ross, M.-A. Meier, E. Hauksson, and T. H. Heaton. Generalized seismic phase detection with deep learning. *Bulletin of the Seismological Society of America*, 108(5A):2894–2901, 2018.
- [138] B. Rouet-Leduc, C. Hulbert, and P. A. Johnson. Continuous chatter of the cascadia subduction zone revealed by machine learning. *Nature Geoscience*, 12:75–79, 2019.
- [139] A. M. Rubin, D. Gillard, and J.-L. Got. Streaks of microearthquakes along creeping faults. *Nature*, 400(6745):635–641, Aug. 1999.

- [140] P. Rydelek and S. Horiuchi. Is earthquake rupture deterministic? *Nature*, 442(7100):E5–E6, 2006.
- [141] C. Satriano, L. Elia, C. Martino, M. Lancieri, A. Zollo, and G. Iannaccone. Presto, the earthquake early warning system for southern italy: Concepts, capabilities and future perspectives. *Soil Dynamics and Earthquake Engineering*, 31(2):137–153, 2011.
- [142] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.
- [143] H. Shan, J. Ma, and H. Yang. Comparisons of wavelets, contourlets and curvelets in seismic denoising. *Appl. Geophys.*, vol. 69, no. 2, pp. 103–115, Oct. 2009. doi: 10.1016/j.jappgeo.2009.08.002, 2009.
- [144] X. Si, X. Wu, H. Sheng, J. Zhu, and Z. Li. Seisclip: A seismology foundation model pre-trained by multimodal data for multipurpose seismic feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–13, 2024.
- [145] C. Singh, A. Askari, R. Caruana, and J. Gao. Augmenting interpretable models with large language models during training. *Nature Communications*, 14(1):7913, 2023.
- [146] C. Singh, J. P. Inala, M. Galley, R. Caruana, and J. Gao. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*, 2024.
- [147] C. Siyuan and C. Xiangpeng. The second-generation wavelet transform and its application in denoising of seismic data. *Appl. Geophys.*, vol. 2, no. 2, pp. 70–74, Jun. 2005, 2005.
- [148] D. Soboleva, F. Al-Khateeb, R. Myers, J. R. Steeves, J. Hestness, and N. Dey. Slimpajama: A 627b token cleaned and deduplicated version of redpajama, 2023.
- [149] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- [150] J. Song, J. Zhu, and S. Li. Meanet: Magnitude estimation via physics-based features time series, an attention mechanism, and neural networks. *Geophysics*, 88(1):V33–V43, 2023.
- [151] D. Spallarossa, S. R. Kotha, M. Picozzi, S. Barani, and D. Bindi. On-site earthquake early warning: a partially non-ergodic perspective from the site effects point of view. *Geophysical Journal International*, 216(2):919–934, 2019.
- [152] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7, 2018, Proceedings, Part III 27*, pages 270–279. Springer, 2018.
- [153] G. Tang and J. Ma. Application of total-variation-based curvelet shrinkage for three-dimensional seismic data denoising. *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 1, pp. 103–107, Jan. 2011, 2011.

- 
- [154] Z. Tang, Z. Qiao, X. Hong, Y. Wang, F. A. Dharejo, Y. Zhou, and Y. Du. Data augmentation for graph convolutional network on semi-supervised classification. In *Web and Big Data: 5th International Joint Conference, APWeb-WAIM 2021, Guangzhou, China, August 23–25, 2021, Proceedings, Part II* 5, pages 33–48. Springer, 2021.
  - [155] TinyLlama. Tinyllama-1.1b-chat-v1.0, 2024. Accessed: 2024-10-16.
  - [156] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
  - [157] D. Trappolini. Stead subsample 4 cdiffsd, 2024.
  - [158] D. Trappolini, L. Laurenti, G. Poggiali, E. Tinti, F. Galasso, A. Michelini, and C. Marone. Cold diffusion model for seismic denoising. *Journal of Geophysical Research: Machine Learning and Computation*, 1(2):e2024JH000179, 2024. e2024JH000179 2024JH000179.
  - [159] D. Trappolini, I. Oliveti, L. Faenza, D. Jozinović, and A. Michelini. Masked graph neural network for rapid ground motion prediction in italy. *Seismica*, 4(2), 2025.
  - [160] D. T. Trugman and P. M. Shearer. Strong correlation between stress drop and peak ground acceleration for recent m 1–4 earthquakes in the san francisco bay area. *Bulletin of the Seismological Society of America*, 108:929–945, 2018.
  - [161] G.-A. Tselentis, N. Martakis, P. Paraskevopoulos, A. Lois, and E. Sokos. Strategy for automated analysis of passive microseismic data based on s-transform, otsu’s thresholding, and higher order statistics. *Geophysics*, 77(6):KS43–KS54, 2012.
  - [162] M. Van Den Ende and J.-P. Ampuero. Automated seismic source characterisation using deep graph neural networks. *Geophysical Research Letters*, 47(17):e2020GL088690, 2020.
  - [163] M. van den Ende, I. Lior, J.-P. Ampuero, A. Sladen, A. Ferrari, and C. Richard. A self-supervised deep learning approach for blind denoising and waveform coherence enhancement in distributed acoustic sensing data. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
  - [164] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.
  - [165] U. b. Waheed, T. Alkhalifah, E. Haghighat, C. Song, and J. Virieux. Pinntomo: Seismic tomography using physics-informed neural networks. *arXiv preprint arXiv:2104.01588*, 2021.
  - [166] F. Waldhauser, W. L. Ellsworth, and A. Cole. Slip-parallel seismic lineations on the Northern Hayward Fault, California. *Geophysical Research Letters*, 26(23):3525–3528, 1999. \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1029/1999GL010462>.
  - [167] A. Wang, S. Li, J. Lu, H. Zhang, B. Wang, and Z. Xie. Prediction of pga in earthquake early warning using a long short-term memory neural network. *Geophysical Journal International*, 234(1):12–24, 2023.
-



- [168] C.-Y. Wang, T.-C. Huang, and Y.-M. Wu. Using lstm neural networks for onsite earthquake early warning. *Seismological Research Letters*, 93(2A):814–826, 2022.
- [169] J. Wang and Z. Duan. Agent ai with langgraph: A modular framework for enhancing machine translation using large language models. *arXiv preprint arXiv:2412.03801*, 2024.
- [170] L. M. Wang and L. C. Brill. Speech and noise levels measured in occupied k–12 classrooms. *The Journal of the Acoustical Society of America*, 150(2):864–877, 2021.
- [171] Y. Wang. Deciphering the enigma: A deep dive into understanding and interpreting llm outputs. *Authorea Preprints*, 2023.
- [172] Z. Wang, Z. Chen, and H. Zhang. Eqgraphnet: Advancing single-station earthquake magnitude estimation via deep graph networks with residual connections. *Artificial Intelligence in Geosciences*, 5:100089, 2024.
- [173] E. Weber, G. Iannaccone, A. Zollo, A. Bobbio, L. Cantore, M. Corciulo, V. Convertito, M. Di Crosta, L. Elia, A. Emolo, et al. Development and testing of an advanced monitoring infrastructure (isnet) for seismic early-warning applications in the campania region of southern italy. *Earthquake early warning systems*, pages 325–341, 2007.
- [174] M. Weber, D. Fu, Q. Anthony, Y. Oren, S. Adams, A. Alexandrov, X. Lyu, H. Nguyen, X. Yao, V. Adams, et al. Redpajama: an open dataset for training large language models. *arXiv preprint arXiv:2411.12372*, 2024.
- [175] J. Woollam, J. Münchmeyer, F. Tilmann, A. Rietbrock, D. Lange, T. Bornstein, T. Diehl, C. Giunchi, F. Haslinger, D. Jozinović, A. Michellini, J. Saul, and H. Soto. Seisbench—a toolbox for machine learning in seismology. *Seismological Research Letters*, 93(3):1695–1709, 2022.
- [176] H. Xue and F. D. Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2023.
- [177] L. Yang, F. Wu, Y. Wang, J. Gu, and Y. Guo. Masked graph convolutional network. In *IJCAI*, pages 4070–4077, 2019.
- [178] H. Yen, F. G. Germain, G. Wichern, and J. Le Roux. Cold diffusion for speech enhancement. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.
- [179] H. Zhang, D. Melgar, V. Sahakian, J. Searcy, and J.-T. Lin. Learning source, path and site effects: Cnn-based on-site intensity prediction for earthquake early warning. *Geophysical Journal International*, 231(3):2186–2204, 2022.
- [180] P. Zhang, G. Zeng, T. Wang, and W. Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.
- [181] R. Zhang, Y. Liu, and H. Sun. Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling. *Engineering Structures*, 215:110704, 2020.

- [182] R. Zhang and T. J. Ulrych. Physical wavelet frame denoising. *Geophysics*, 68(1):225–231, 2003.
- [183] X. Zhang, J. Zhang, C. Yuan, S. Liu, Z. Chen, and W. Li. Locating induced earthquakes with a network of seismic stations in oklahoma via a deep learning method. *Scientific reports*, 10(1):1941, 2020.
- [184] J. Zhu, S. Li, J. Song, and Y. Wang. Magnitude estimation for earthquake early warning using a deep convolutional neural network. *Frontiers in Earth Science*, 9:653226, 2021.
- [185] W. Zhu and G. C. Beroza. Phasenet: A deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, 216(1):261–273, 2019.
- [186] W. Zhu, I. W. McBrearty, S. M. Mousavi, W. L. Ellsworth, and G. C. Beroza. Earthquake phase association using a bayesian gaussian mixture model. *Journal of Geophysical Research: Solid Earth*, 127(5):e2021JB023249, 2022.
- [187] W. Zhu, S. M. Mousavi, and G. C. Beroza. Seismic signal denoising and decomposition using deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11):9476–9488, 2019.

# Appendix A

## Supplementary material for Chapter 2

### A.0.1 Diffusion Model overview

In this Appendix, we will briefly explain how *DMs* work. *DMs* are inspired by the physical process of diffusion, where particles spread out from areas of higher concentration to areas of lower concentration over time [149]. In the context of generative modeling, this process is simulated in a reverse manner. The model starts with a distribution of random noise and gradually refines this noise into a coherent sample from the target distribution over a series of steps. The theoretical foundation of *DMs* is rooted in stochastic differential equations (SDEs) and involves two key phases: the forward diffusion (or noise addition) process and the reverse diffusion (or denoising) process.

**Forward diffusion process:** In this phase, the model incrementally adds noise to data from the original distribution over a series of steps, transforming it into a distribution of pure noise. Mathematically, this can be represented as a Markov chain that gradually transitions the data distribution  $p(x_0)$  to a noise distribution  $p(x_T)$ , where  $T$  is the total number of diffusion steps and  $x_0$  to  $x_T$  represents the data at each step of the forward diffusion process. The step sizes are controlled by a variance schedule  $\{\beta_t \in (0, 1)\}_{t=1}^T$ .

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I) \quad (\text{A.1})$$

$$q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1}) \quad (\text{A.2})$$

A nice property of the process above is that we can reparametrize  $x_t$  in terms of  $\epsilon$  (i.e. the added gaussian noise), which is independent of the model parameters, allowing the gradient of the loss function to be backpropagated through the deterministic part of the model. We can perform this using the reparametrization trick. Let  $\alpha_t = 1 - \beta_t$  and  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ :

$$\begin{aligned} x_t &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \\ &= \dots \\ &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon \\ q(x_t|x_0) &= N(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \end{aligned} \quad (\text{A.3})$$

**Reverse diffusion process:** The reverse process involves learning to denoise the data, starting from the noise distribution and progressively reconstructing the data distribution through a series of learned denoising steps. The goal of the model during this phase is to learn the conditional distribution  $p(x_{t-1}|x_t)$ .

$$p_\theta(x_{0:T}) = p(x_T) \prod p_\theta(x_{t-1}|x_t) \quad (\text{A.4})$$

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (\text{A.5})$$

For a more detailed discussion, please refer to the paper: [63].

**Training:** Training *DMs* involves optimizing the parameters of the reverse diffusion process to minimize the difference between the original data distribution and the distribution of the generated samples. This is typically achieved through variational inference, where the model learns to predict the noise that was added at each step of the forward process, thereby allowing it to reverse the diffusion.

---

**Algorithm 2** Diffusion Model Training

---

**repeat**

$x_0 \sim q(x_0)$

$t \sim \text{Uniform}(\{1, \dots, T\})$

$\epsilon \sim N(0, I)$

Take gradient descent step on  $\nabla_\theta \|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t)\|^2$

**until** converged

---

The training of *DMs* begins with the application of the forward diffusion process to the training data, which generates noisy versions of the data at various timesteps, sampled using the uniform distribution in Algorithm 2. Following this initial step, the model begins the phase of noise prediction for each noisy sample. It attempts to accurately predict the specific noise that was added at each timestep. The accuracy of this prediction is measured against the actual noise used during the forward process, utilizing typically the mean squared error (MSE) as the loss function. Once the loss has been calculated, it is then backpropagated through the model to update its parameters. It is during this phase that the reparametrization trick plays an important role, as it allows for the gradients to flow through the stochastic sampling of noise, thus enabling the optimization process to proceed (letting the  $\epsilon$  parameter to be a trainable parameter).

**Sampling:** After the model has been trained to reverse diffusion process. It can generate new samples starting from noise, or denoise new inputs. This process is the inverse of the forward diffusion process, where noise is gradually added to the data. Instead, starting with a purely noisy distribution, the model iteratively generates/denoise data that increasingly resembles the target distribution. The Algorithm 3 gives a closer look at how the sampling process unfolds. The sampling process begins with an initial noise vector sampled from a Gaussian distribution. This noise vector  $x_T$  represents the final state of the forward diffusion process and serves as the starting point for generation. From this initial state, the model iteratively applies the learned reverse diffusion steps to reduce the noise and move closer to the data distribution. At each step  $t$ , the model uses its parameters to estimate the cleaner version of the current state  $x_{t-1}$  from  $x_t$ . This is based on the

**Algorithm 3** *DM Sampling*


---

```

 $x_T \sim N(0, I)$ 
for  $t = T, \dots, 1$  do
 $z \sim N(0, I)$  if  $t > 1$ , else  $z = 0$ 
 $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, t)) + \sigma_t z$ 
end for
return  $x_0$ 

```

---

conditional probability learned during training, which models how to reverse the noise addition for that particular step. After the final reverse diffusion step, the output is a sample that closely resembles/denoises the target data distribution. This sample is the model's "best guess" at a real data point, having transformed from pure noise (for generation) or a noised input (for denoising) to structured data through the reverse diffusion process. The sampling process in *DMs* exemplifies how structured data can emerge from randomness (for generation) or from noisiness (for denoising) through iterative refinement, helped by the complex statistical relationships learned during training.

**A.0.2 Cold Diffusion Model overview**

Cold *DMs* are very recent designs, and currently, there are limited works implementing such architecture. The original concept of *DMs* [11] involves extending and generalizing degradation using non-Gaussian noise. This becomes achievable due to enhancements in the sampling algorithm. In particular, as stated in [11], they start from a simple assumption: the original sampling: Algorithm 3 works well when the restoration operator is perfect. This means that:

$$R(D(x_0, t), t) = x_0 \text{ for all } t. \quad (\text{A.6})$$

With restoration operator is  $p_\theta$  in Equation A.5 that here is referred as  $R$ . However, in the scenario where the restoration is imperfect, this causes the model to make errors, leading it to deviate from  $D(x_0, s)$ ,  $D$  stands for Degradation operator hence:  $q(x_{1:T}|x_0)$  in Equation A.1. The implemented sampler possesses excellent mathematical capabilities that are not detailed in this work (for further details, refer to 3.3 Properties of the Algorithm in [11]), enabling the accurate reconstruction of the signal even in cases where the restoration operator  $R$  fails to completely invert  $D$ .

**Algorithm 4** Improved Sampling for Cold Diffusion

---

```

Input: A degraded sample  $x_t$ 
for  $s = t, t-1, \dots, 1$  do
 $\hat{x}_0 \leftarrow R(x_s, s)$ 
 $\hat{x}_{s-1} = x_s - D(\hat{x}_0, s) + D(\hat{x}_0, s-1)$ 
end for

```

---

As a starting point to address our task, we have taken animorphosis as a reference (for further details, see Section 5.3 "Generation using other transformations" [11]). In this context, a "clean" sample (an image of a person) is systematically subjected to a series of transformations resulting in an out-of-domain "degraded" sample (an image of an animal). However, it's important to highlight that our approach deviates from this process. Our degraded sample retains the underlying information of the

clean sample, as our degradation process now introduces an out-of-domain sample in conjunction with the clean sample. Such a degradation does not correspond to any of those addressed in [11].

### A.0.3 Gaussian Diffusion Model

Figure A.1 shows insights into preliminary experiments focusing on Gaussian methods, specifically on results using only Gaussian noise. We aim to showcase both the validity of *DMs* and the limitations of employing Gaussian noise within a seismic context.

### A.0.4 Qualitative Comparison

Figures from A.2 to A.8 provide additional examples of the Denoiser’s application in specific scenarios, showing comparisons between different methods, cases highlighting good performances as well as criticalities.

### A.0.5 Comparison with other methods

This Section is an expansion of the comparative analyses. Here we show the results obtained with a classical method (Wiener Filter) and with a version of *DD* to which we added an attention mechanisms (*AttDD*). This section concludes with both quantitative (Figures from A.9 to A.11) and qualitative (Figures A.12, A.13) representations of the results.

**Classic Method: Wiener Filter** The Wiener filter is a statistical approach used in signal processing to produce an estimate of a desired or target signal by suppressing noise from a measured signal. It is designed to minimize the mean square error between the estimated random process and the desired process. In our specific case, we utilized the Wiener filter implementation found in SciPy [164]. This implementation is grounded on the methodologies and principles detailed in the following paper: [94].

**Improved Deep Denoiser with Attention: AttDD** We adopted the *DD* model from Seis-Bench [175] and introduced self-attention to it. This modification was undertaken to discern whether the improvements attributed to *CDiffSD* were primarily due to the introduction of attention mechanisms or derived from the comprehensive framework associated with the diffusive model. Specifically, we decided to create three different models to evaluate the impact of attention mechanisms at various stages of the convolutional process.

- *AttDDI*: Incorporates the self-attention mechanism immediately after the first convolutional layer. In this model, the self-attention layer at the beginning focuses on capturing global dependencies and contextual information early in the network. By positioning the self-attention layer at the start of a UNet architecture, we aim to enhance the model’s ability to recognize and prioritize relevant features across the entire input space.
- *AttDDII*: Features self-attention layers following every convolutional layer throughout the model. The self-attention layers are placed after every convolutional layer within the UNet architecture. By applying self-attention after each convolutional layer, the model ensures

that both local and global dependencies are considered throughout the depth of the network, enhancing the ability to preserve intricate details while effectively reducing noise.

- *AttDDIII*: Implements the self-attention mechanism only after the final convolutional layer. In this configuration, where the self-attention layer is positioned in the last convolutional layer, the model emphasizes important features and details at the final stages of processing.

This differentiation in the placement of self-attention layers across the three models was designed to systematically evaluate the efficacy of integrating attention mechanisms within deep denoising frameworks and their influence on seismic data analysis and interpretation.

**Comparison** Here we report the same metrics and comparisons as in the main text, but including the additional denoising techniques described above.

Wiener filter shows the lowest performance for all considered metrics: *SNR* (A.9), *CC* (A.10) and picking (A.11).

Interpretation of *AttDD* results, on the other hand, are less straightforward to interpret. While *CC* values are in line with *DD* (A.10), *SNR* is much higher than any other method and also higher than the original *EQ* signals (A.9). The distribution of picks time difference shows a double peak for *AttDD* (A.11), which indicates that frequently a P is detected about 20 samples before the true P. The reason for these results becomes clearer by inspecting the waveforms. We show two examples comparing *DD* and *AttDD* in A.12 and A.13. The cause of both the high *SNR* values and the picks distribution can be attributed to the very strong denoising of *AttDD* before the P arrival, that often almost zeroes out the amplitudes. The anomalous *SNR* values are therefore due to the very low  $\sigma_{noise}$ , while the incorrect picks are detected by PhaseNet as soon as the waveform amplitude rises slightly from the "zeroed level".

These results lead us to think that the better performance of *CDiffSD* described in the main text are not due not only to the attention mechanism (not featured in the original *DD*), but to the whole framework associated with the diffusive models.

### A.0.6 Training & Testing Details

Here we detail the hyperparameter tuning process (Table A.2) and the operational considerations for model training & testing (Table A.1).

#### Training Details

Table A.1 provides a comprehensive comparison among various *ML* models specifically designed for denoising applications. The models evaluated include *DD*, *AttDDI*, *AttDDII*, *AttDDIII*, and *CDiffSD* across different configurations ( $T=20$ ,  $T=100$ ,  $T=300$ ). The comparison criteria focus on Training Time (minutes per epoch), Model Parameters (indicating the complexity of each model), Testing Time (minutes per test set), and GPU Memory usage (GB). These metrics offer insight into the efficiency, complexity, and operational demands of each model. All models were benchmarked with a batch size of 16 and utilizing a single NVIDIA RTX 4090 GPU.

Regarding the other training parameters: we set the learning rate at 0.001, a value that strikes a balance between rapid convergence and the risk of potentially overshooting the minimum loss.

Recognizing the importance of adaptability in learning rate adjustment, we implemented a step scheduler. This scheduler is configured with a gamma value of 0.1 and a step\_size of 8. Such configuration means that for every 8 epochs, the learning rate is reduced by a factor of 10. This methodical reduction allows the model to make larger updates initially for faster learning, and then refine its adjustments with smaller steps as it converges towards optimal weights, enhancing the precision of the model's learning process. In tandem with our learning rate strategy, we selected the Adam optimizer for weight updates.

**Hyperparameter Tuning: a sensitive analysis** For the tuning of hyperparameters, we have explored a diverse array of hyperparameter settings to optimize our model's performance. The specific hyperparameters under investigation are as follows:

- scheduler type: the scheduler that controls the level of noise introduced at each step in the forward process. We evaluated two scheduler types: ["linear", "cosine"].
- s: s is a parameter that regulates the behavior of the cosine scheduler, specifically how the beta values are calculated over the timesteps: [0.0001, 0.0008, 0.001].
- RNF: This factor, the reduce noise factor, is a ratio used to control the level of normalized noise before it's introduced into the diffusion process: [(50, 75), (30, 80), (40, 65), (10, 90)].

To systematically explore and optimize these hyperparameters, we utilized Optuna [4], a highly efficient hyperparameter optimization framework. Optuna enables the automation of hyperparameter search by efficiently navigating the space of possible parameter configurations through trial and error, guided by past trial outcomes.

For our study, we decided to conduct 30 trials to thoroughly investigate the impact of the aforementioned hyperparameters on model performance. Having previously conducted a thorough analysis on the timestep, here we have set  $T=50$  and conducted an analysis on the other hyperparameters. Below is a synthetic table summarizing the conducted study, highlighting the best-performing hyperparameters as identified by Optuna:



## A.0.7 Supplementary tables

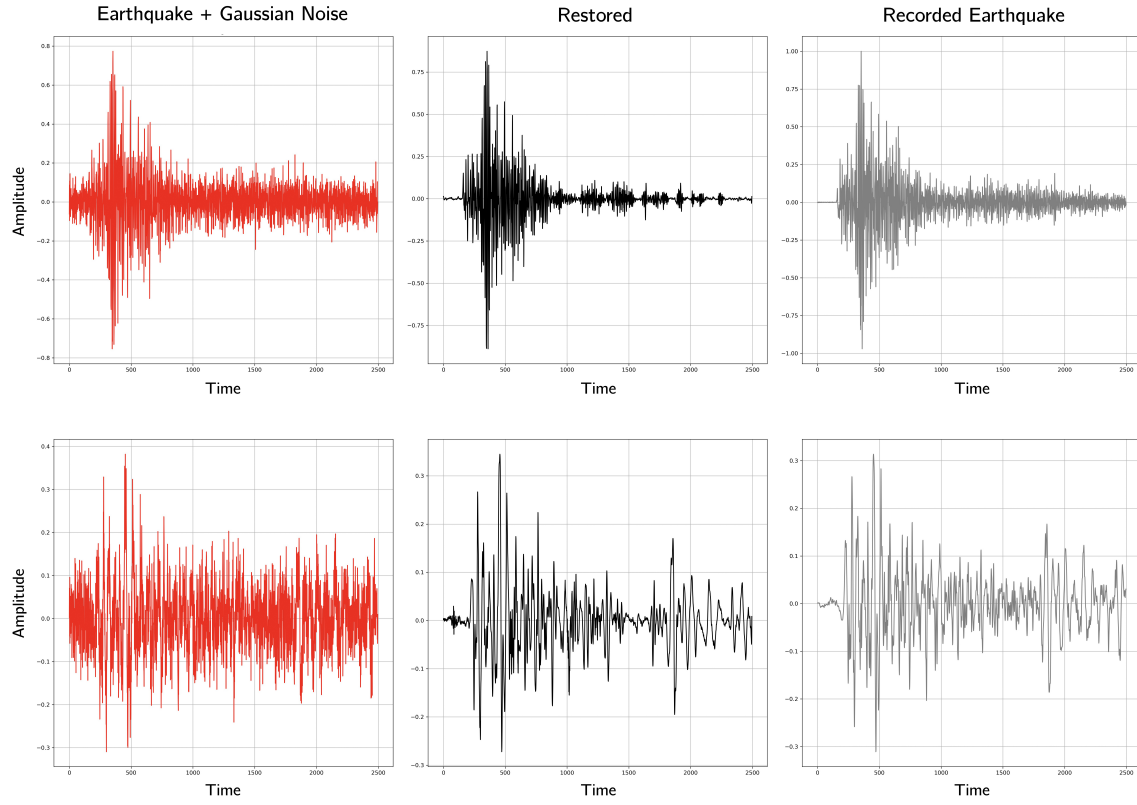
| Model       | Training Time<br>(minutes per epoch) | Model<br>Parameter | Testing Time<br>(minutes per Test Set) | GPU Memory<br>(GB) |
|-------------|--------------------------------------|--------------------|----------------------------------------|--------------------|
| DD          | 0.36                                 | 1.37M              | 0.107                                  | 0.71               |
| AttDD I     | 2.20                                 | 1.38M              | 0.128                                  | 7.44               |
| AttDD II    | 3.10                                 | 1.52M              | 0.145                                  | 8.16               |
| AttDD III   | 2.33                                 | 1.38M              | 0.131                                  | 7.84               |
| CDiffSD 20  | 5.02                                 | 14.8M              | 5.13                                   | 11.38              |
| CDiffSD 100 | 5.02                                 | 14.8M              | 10.37                                  | 11.38              |
| CDiffSD 300 | 5.02                                 | 14.8M              | 53.28                                  | 11.38              |

**Table A.1:** Training Time and Hardware Details

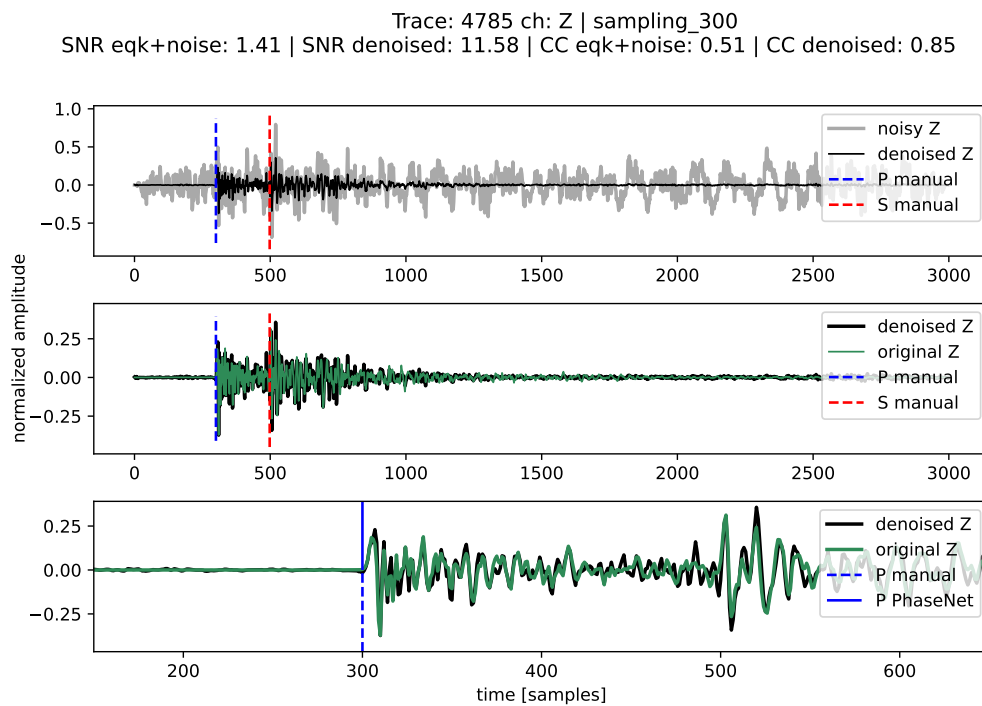
| Trial    | Timestep | Scheduler<br>(Type) | s      | Range RNF | Model Performance<br>(in Val. Set) |
|----------|----------|---------------------|--------|-----------|------------------------------------|
| Trial 13 | 50       | cosine              | 0.0008 | (40, 65)  | 0.025467                           |
| Trial 19 | 50       | cosine              | 0.0010 | (50, 75)  | 0.025588                           |
| Trial 28 | 50       | cosine              | 0.0001 | (40, 65)  | 0.026169                           |
| Trial 27 | 50       | cosine              | 0.0008 | (30, 80)  | 0.026911                           |
| Trial 15 | 50       | cosine              | 0.0008 | (50, 75)  | 0.028347                           |
| Trial 29 | 50       | linear              | /      | (10, 90)  | 0.029513                           |

**Table A.2:** Sensitivity Analysis on hyper-parameters

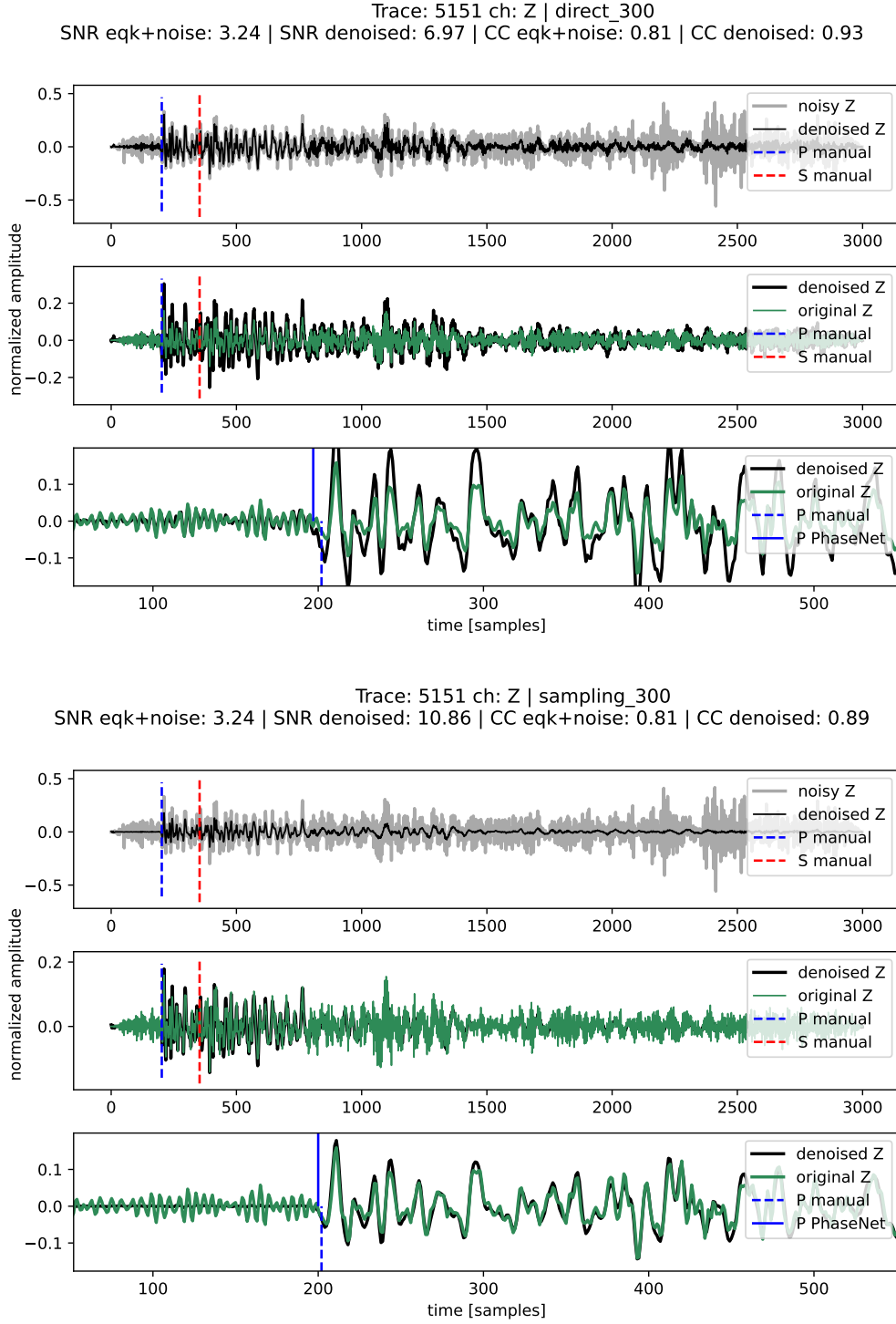
## A.0.8 Supplementary figures



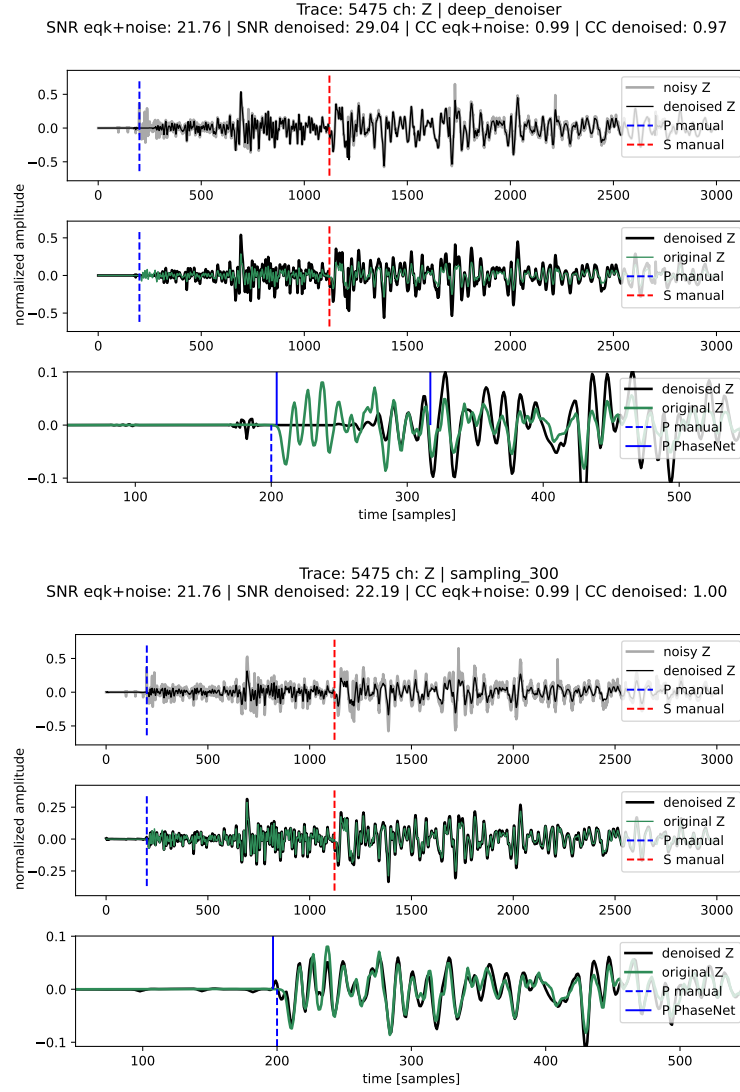
**Figure A.1:** This example clearly highlights that the model has to be trained with real noise to be able to remove it accurately. This also proves that the seismic noise we want to remove is not just Gaussian noise, and the method has to be refined according to our final objective.



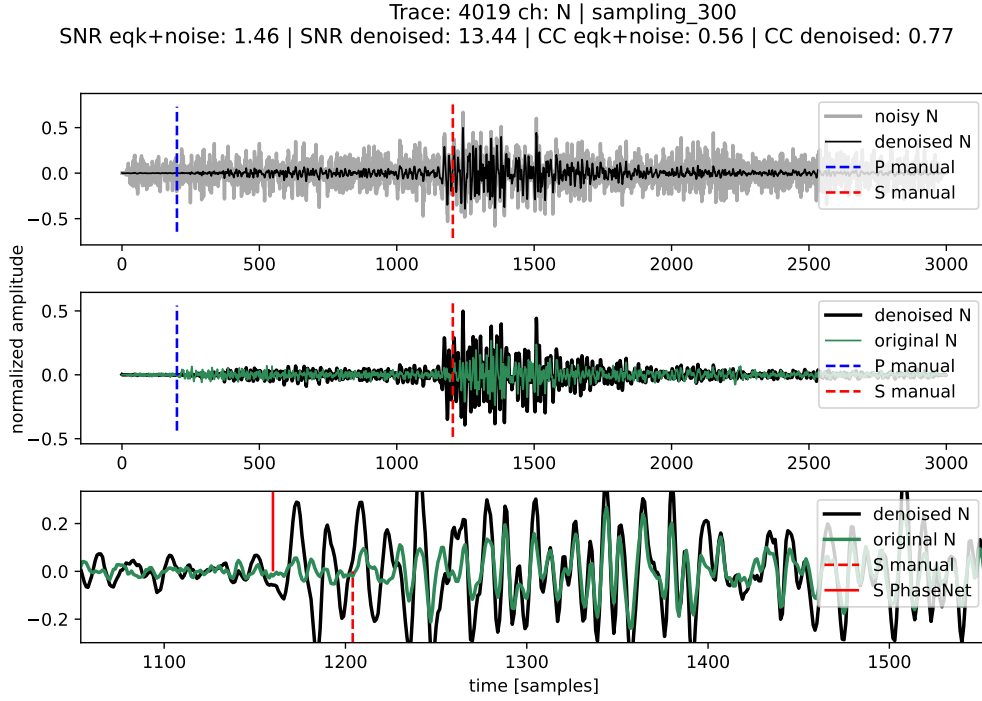
**Figure A.2:** *SNR* increased by more than 10 dB after denoising. P arrival is preserved and the automatic pick on the denoised signal is aligned with the manual label.



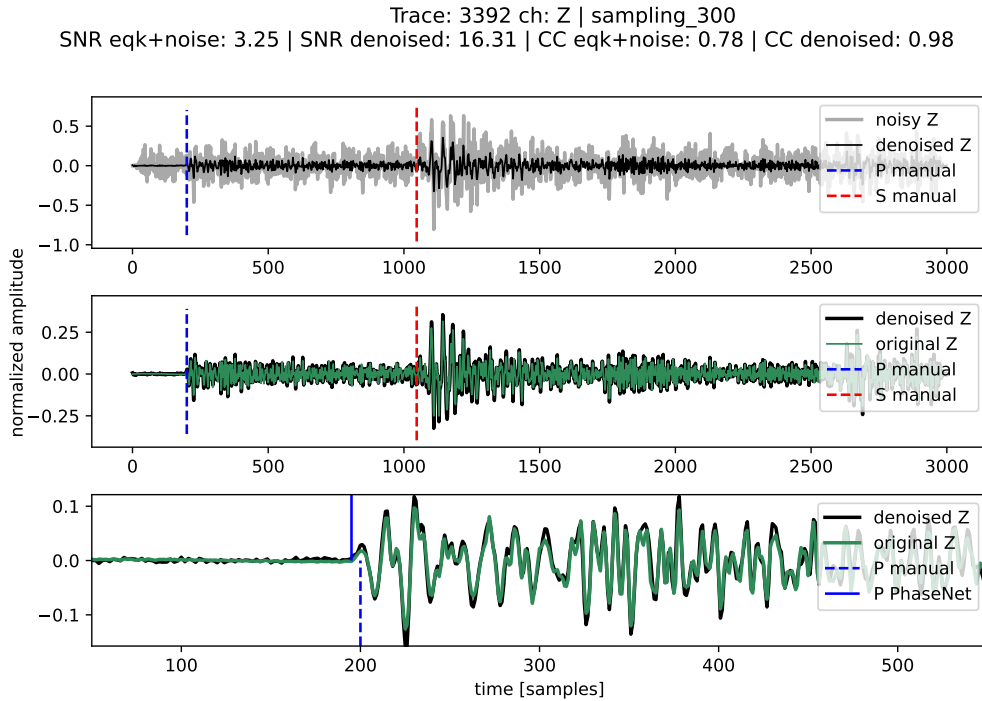
**Figure A.3:** comparison of 'direct\_300' and 'sampling\_300' methods. The 'direct\_300' retains some of the noise before the P-wave, which is present also in the original trace. The 'sampling\_300' method instead effectively removes it, demonstrating its superior noise reduction capability in the pre-P-wave segment.



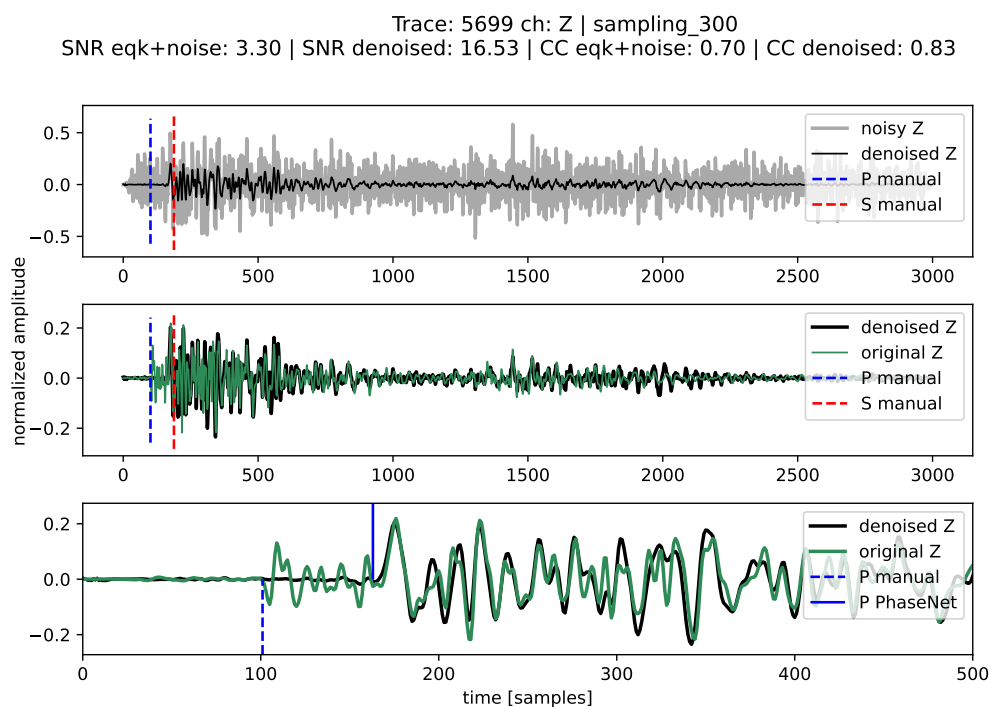
**Figure A.4:** Comparison between *DD* and 'sampling\_300' in a high *SNR* conditions. This example elaborates on the situation described in the main text about cases with high *SNR* in input. This figure provides an example of the fact that the value of the output *SNR* could be misleading sometimes in evaluating the actual performance of the denoiser. Here the *SNR* values on denoised traces do not reflect the actual quality of the denoising: while *DD* shows a higher *SNR* value, it clearly deletes one second of signal after the labeled P wave arrival. 'sampling\_300' instead does a better job in preserving the P wave.



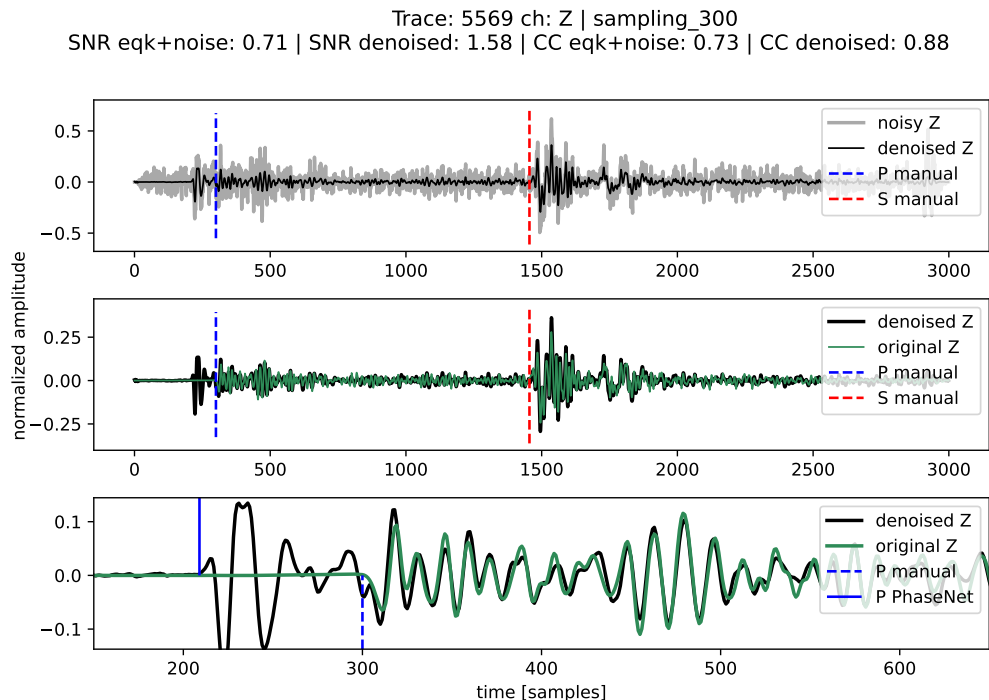
**Figure A.5:** The automatic S pick on the horizontal channel N is early with respect to the labeled one. This could indicate either a very good denoising performance of the 'sampling\_300' model leading to a better S pick or unfiltered noise which is mistaken by PhaseNet for an S- arrival.



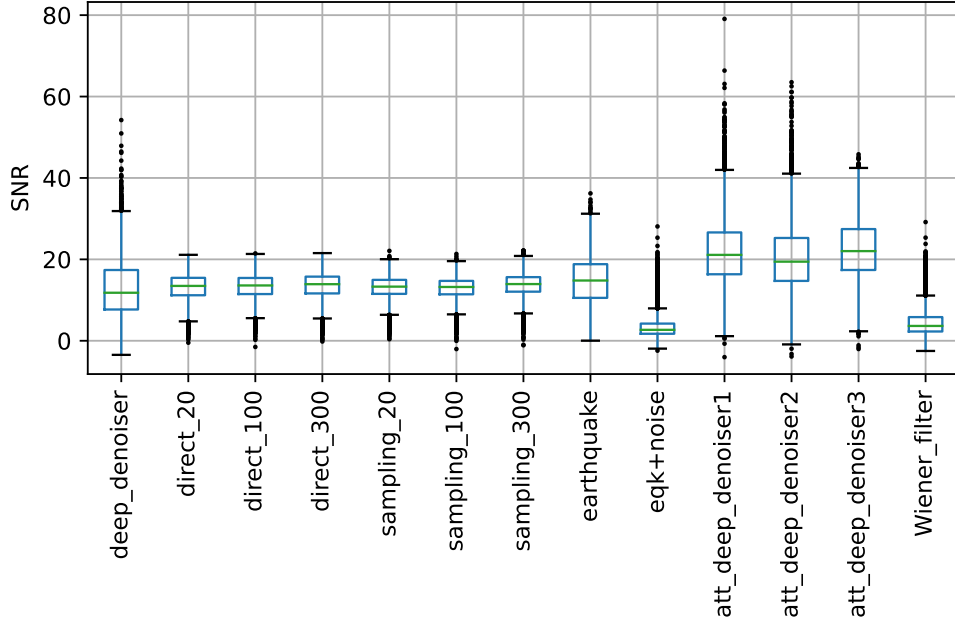
**Figure A.6:** A low-amplitude noise that was not present in the original trace is instead visible in the denoised trace before the P wave arrival.



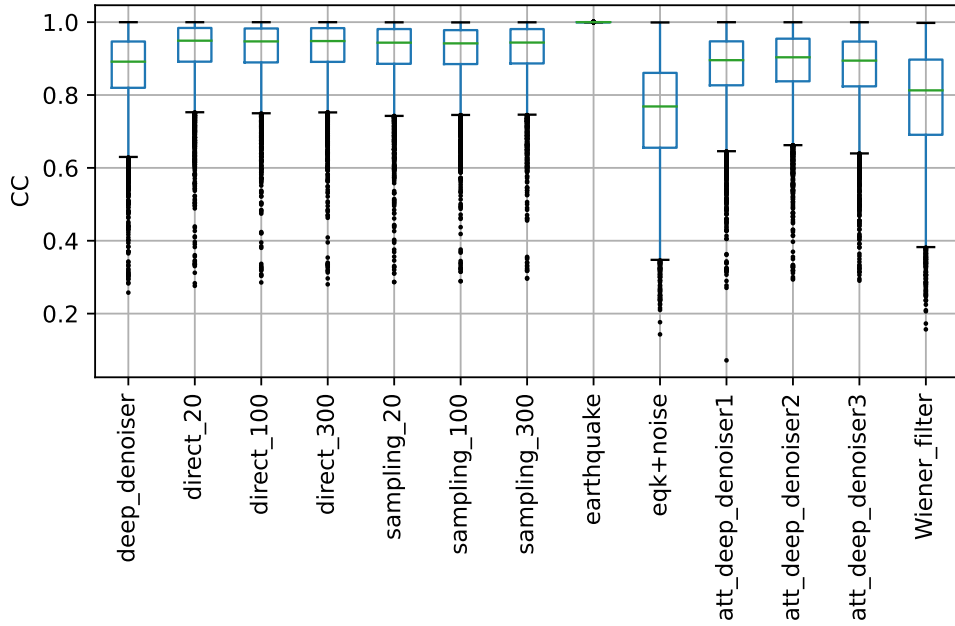
**Figure A.7:** The P wave is treated as noise and filtered out, this results in an incorrect late P pick by PhaseNet.



**Figure A.8:** High amplitude noise before the P wave arrival is not filtered out. As a result the P pick is placed almost 100 samples early with respect to the true pick.

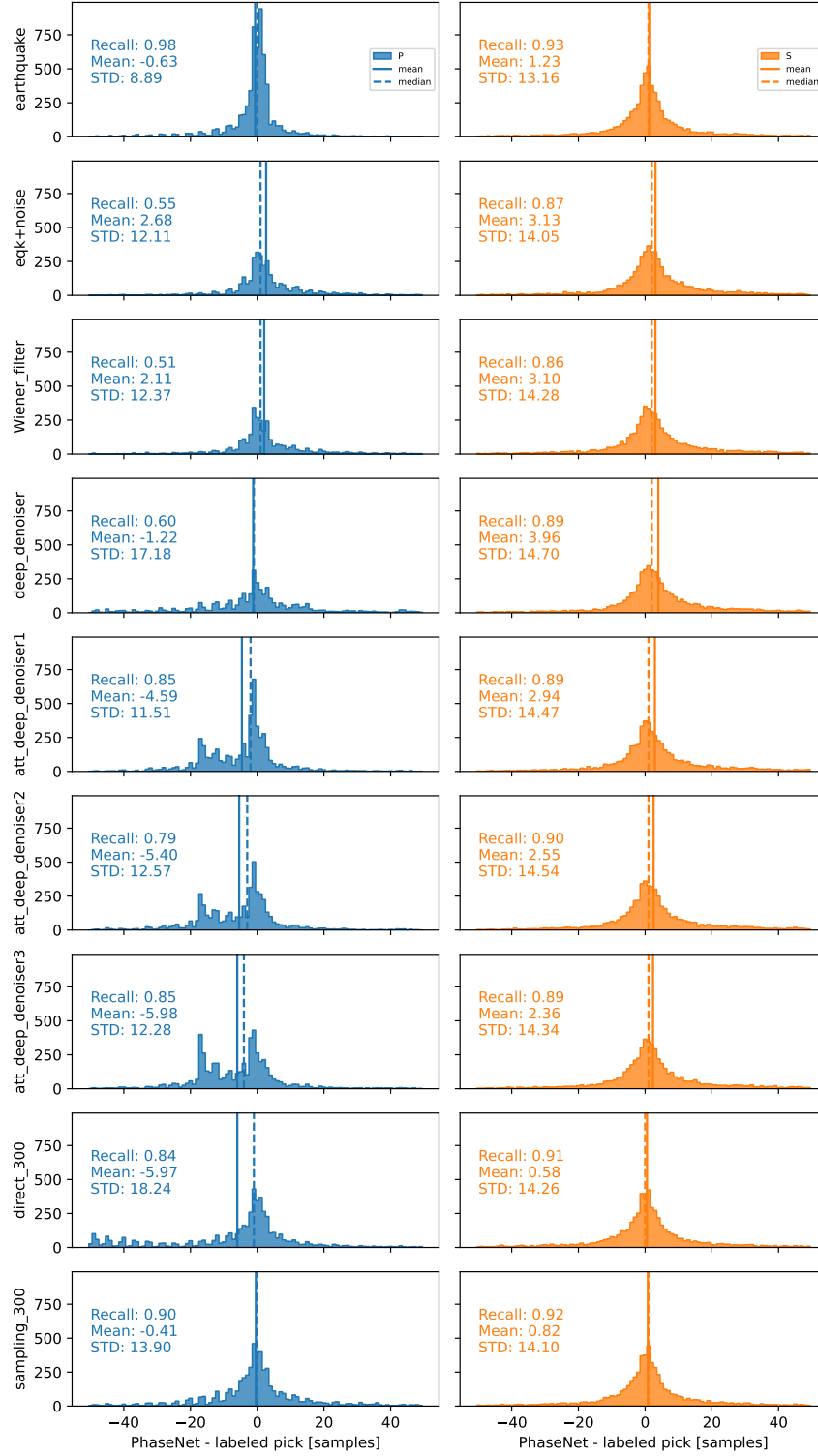


**Figure A.9:** Same *SNR* boxplot as in the main text, but with additional methods: AttDD (3 versions) and Wiener filter. Wiener filter shows lower performances than any other method. AttDD exhibit a high *SNR*, due to the very strong denoising of the waveform before the P arrival.

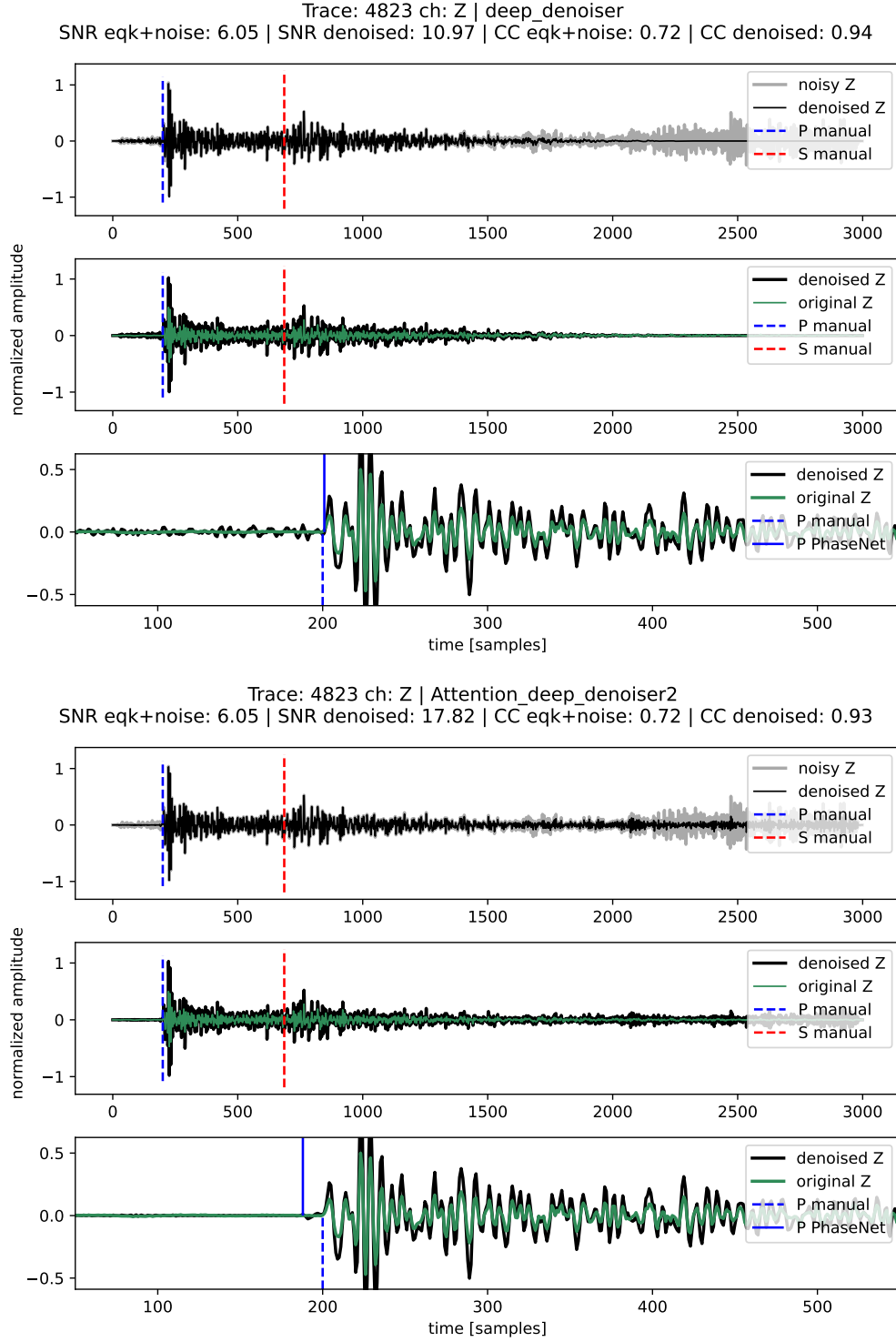


**Figure A.10:** Same *CC* boxplot as in the main text, but with additional methods: AttDD (3 versions) and Wiener filter. Wiener filter shows lower performances than any other method. The performances of AttDD are aligned with the original DD.

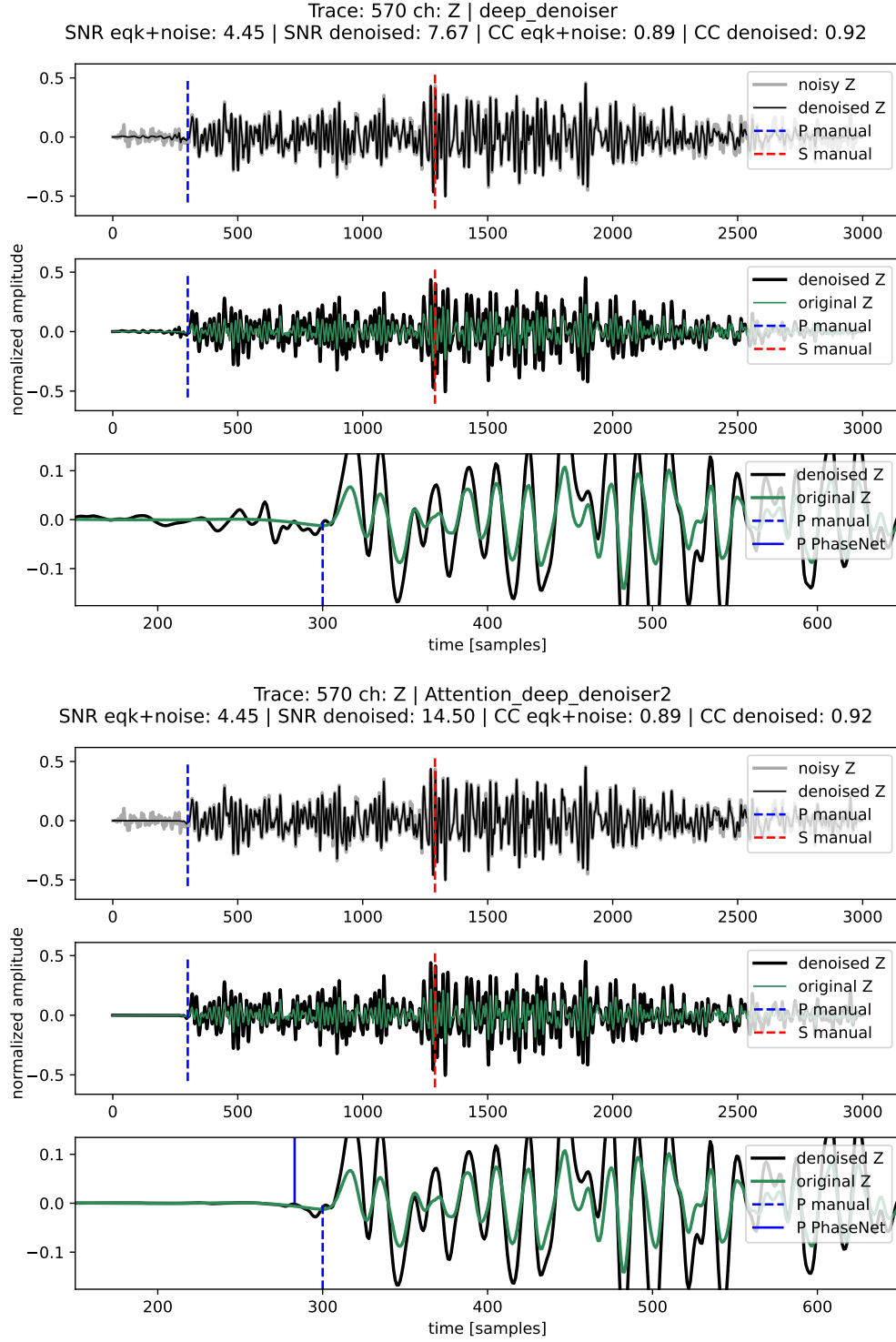




**Figure A.11:** Histograms of picks difference between PhaseNet and manual picks comparing additional methods. Wiener filter shows little improvement with respect to applying PhaseNet directly on the noisy waveforms (eqk+noise). The behaviour of the 3 AttDD methods exhibit double maxima in the distributions, highlighting a frequent early pick around 20 samples before the labeled one.



**Figure A.12:** Comparison of *DD* vs *AttDD*. This figure highlights both the reason for the very high *SNR* of *AttDD* and the double maxima shown in A.11. Despite a lower *SNR*, the waveform denoised with *DD* picks the P arrival correctly. In *AttDD* instead PhaseNet picks a P as soon as the amplitude rises from the almost-zero level at the beginning.



**Figure A.13:** Comparison of *DD* vs *AttDD*. In this example *DD* doesn't denoise the waveform enough to enable a P pick. *AttDD* does a better job in denoising the first portion of the waveform, but the transition from noise to signal is too abrupt, giving again rise to an early pick.

## Appendix B

# Supplementary Material for Chapter 3

| Experiment | A   | B   | C   | PGA   | PGV   | SA(0.3) | SA(1.0) | SA(3.0) |
|------------|-----|-----|-----|-------|-------|---------|---------|---------|
| 1          | no  | no  | yes | 0.188 | 0.199 | 0.182   | 0.230   | 0.267   |
| 2          | yes | yes | yes | 0.205 | 0.210 | 0.195   | 0.243   | 0.280   |
| 3          | yes | yes | no  | 0.202 | 0.209 | 0.194   | 0.241   | 0.276   |
| 4          | no  | yes | no  | 0.205 | 0.219 | 0.198   | 0.257   | 0.300   |
| 5          | yes | no  | no  | 0.197 | 0.209 | 0.192   | 0.243   | 0.280   |
| 6          | no  | yes | yes | 0.189 | 0.200 | 0.181   | 0.229   | 0.268   |
| 7          | no  | no  | no  | 0.198 | 0.207 | 0.190   | 0.238   | 0.274   |
| 8          | yes | no  | yes | 0.208 | 0.218 | 0.201   | 0.254   | 0.290   |

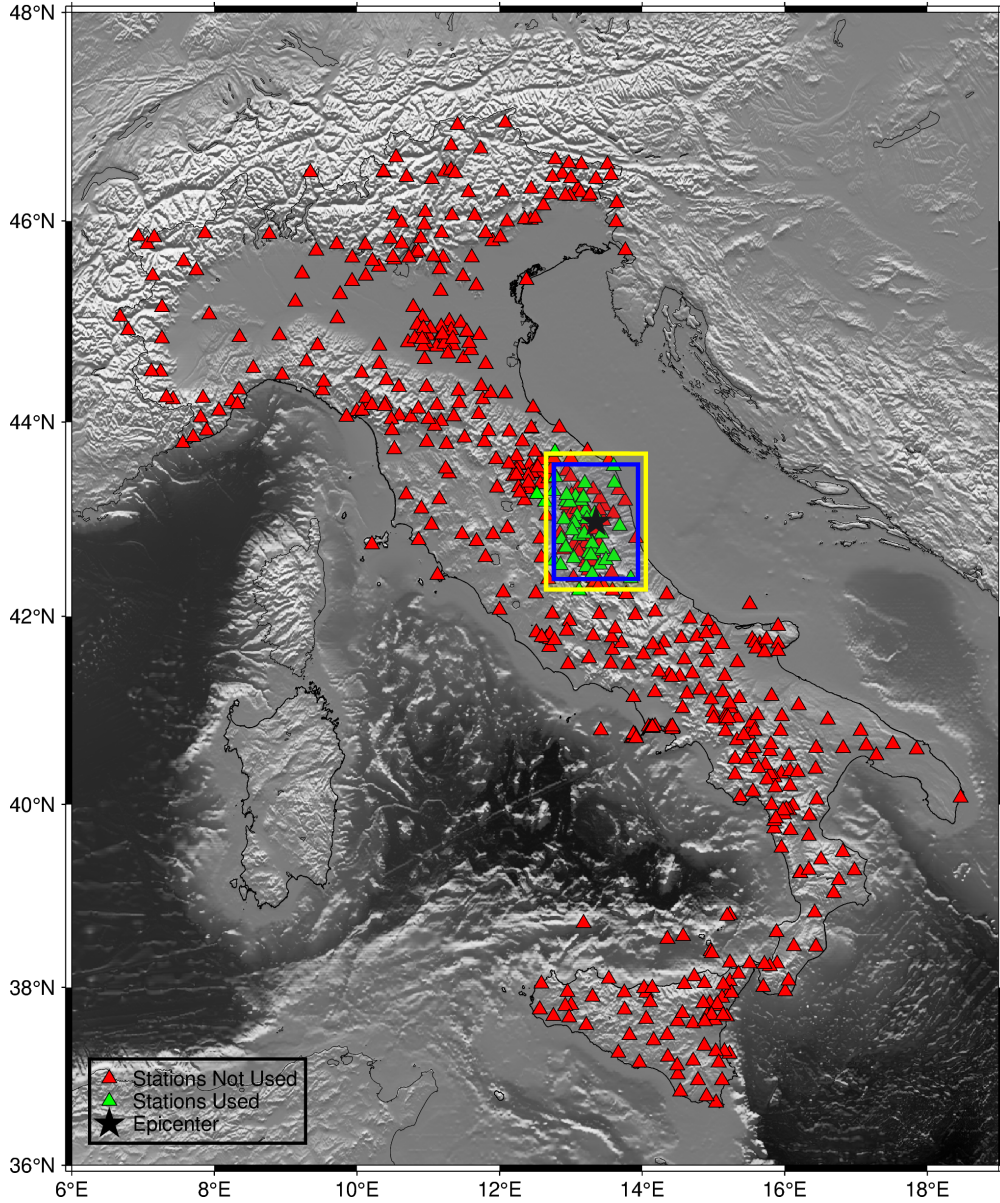
**Table B.1:** MSE results for the eight experiments on our data set. The letters A, B, and C denote the coordinates of the first station that records the P-wave, the interstation distances and the maximum amplitude information, respectively. These values are inserted as additional metadata into the flattened layer.

| IM             | ML (Masked GCN) |        |       | GMM (Bindi et al., 2011) |        |       |
|----------------|-----------------|--------|-------|--------------------------|--------|-------|
|                | Median          | Mean   | STD   | Median                   | Mean   | STD   |
| <b>PGA</b>     | 0.008           | 0.006  | 0.318 | -0.037                   | -0.027 | 0.310 |
| <b>PGV</b>     | 0.004           | 0.008  | 0.310 | -0.240                   | -0.214 | 0.289 |
| <b>SA(0.3)</b> | -0.019          | -0.007 | 0.323 | -0.292                   | -0.265 | 0.319 |
| <b>SA(1.0)</b> | -0.024          | -0.005 | 0.346 | -0.259                   | -0.240 | 0.313 |
| <b>SA(3.0)</b> | -0.034          | -0.012 | 0.362 | -0.037                   | -0.027 | 0.310 |

**Table B.2:** IM residual statistics  $\log_{10} \frac{IM_{obs}}{IM_{pred}}$  for the Masked GCN predictions and for the estimates obtained using the Bindi et al. (2011) GMM. The comparison was performed solely for the observed IMs at stations with available recorded data. The statistics are calculated after the outlier residuals (those with a residual  $> |1|$ ) have been removed. The number of outliers of the GCN is 5.8% and for the GMM is 2.4%

| Model            | PGA  |      | PGV  |      | SA(0.3) |      | SA(1.0) |      | SA(3.0) |      |
|------------------|------|------|------|------|---------|------|---------|------|---------|------|
|                  | MAE  | MSE  | MAE  | MSE  | MAE     | MSE  | MAE     | MSE  | MAE     | MSE  |
| CNN              | 0.34 | 0.22 | 0.35 | 0.26 | 0.36    | 0.24 | 0.35    | 0.26 | 0.36    | 0.25 |
| TISER-GCN        | 0.31 | 0.20 | 0.32 | 0.21 | 0.31    | 0.19 | 0.31    | 0.20 | 0.32    | 0.21 |
| Masked TISER-GCN | 0.32 | 0.19 | 0.32 | 0.20 | 0.32    | 0.18 | 0.35    | 0.23 | 0.35    | 0.27 |

**Table B.3:** MAE and MSE results of each IM for the CNN model from Jozinovic et al. (2020), the TISER-GCN model from Bloemheuvel et al. (2023) and our proposed model (Masked TISER-GCN).



**Figure B.1:** This map illustrates the application of the mask on the nearly 600 seismic stations from the INSTANCE benchmark dataset (Michellini et al., 2021). A sample earthquake from those used in this study was examined: 9068331. All green triangles denote stations that detected the P-wave arrival within 10 seconds for this event according to the INSTANCE metadata, and for this reason, they are all included by the mask during the training phase. Conversely, all red triangles indicate stations where the P-wave does not arrive within the first 10 seconds starting from the P arrival at the first recording station, and for this reason, they are excluded by the mask in the training phase. In the test phase, the stations included by the mask are represented exclusively by the green triangles within the blue bounding box, located approximately  $\pm 0.58$  degrees from the first recording station. This setup replicates an operational scenario in which the model estimates travel times assuming a P-wave velocity of 6.5 km/s based on the P arrival at the closest station provided in the INSTANCE metadata. The yellow bounding box defines the area used for metric calculations, set at approximately  $\pm 0.7$  degrees from the first recording station.

## Appendix C

# Supplementary Material for Chapter 4

### C.0.1 Model input and output

Below is an example of model input and output after removing leading zeros and converting values to integers.

| Input :                   |
|---------------------------|
| (426760,133167,0,372)     |
| (427595,132086,54,2044)   |
| (427035,132248,83,18747)  |
| (425357,134090,103,8805)  |
| (426965,134483,112,12250) |
| (425089,132145,124,4854)  |
| (427569,132977,128,9017)  |
| (426699,131152,131,8578)  |
| (427249,131257,161,2759)  |
| (426227,136039,270,4636)  |
| (427515,130446,276,1475)  |
| (424531,135728,328,428)   |
| (428335,131142,335,18622) |
| (428993,133268,382,2565)  |
| Output :                  |
| 426227,132993,860,229,241 |

These are the input and output values for the event shown in Fig 4.1b. Each row in the input is in the form *latitude,longitude, $\Delta T$ ,PGV*, relating to a single station. The output is in the form *latitude,longitude,depth,magnitude,travel time of the first P-wave* and is related to the earthquake.

### C.0.2 Test set error distributions

Figure C.1 illustrates the location error distributions obtained on the test set events comparing the model estimates with the INGV bulletin values. They all include a few outliers, especially the depth and hypocenter distributions with errors up to 500 km and the epicenter up to 140 km.

Figure C.2 presents the error distribution of magnitude estimation, defined as the difference between

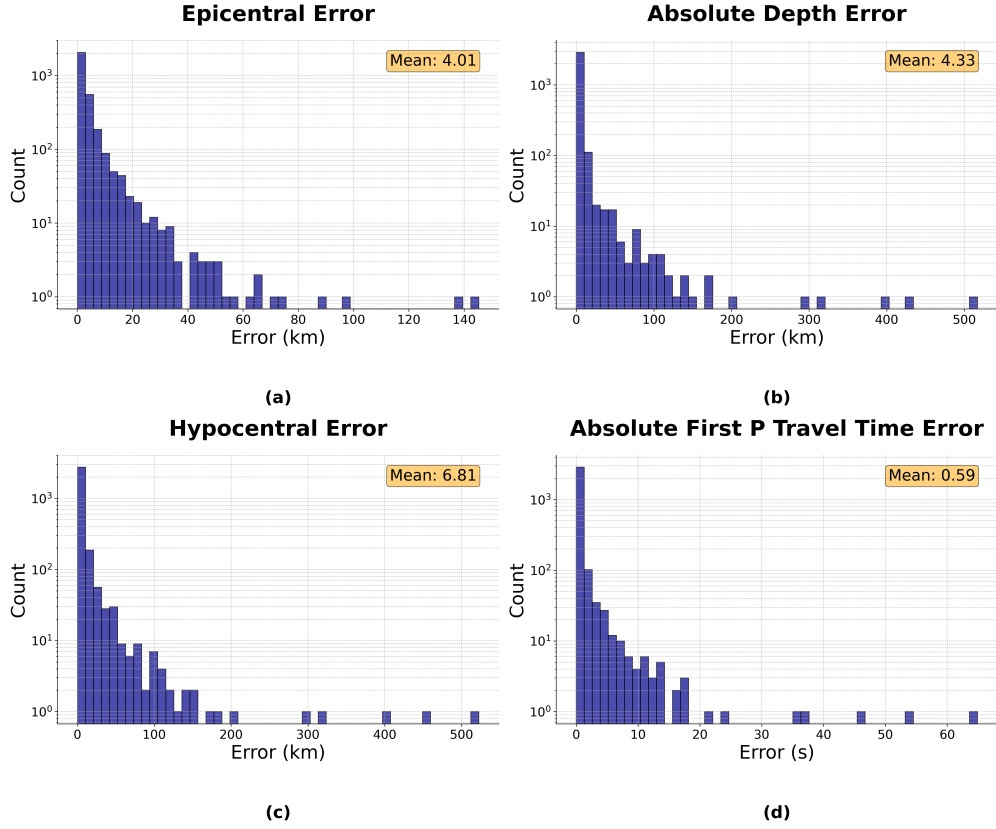


Figure C.1: Location Error Distributions on the Test Set

estimated and INGV bulletin values. The distribution is fairly symmetric with a median of 0 and an overall MAE of 0.2.

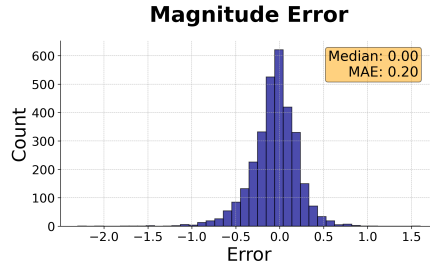


Figure C.2: Magnitude Error Distribution on the Test Set

### C.0.3 Model performance and explainable machine learning

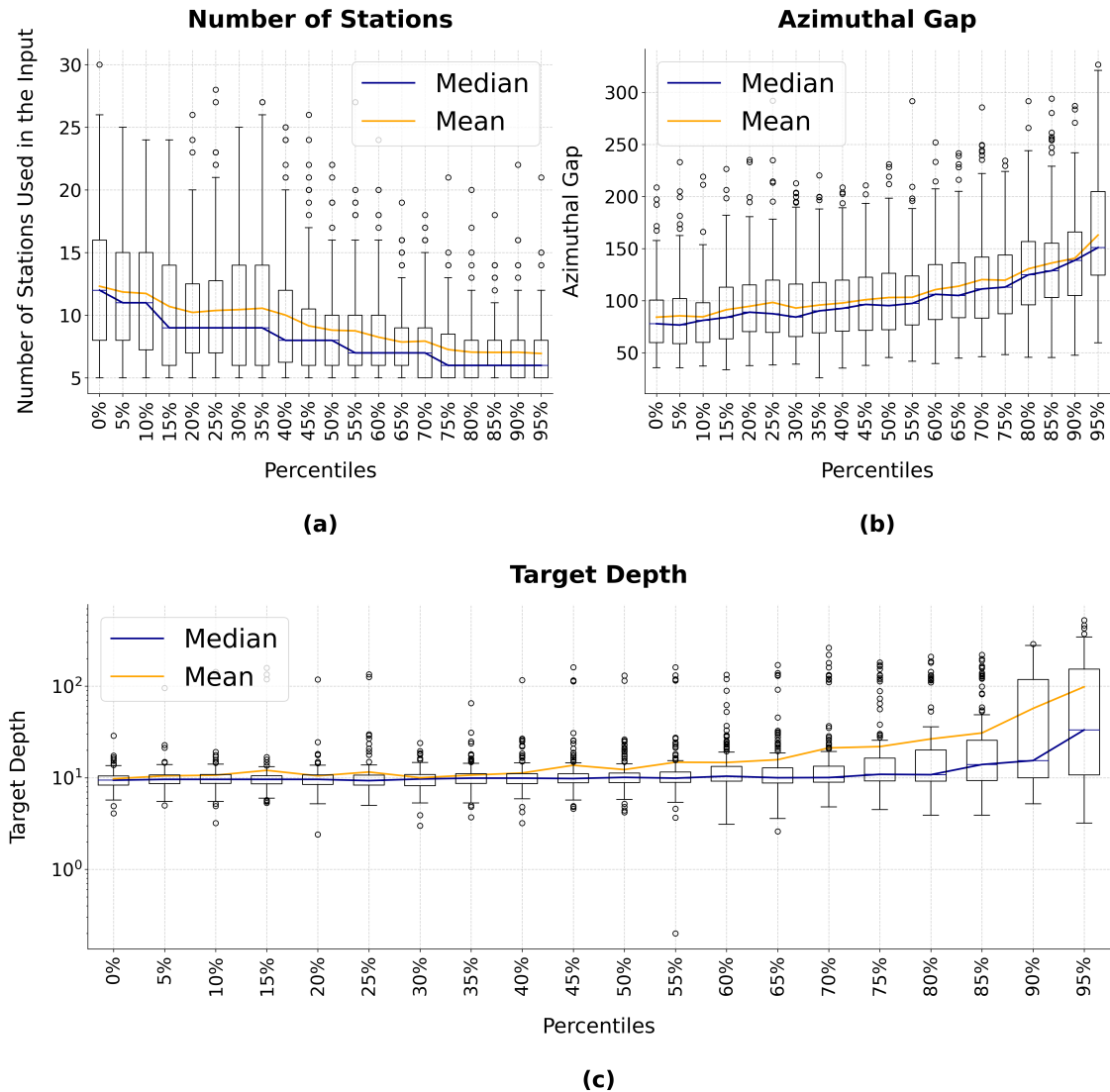
Interpreting the behavior of LLMs is inherently challenging due to their complexity [145], [146], [171]. Thus, we performed several experiments aimed at inferring what the model has learned and identifying its limitations. Specifically, we analyzed the impact of input features, such as station coverage and azimuthal gap, on errors, tested how the model responded to transposed input data to assess its internal understanding of P-wave velocities and also tested the influence of PGV values for location estimation. Our goal was to analyze key factors impacting model performance, shedding light on its strengths and weaknesses.

**Impact of input event features on epicentral error** Figure C.1 highlights the presence of



large outliers in both epicentral and depth estimates. While the latter is not surprising since the process of estimating the correct depth is usually more difficult even for experts, the former led us to wonder if there are some aspects that particularly influence the epicentral errors.

A preliminary observation is that the LLM estimates the epicenter and the depth together, leading to a possible process of balancing the overall hypocentral distance from stations by moving the epicenter, thus increasing its error, to compensate for the depth error. Moreover, in Figure 2 it can be noted that the larger errors are mostly located in the region between Sicily and Calabria, where the earthquakes are generally deeper, and in the sea, where the coverage of the stations is lower. The limited station coverage generally leads to a large azimuthal gap, which makes it more difficult to correctly locate the earthquake due to the lack of geometric constraints. For our model in particular, it also leads to a lower number of stations used for the prediction due to the  $\Delta T$  constraint. We further analyzed whether these parameters really affected the epicentral error with



**Figure C.3:** Location Error. On the x-axis there are the percentile of the left edge of the epicentral error interval taken in consideration C.1. Each interval contains 5% of the test set events. On the y-axis are the mean and median of, from left to right, the number of stations used in the input (a), the azimuthal gap (b) and the target depth (c).

different plots (Figure C.3), following the same scheme for all of them. We computed the percentiles



of the epicentral error at 5% intervals, and then plotted the epicentral error intervals, each containing 5% of the entire test set, on the x-axis so that the plots would be statistically significant (Table C.1). For example, the first interval will contain all test examples with an epicenter error in  $(0.08km, 0.43km)$ , while the last will have an error in  $(14.52km, 145.31km)$ , thus containing the outliers we are interested in. For each interval, we calculated the mean and median of each factor we wanted to examine, and we overlaid the box plot of the values in the bin to better visualize their distribution. Figure C.3(a) highlights that as the error interval increases, the average number of

Table C.1: Error Interval used in Fig.C.3

| X-Axis Label<br>Interval (km)<br>Interval (km) | Epicentral Error<br>X-Axis Label | Epicentral Error |                |
|------------------------------------------------|----------------------------------|------------------|----------------|
| 0%                                             | 0.08 – 0.43                      | 50%              | 2 – 2.24       |
| 5%                                             | 0.43 – 0.61                      | 55%              | 2.24 – 2.51    |
| 10%                                            | 0.61 – 0.78                      | 60%              | 2.51 – 2.84    |
| 15%                                            | 0.78 – 0.95                      | 65%              | 2.84 – 3.27    |
| 20%                                            | 0.95 – 1.1                       | 70%              | 3.27 – 3.85    |
| 25%                                            | 1.1 – 1.27                       | 75%              | 3.85 – 4.67    |
| 30%                                            | 1.27 – 1.42                      | 80%              | 4.67 – 5.97    |
| 35%                                            | 1.42 – 1.6                       | 85%              | 5.97 – 8.48    |
| 40%                                            | 1.6 – 1.78                       | 90%              | 8.48 – 14.52   |
| 45%                                            | 1.78 – 2                         | 95%              | 14.52 – 145.31 |

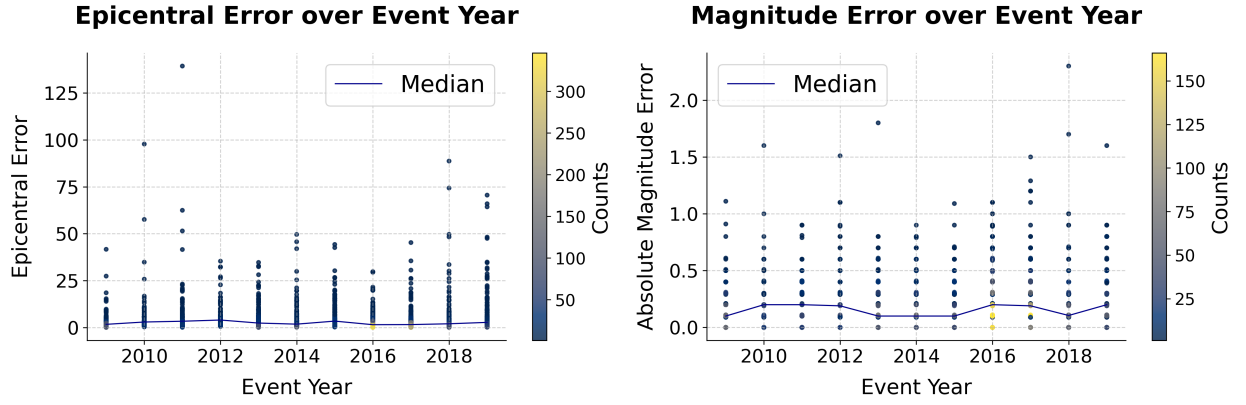
stations used as input decreases and the overall distribution within the bins becomes more concentrated, showing fewer examples of inputs with a high number of stations. Also for the azimuthal gap (Fig.C.3(b)), although less clear than before, a trend still seems to be present. More specifically, as the error increases, the azimuthal gap seems to grow on average and to have a more dispersed distribution with outliers up to more than 300 degrees.

In Fig.C.3(c) it is visible that the final target depth distributions, especially in the last interval, are significantly wider than the initial ones. In addition, in the last interval, which contains the outliers in the epicentral error, it can be seen that there are events with a true depth greater than  $300km$  that are not present in the other cases.

These plots seem to confirm how the epicentral error can be influenced by the three factors considered. Even though there are cases in the last error range where the number of stations is high, the event is shallow, or the azimuthal gap is small, it is likely that just one of these three values being an outlier will cause the model to incorrectly locate the earthquake.

**Relation between error and event year** We tested the idea that model performance increased inversely with occurrence date because more information is available for older events. TinyLlama was pre-trained using two publicly available datasets, namely SlimPajama [148] and StarCoder [93]. The former is a reduced version of RedPajama [174] derived from several open datasets covering a variety of sources such as Common Crawl, Wikipedia, and ArXiv. Information about events, especially the largest ones, could be included in these resources with a possible number of related data depending on the number of years that have passed since the event itself. We analyzed whether there is a relation between the year of the event and the epicentral or magnitude error. Figure C.4 highlights the lack of such a correlation, showing how the year of the event does not affect the model performance.

**P-waves Velocity** Traditional earthquake location methods use P-wave arrival times along with velocity models that describe how seismic wave velocities in the subsurface vary with position. The



**Figure C.4:** (a) shows the epicentral error as a function of the year of the event. (b) shows the absolute magnitude error as a function of the year of the event. In both cases, a blue line is overlaid representing the median of the errors for each year, and years with a number of events less than 100 have been excluded.

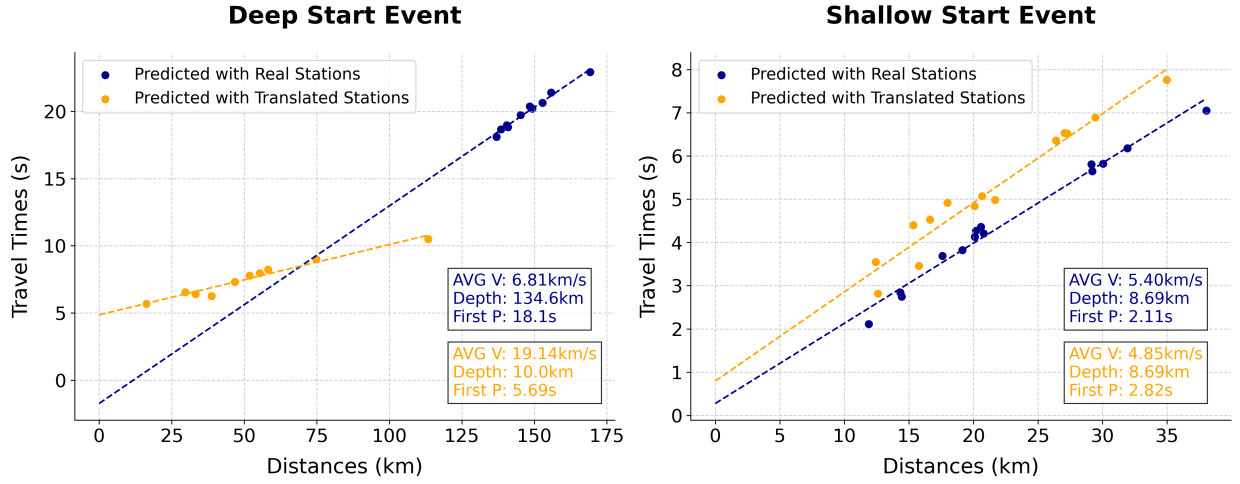
latter can be 1-dimensional [85] (varying only with depth) or 3-dimensional [18] (varying with total space position). Seismic velocity models depend on Earth structure and composition, so they can vary widely with location. However, note that our LLM model does not use a velocity model, but rather only the time differences between P-wave arrival times. The promising results on the test set and with the INGV comparison motivated us to test whether the fine-tuned LLM had learned velocity patterns during the fine-tuning process or the pre-training process.

For this analysis, we moved some events and their associated stations to other parts of Italy. Specifically, we translated the coordinates of the stations associated with each event by applying an offset to the latitude and longitude of all stations. During this process, we preserved the relative positions of the stations and used the UTM projection [87] to avoid problems related to meridian convergence and Earth’s shape as an oblate spheroid. All the other input parameters were kept unchanged.

Here, we show an example of the two types of results we obtained during this process. Specifically we observed different behaviors depending on event depths the LLM saw during the training phase in the real and fake regions. When we moved events to a region with similar earthquake depths the model predictions along with the resulting P-wave velocity seem reasonable (Fig. C.4(b)). But the results are different when we moved events to a region where the earthquakes were mainly deeper. An example is moving events from the area between Sicily and Calabria to central Italy. In this case the model predictions are aligned in terms of P-wave travel times and hypocentral distances but the average P-wave velocity is completely out of range (Fig. Supp. 5).

The conclusion of these tests is that the model does not learn underlying velocity models. We believe this could be an area for future work, where physical information could be incorporated into the model using approaches such as physics-informed deep learning frameworks. For example, studies like [165],[129], [131],[82] have demonstrated the potential of physics-informed neural networks in addressing seismological challenges by embedding domain knowledge directly into the training process. Incorporating such methods could improve the model’s ability to account for P-wave velocity patterns and enhance its predictive accuracy.

**Analysis of PGV Influence on Location Performance** The model proposed in this study performs the location and magnitude estimation at the same time by using in input parameters that are typically used separately for one or the other estimation. With this analysis we wanted to check



**Figure C.5:** Both plots show the comparison between model predictions when using the real stations coordinates and the transposed ones. (a) is related to an originally deep event located in the sea area between Sicily and Calabria and then transposed in central Italy. (b) is related to an originally shallow event located in central Italy and then transposed further north.

if and in what proportion the PGV values influence the location estimation. To do this, we run the model on the test set with all PGVs set to 0. Table C.2 presents the results, comparing them with the errors obtained when using the actual PGV values. It can be observed that the location estimates are slightly worse when PGV is not used. The errors increase by less than 15%, which could be attributed to the missing amplitude information (that may help the model in constraining the location), as well as to the unusual input values. This highlights the critical importance of the PGV input parameters for accurate magnitude estimation, which is entirely reasonable given their relevance to ground motion characteristics.

**Table C.2:** Test Results with zero PGVs

| Metric                      | Test with Real PGVs | Test with 0 PGVs | Error Increase Percentage |
|-----------------------------|---------------------|------------------|---------------------------|
| Mean Epicentral Error (km)  | 4.01                | 4.51             | 12%                       |
| Mean Hypocentral Error (km) | 6.81                | 7.73             | 14%                       |
| MAE Depth (km)              | 4.33                | 4.95             | 14%                       |
| MAE Origin Time (s)         | 0.59                | 0.68             | 15%                       |
| MAE Magnitude               | 0.2                 | 0.86             | 330%                      |
| RMSE Magnitude              | 0.28                | 1.05             | 275%                      |

## Appendix D

# Supplementary Material for Chapter 5

### D.0.1 SeisCLIP

The model SeisCLIP [144] is inspired by CLIP [128], one of the most popular FMs for multimodal image and video data. The SeisCLIP design consists of two encoders, a transformer encoder the short-time Fourier transform (STFT) of the input seismic waveform and a multi-layer perceptron (MLP) encoder integrates the phase and source information of the same event. The transformer encoder is a pre-trained model developed with the ImageNet dataset that extracts features from time-frequency seismic spectrum. The encoders are jointly trained using the STEAD dataset using contrastive learning on multimodal data. The transformer encoder is subsequently fine-tuned for various downstream tasks.

SeisCLIP is evaluated for: (a) single-station event classification. Six different models are trained holding one class label as positive and all others as negative for earthquake, explosion, surface event, sonic boom, thunder and plane crash. (b) Localization defined as the distance of the earthquake from the station. And (c) focal mechanism analysis using a classification setup for a model to identify the type of mechanisms as reverse, strike-slip, or normal labels, without predicting the strike, rake, and dip information to constrain the orientation. These tasks employ distinct datasets from different regions to train and evaluate the different models.

Three distinct training strategies are adopted for evaluation: (1) fine-tune by further training the spectrum encoder, (2) freeze the model by not allowing the encoder weights to update, and (3) train from scratch by not employing a pre-trained model. In all tasks the models perform well and the generalization capabilities of the fine-tuned model is found to be the best.

### D.0.2 SeisT

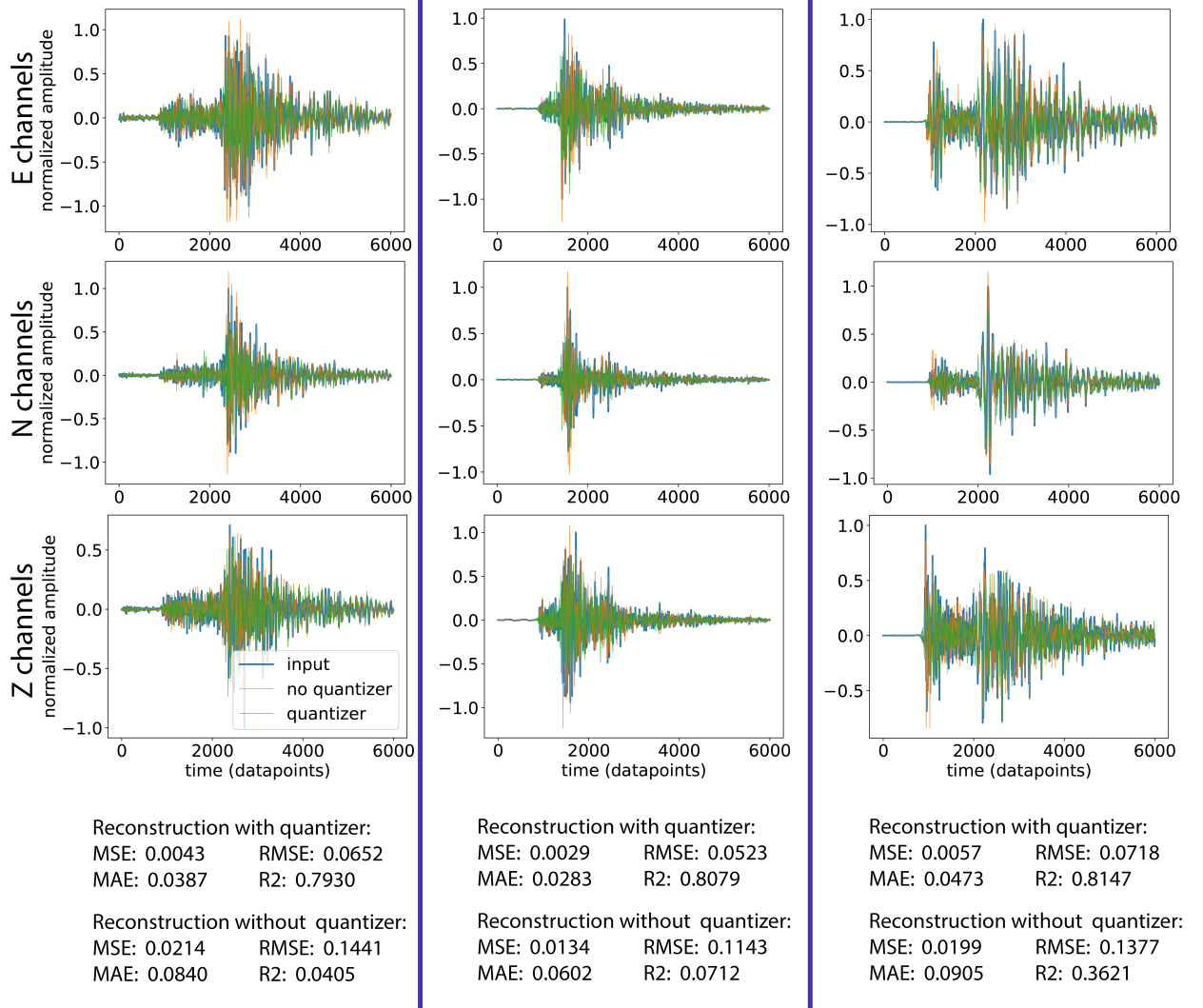
The model Seismogram Transformer (SeisT) [?] is a FM4S that combines CNNs for local feature extraction with transformers for capturing interactions between global and local features. It incorporates Multi-Scale Mixed Convolution (MSMC) modules, Local Aggregation (LA) modules, and Multi-Path Transformer (MPT) modules. It has various output head modules to make the model suitable for various tasks. The first type of output head is composed of multiple consecutive up-sampling stages, and it is designed for earthquake detection and seismic phase picking. The second type of output head is designed for classification tasks like first-motion polarity. The third type of output head is similar to the classification head, but the output layer is a sigmoid and is

employed for regression tasks estimating the back-azimuth, magnitude, and epicentral distance. The model inputs data from multiple stations in the network. They change the tokenization procedure in the Transformer, by aggregating information from multiple nearby stations recording, to facilitate interactions between local features (replacing point-to-point feature interactions). To meet the requirements of different application scenarios and varying training data volumes, SeisT is designed in three versions, based on its number of layers: Small (S), Medium (M), and Large (L). Model training is performed using seismic data from Chinese seismic networks and the performance is evaluated on records from China and the Northwestern Pacific region of the United States, showing good generalization capabilities in the latter. The model results for SeisT outperform or meet the state-of-the-art models. The most significant improvements when comparing to existing models are in seismic phase picking (+1.7% for P-waves and +9.5% for S-waves) and magnitude estimation (+8.0%). The model can perform earthquake detection, seismic phase picking, first-motion polarity classification, magnitude estimation, back-azimuth estimation, and epicentral distance estimation thanks to the multi-output head architecture, and it is trained jointly for all the tasks.

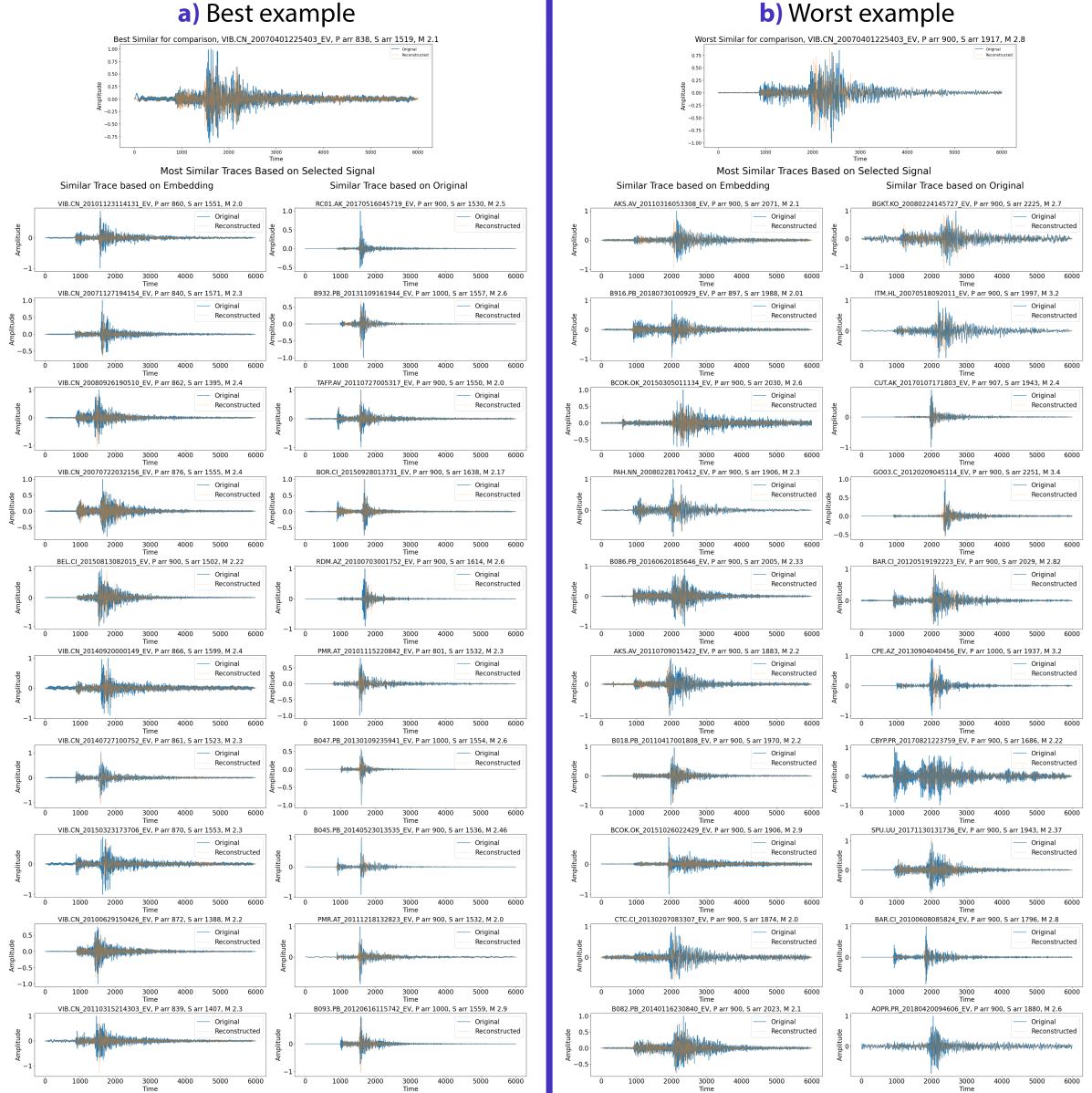
### D.0.3 SeisLM

The model Seismic Language Model (SeisLM) [95] is an encoder-only Transformer that focuses on the task of predicting features of masked timesteps. The architecture follows Wav2Vec2 for speech signal modeling and BERT for text modeling. It consists of a CNN, a quantizer, and transformer. The inputs are 3-channel raw seismic waveforms and it is pre-trained on a large collection of open-source unlabeled seismic datasets using a self-supervised contrastive loss. At each forward pass of pertaining, a portion of the convolutional features is randomly replaced by a shared trainable feature vector. For each masked time step, the pre-training goal is to identify the correct quantized latent representation from a candidate set. The pertaining is conducted on a global compilations of 8 seismic datasets from the SeisBench framework. They pre-trained two variants of models: SeisLM-base and SeisLM-large.

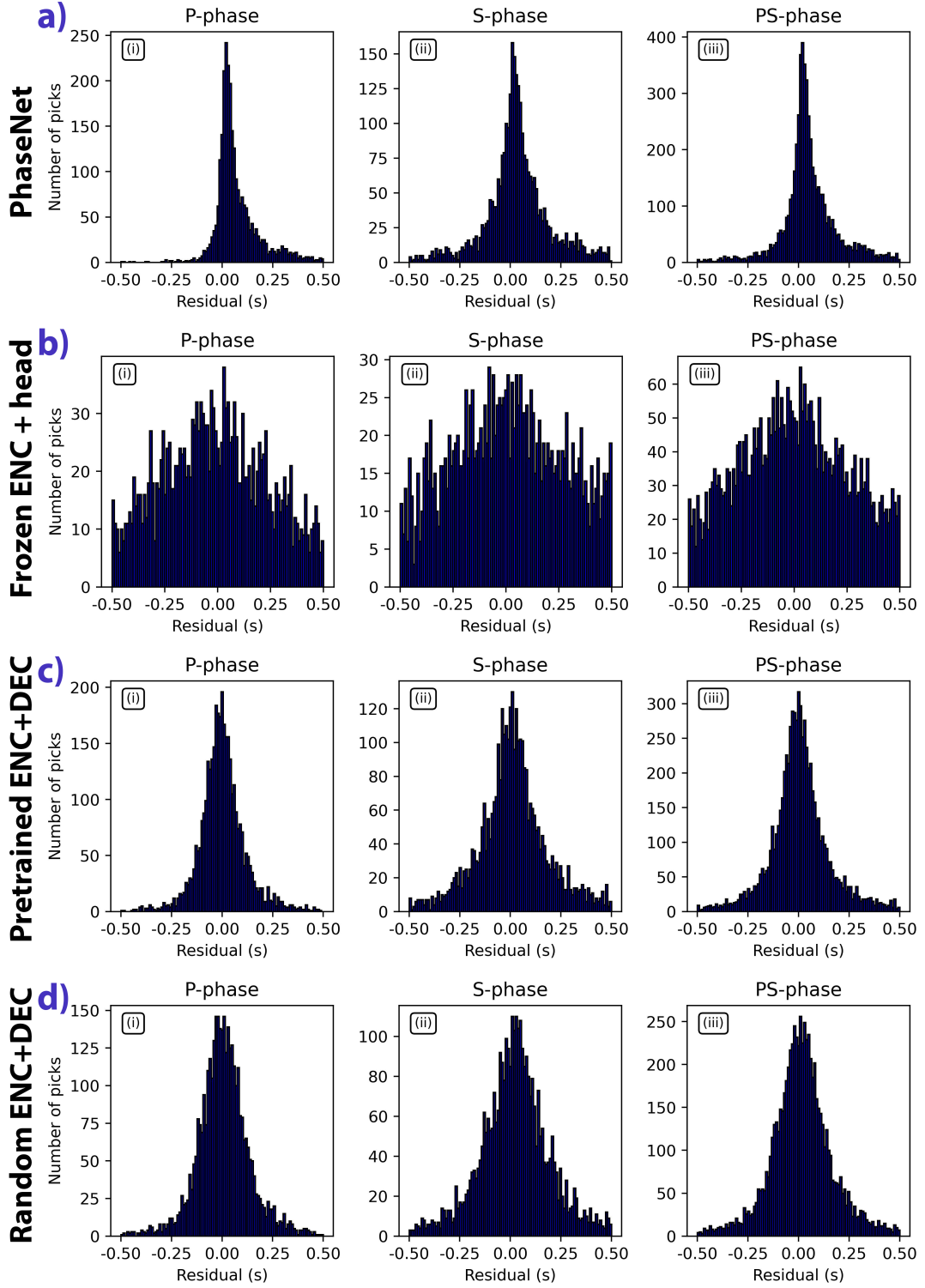
Four evaluation tasks are performed for a quantitative analysis: event detection, phase identification, onset regression and foreshock–aftershock classification task. To fine-tune the pre-trained models a randomly initialized shallow network is added to process the output of SeisLM. During fine-tuning the parameters of both the SeisLM and the task head are trained. SeisLM exceeds the baseline in all tasks, moreover they show that consistent improvements are not observed when freezing parts of the pre-trained model or using a scheduler to unfreeze it gradually.



**Figure D.1:** Three examples of waveform reconstruction from STEAD. The trace names in the dataset are (left to right) "AMT.HP\_20120416124803\_EV", "KHIT.AK\_20140110013613\_EV", "AKGG.AV\_20111028181955\_EV". The input signal is shown in blue, in orange is the reconstruction without the quantizer, and in green is the reconstruction with the quantizer. The listed metrics are mean-squared-error (MSE, the lower the better), root-mean-squared-error (RMSE), mean-absolute-error (MAE, the lower the better), and correlation (R2, the higher the better) provide quantitative reconstruction quality measures.

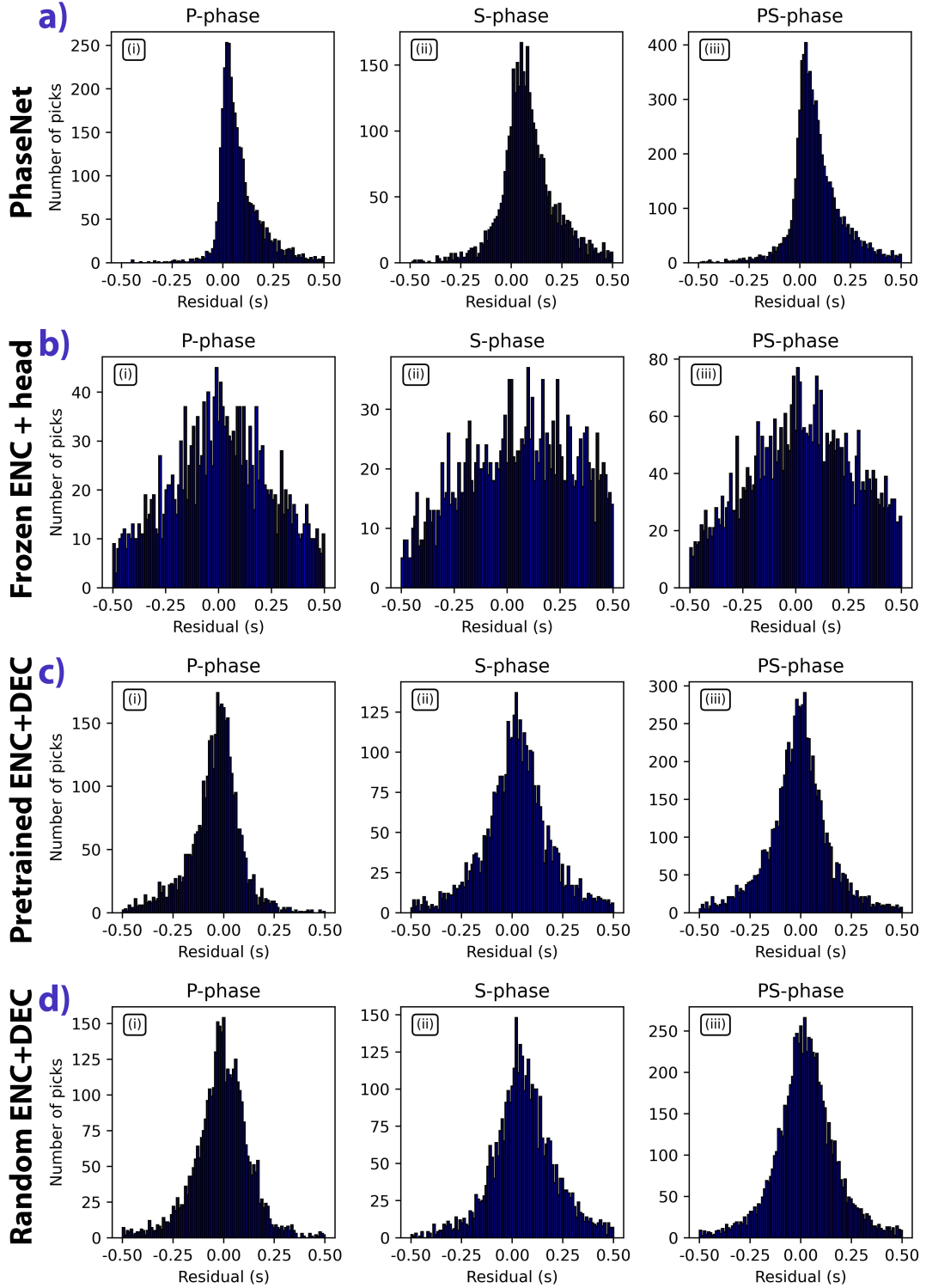


**Figure D.2:** Two examples of similar traces retrieved based on embedded and original seismic waveform. Here we display the best (panel a) and worst (panel b) examples with the 10 most similar traces based on the difference between the selected P arrival, S arrival and Magnitude, versus the other detected similar traces. Only channel E is presented. The "Best example" shows a very close match between the target trace and the retrieved similar trace, preserving important seismic features like P and S wave arrivals and overall envelope shape. The "worst example" still shows similarity in overall shape, but with more noticeable differences in timing and peaks.

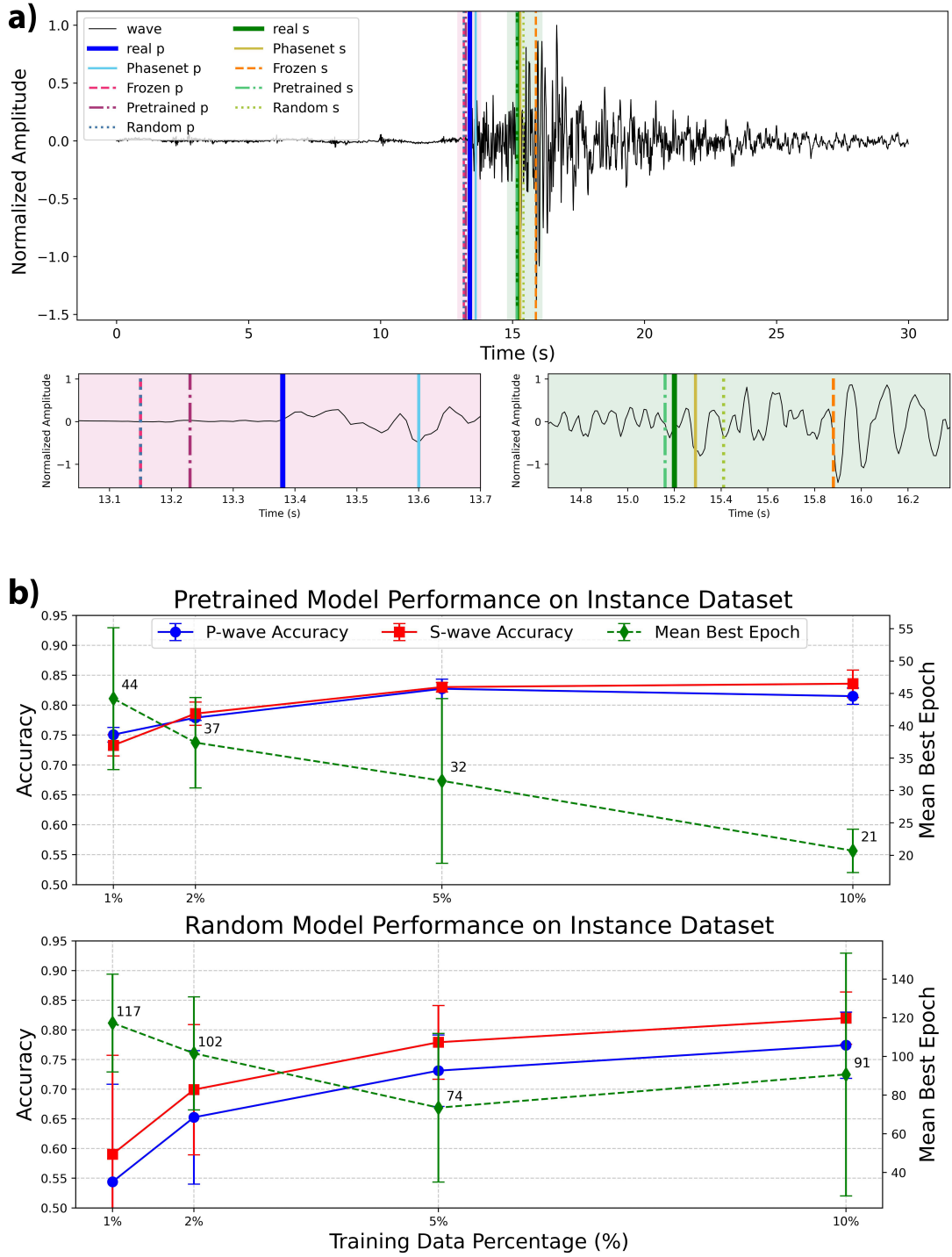


**Figure D.3:** Distribution of residuals ( $\Delta t$ ) of the tested models on STEAD dataset. Panel (a) PhaseNet. P-arrival: mean=0.070797, std=0.110022; S-arrival: mean=0.051755, std=0.138495. Panel (b) Frozen ENC + head. P-arrival: mean=-0.013342, std=0.247148; S-arrival: mean=-0.002217, std=0.253981. Panel (c) Pretrained ENC + DEC. P-arrival: mean=0.000846, std=0.115684; S-arrival: mean=0.005798, std=0.145941. Panel (d) Random ENC + DEC. P-arrival: mean=0.007378, std=0.129801; S-arrival: mean=0.014513, std=0.151914.

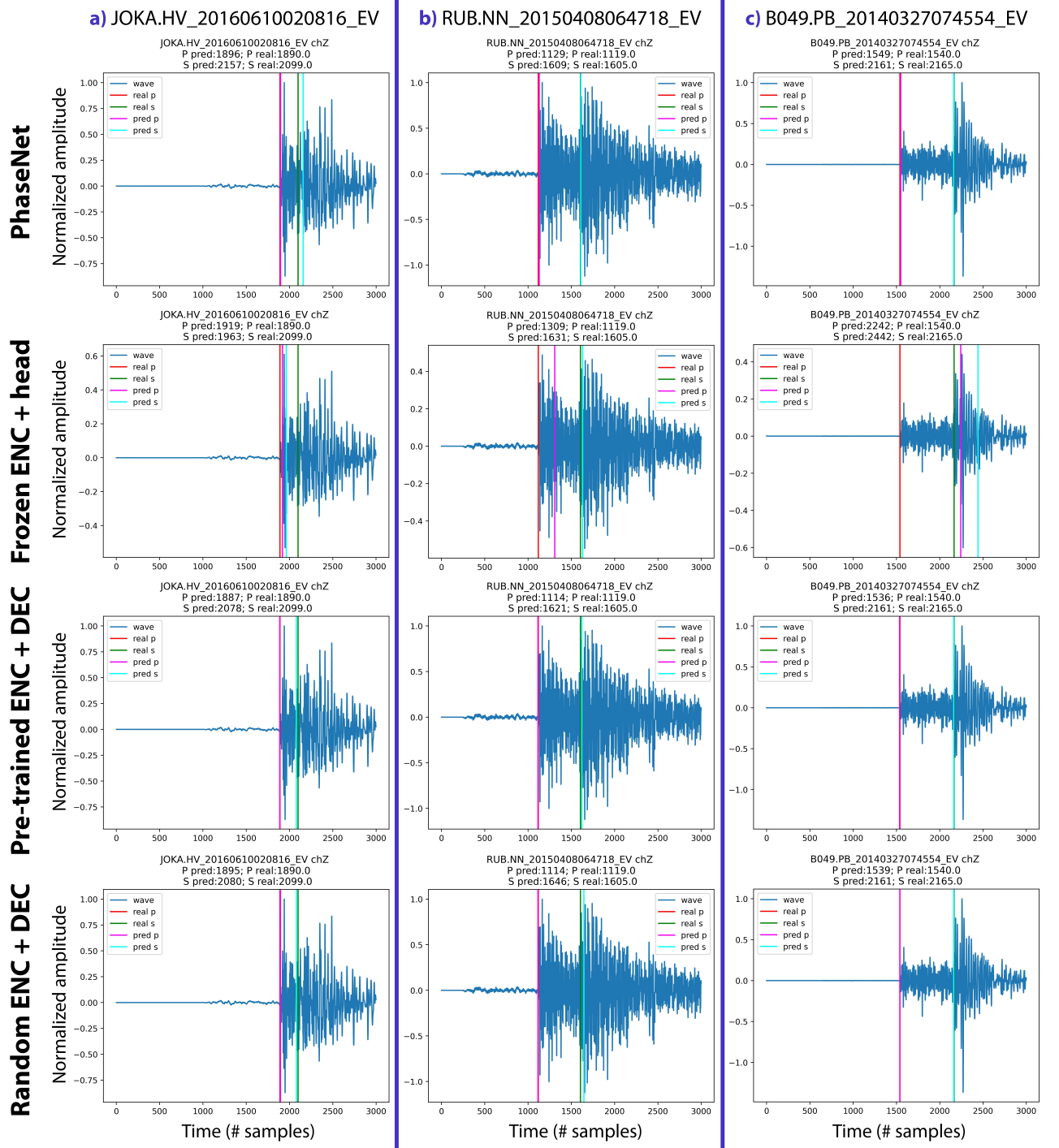




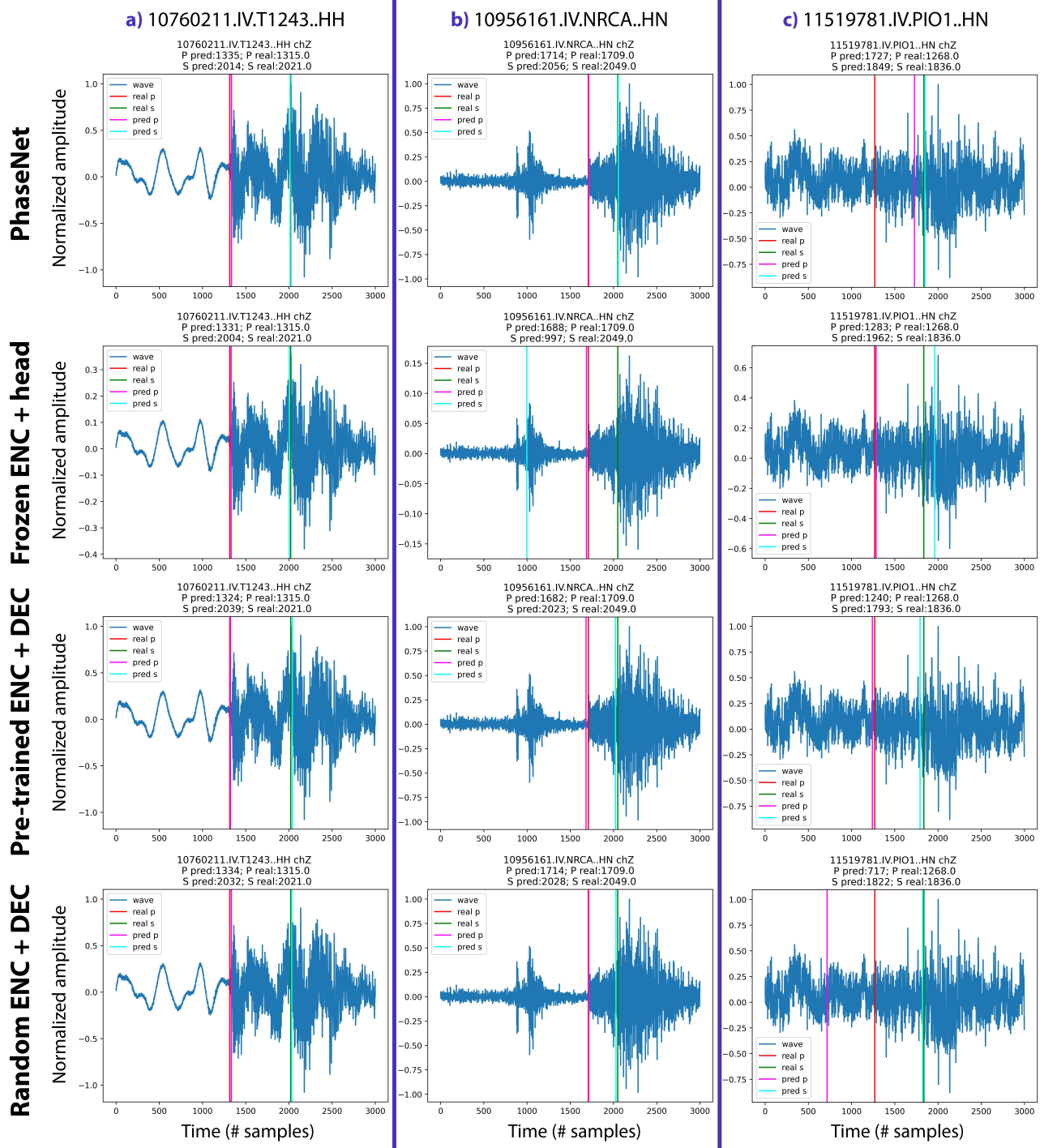
**Figure D.4:** Distribution of residuals ( $\Delta t$ ) of the tested models on the INSTANCE dataset, to evaluate generalizability. Panel (a) PhaseNet. P-arrival: mean=0.082230, std=0.105384; S-arrival: mean=0.082885, std=0.123061. Panel (b) Frozen ENC + head. P-arrival: mean=0.010321, std=0.236151; S-arrival: mean=0.030189, std=0.247748. Panel (c) Pre-trained ENC + DEC. P-arrival: mean=-0.044369, std=0.128707; S-arrival: mean=-0.010979, std=0.153109. Panel (d) Random ENC + DEC. P-arrival: mean=-0.010560, std=0.136742; S-arrival: mean=0.023156, std=0.151109.



**Figure D.5:** In panel (a), we present a detailed qualitative comparison among models, we randomly selected one example from the test set of the INSTANCE dataset. We display here the Z channel of the waveform with trace name “10233131.IV.T1256.HH”. The input signal is shown in black, the vertical lines are the real or predicted phases for each tested model. We also zoom around the P (highlighted in pink) and S (highlighted in green) arrival, for a better visual analysis. In panel (b), we compare the pretrained and random models in a few shot scenario. Boxplots refer to 10 runs for each configuration. 10 in total for each configuration.



**Figure D.6:** Three qualitative examples to compare the performance of the analyzed models. Rows are the four compared models, columns are three examples (trace names). We only display the Z channel of three randomly selected earthquakes from the test set of the STEAD dataset.



**Figure D.7:** Three qualitative examples to compare the performance of the analyzed models. Rows are the four compared models, columns are three examples (trace names). We only display the Z channel of three randomly selected earthquakes from the INSTANCE dataset, to evaluate generalizability.

## Acknowledgements

I am grateful to all the colleagues and collaborators who supported this work and contributed to fruitful discussions throughout the PhD. I would like to acknowledge the direct contributions of: Laura Laurenti, who collaborated with me on Chapters 2 and 5; Aurora Bassani, who collaborated on Chapters 4 and 7; Giulio Poggiali, who collaborated on Chapters 2 and 4; Ilaria Oliveti, who collaborated on Chapter 3 and Alessandro Scifoni, who collaborated on Chapter 7. Their input and shared efforts substantially improved the quality and scope of the thesis.

Although not formally appointed as a co-advisor, Chris Marone provided guidance and feedback throughout my doctoral studies that effectively amounted to that of a third co-advisor. I am sincerely grateful for his insightful advice and support.

I also wish to thank all my colleagues at the Perception and Intelligence Laboratory (PinLab), the entire ERC TECTONIC group, and all the researchers with whom I collaborated at INGV for their support, discussions, and a stimulating working environment.

This research was supported by the INGV Pianeta Dinamico 2021 Tema 8 SOME project (CUP D53J1900017001) funded by the Italian Ministry of University and Research, and by the European Research Council (ERC) under grant 835012 (TECTONIC).

Daniele Trappolini, Rome, 25/10/2025

