WaveCastNet: An AI-enabled Wavefield Forecasting Framework for Earthquake Early Warning

Dongwei Lyu[1,2]   Rie Nakata[2,3,4*]   Pu Ren[3]   Michael W. Mahoney[2,3,5]
Arben Pitarka[6]   Nori Nakata[2,3,7]   N. Benjamin Erichson[2,3*]

[1]Department of Mathematics, UC Berkeley
[2]International Computer Science Institute
[3]Lawrence Berkeley National Laboratory
[4]Earthquake Research Institute, University of Tokyo
[5]Department of Statistics, UC Berkeley
[6]Lawrence Livermore National Laboratory
[7]Massachusetts Institute of Technology

Abstract

Large earthquakes can be destructive and quickly wreak havoc on a landscape. To mitigate immediate threats, early warning systems have been developed to alert residents, emergency responders, and critical infrastructure operators seconds to a minute before seismic waves arrive. These warnings provide time to take precautions and prevent damage. The success of these systems relies on fast, accurate predictions of ground motion intensities, which is challenging due to the complex physics of earthquakes, wave propagation, and their intricate spatial and temporal interactions. To improve early warning, we propose a novel AI-enabled framework, WaveCastNet, for forecasting ground motions from large earthquakes. WaveCastNet integrates a novel convolutional Long Expressive Memory (ConvLEM) model into a sequence to sequence (seq2seq) forecasting framework to model long-term dependencies and multi-scale patterns in both space and time. WaveCastNet, which shares weights across spatial and temporal dimensions, requires fewer parameters compared to more resource-intensive models like transformers and thus, in turn, reduces inference times. Importantly, WaveCastNet also generalizes better than transformer-based models to different seismic scenarios, including to more rare and critical situations with higher magnitude earthquakes. Our results using simulated data from the San Francisco Bay Area demonstrate the capability to rapidly predict the intensity and timing of destructive ground motions. Importantly, our proposed approach does not require estimating earthquake magnitudes and epicenters, which are prone to errors using conventional approaches; nor does it require empirical ground motion models, which fail to capture strongly heterogeneous wave propagation effects.

1 Introduction

Large earthquakes can rapidly devastate landscapes, toppling buildings and rupturing infrastructure, posing a substantial risk in seismically active regions. These seismic events happen when a fault ruptures. The released seismic energy propagates through the Earth in form of seismic waves, eventually reaching the Earth's surface. To mitigate the immediate threats posed by large earthquakes, early warning systems have been developed and implemented [4, 33, 34]. These systems aim to detect fast-traveling P-waves by sensors located in proximity to the earthquake epicenter. Once detected, a processing center estimates the earthquake location, magnitude (M), and fault geometry. Then, the system predicts ground motion intensity parameters (e.g., Modified Mercalli Intensity, Peak Ground Acceleration, and Peak Ground Velocities), which provide information regarding potential damages. Subsequently, warnings are issued, typically a few seconds to a minute before the arrival of the more destructive S-waves and surface waves. These warnings serve as an early alert to enable critical infrastructures to initiate necessary precautions, such as stopping trains and shutting down gas pipelines, which allow people to take protective measures.

The performance of these systems relies on the detection and isolation of earthquake signals, as well as on the accuracy of the earthquake parameter estimation and seismic wave propagation modeling [3]. Inaccuracies in the parameter estimation, most commonly in over/under predictions in earthquake magnitudes, lead to false alert or missing warning opportunities [55, 42]. The conventional use of empirical ground motion models precludes high fidelity representation of the complex source and path effects, and the site-specific variability of ground motion intensities [26, 8, 9, 14, 6]. Alternative approaches forecast future ground motion intensity measures or waveforms up to the time when a sensor detects actual earthquake ground motions [31, 21]. These approaches combine physics-based simulation (e.g., radiative transfer theory or finite-difference wavefield simulations) and data assimilation (e.g., optimum interpolation techniques [32]) to remove the dependence on arrival detection and magnitude estimation, while handling the sparsity of the data and incorporating source and path effects. However, typically their prediction accuracy remains insufficient to be deployed in real cases [3], and they require substantial computational resources [21].

Artificial Intelligence (AI) provides a promising alternative approach for modeling ground motion propagation. That is because deep neural networks are well posed to model the nontrivial spatiotemporal properties of ground motions [19,
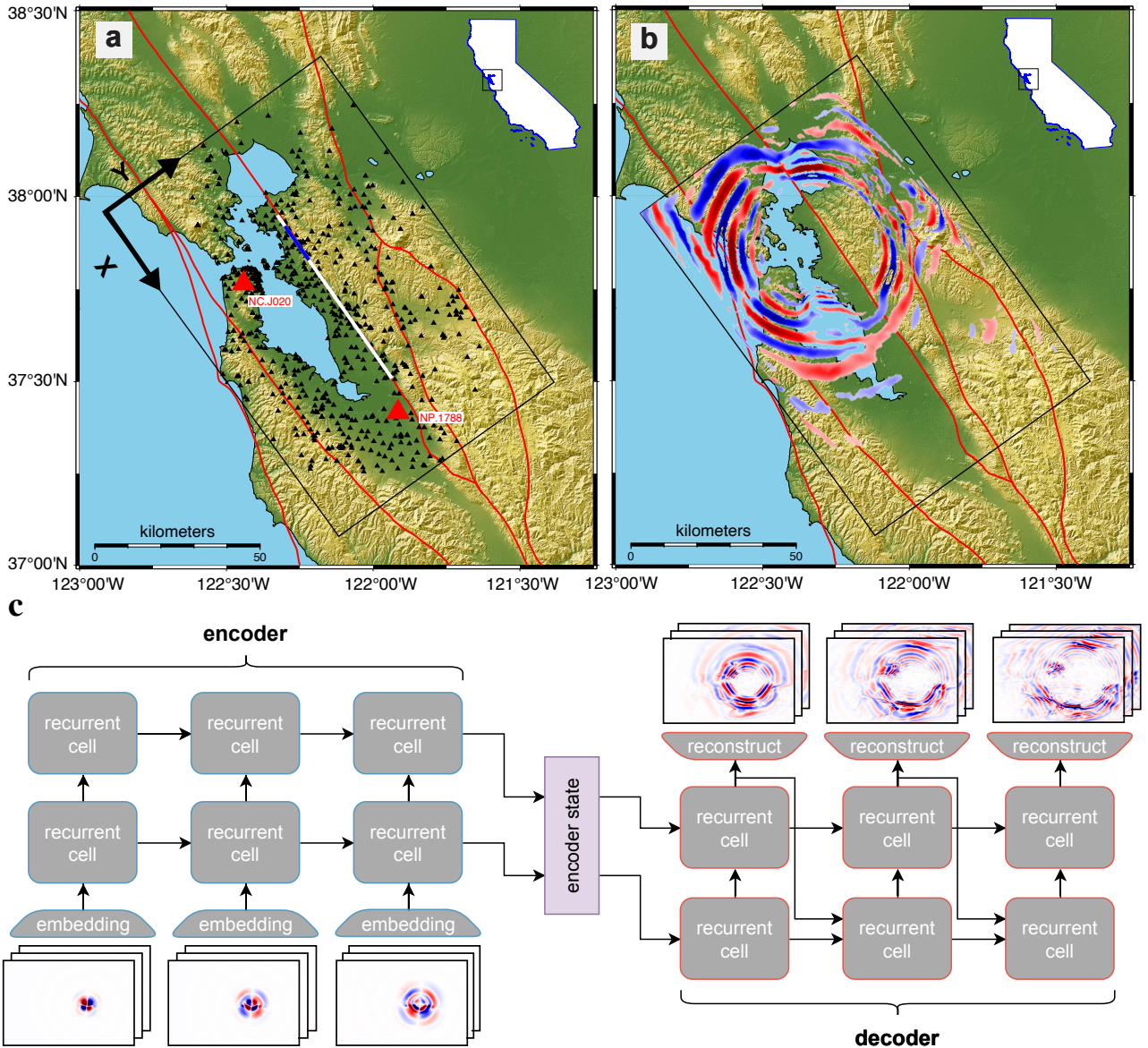

Figure 1: Illustration of the problem setup and our proposed WaveCastNet model. In (a), the simulation area of interest within the San Francisco Bay Area, highlighted by the black rectangular box, is shown. Point-source earthquakes are placed along the thick white line, and an M6 earthquake rupture plane is indicated by the blue line. The red lines indicate known faults, the black triangles show the actual sensor locations, and the red triangles highlight the two sensor stations used in the discussion below. In (b), an example snapshot of visco-elastic wave propagation from the point-source earthquake at T=21.79 seconds is shown. In (c), an illustration of using WaveCastNet to forecast the propagation of seismic waves is shown. The framework consists of encoder and decoder components, which in turn consist of stacked recurrent cells. In this work, we advocate a novel recurrent ConvLEM cell, which can model multiscale structures in space and time.

61, 11, 60, 20, 22, 57]. Moreover, AI methods have the advantage of being computational efficient during inference time, which is of great importance for early warning systems.

Figure 1(a-b) illustrates our problem setup alongside an example snapshot demonstrating visco-elastic wave propagation. Our objective is to predict future wave motions over a time horizon of up to 100 seconds. We approach this as a spatio-temporal sequence prediction task. Specifically, we are given a sequence of J elements, $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_J$, and our goal is to forecast the subsequent K elements, $\mathcal{X}_{J+1}, \mathcal{X}_{J+2}, \ldots, \mathcal{X}_{J+K}$. Each element $\mathcal{X}_t$ within the sequence belongs to $\mathbb{R}^{C \times H \times W}$, representing a 3-D seismic wavefield. Each wavefield provides spatial information, for

a $\mathbb{R}^{H \times W}$ grid, about the particle velocity of the wave propagation across C spatial directions (i.e., X, Y, Z directions). One of the main challenges in modeling ground motion data lies in the necessity to handle multi-scale structures that are complex to model. Thus, it is crucial for a forecasting model to effectively capture the joint correlations present across both spatial and temporal dimensions.

To address this challenge, we propose an AI-enabled framework for forecasting ground motions. Specifically, we develop a wavefield forecasting network (WaveCastNet), which is based on the sequence-to-sequence (seq2seq) framework introduced by [54]. Central to WaveCastNet are two components: an encoder and a decoder. The encoder processes

a sequence of seismic wavefields with the aim to summarizes the input sequence into a single encoder state. The decoder, in turn, generates a target sequence of seismic wavefields which is conditioned on the encoder state. Figure 1(c) illustrates the architecture of our WaveCastNet for predicting seismic waves. Within this architecture, both the encoder and decoder are composed of stacked recurrent units designed to model sequential data. There exist various formulations of modern recurrent cells, including unitary recurrent units [5], gated recurrent units (GRUs)[13], and long-short-term memory (LSTM) units[30]. However, these recurrent cells, with their reliance on fully-connected layers, tend to destroy the intrinsic multi-scale spatial information present in 2-D or 3-D spatial data. The Convolutional Long Short-Term Memory (ConvLSTM) architecture [52] addresses this shortcoming by integrating convolution operations into the LSTM's update and gating mechanisms, a modification that has proven particularly beneficial for modelling spatio-temporal sequences. ConvLSTM can effectively capture multiscale spatial patterns through convolutional filters, however, this model falls short in modeling temporal multiscale structures. To address this shortcoming, we design a novel convolutional long expressive memory (ConvLEM) model, which extends the LEM model [50] by integrating convolutional layers into the LEM architecture. This ConvLEM model is used as backbone for designing the WaveCastNet's encoder and decoder.

Our results demonstrate that WaveCastNet improves the predictive accuracy compared to seq2seq frameworks that leverage ConvLSTM, or gated variants. Our WaveCastNet even surpasses the capabilities of newly introduced transformer networks in the context of ground motion forecasting. Importantly, our approach shows robustness and enhanced generalization capabilities, especially in scenarios involving wavefields of greater magnitudes unseen during the training phase. The versatility of our framework is further evidenced by its flexibility, transitioning seamlessly from scenarios with dense, fully captured wavefields to those characterized by sparse, selectively sampled measurements. Expanding on these findings, we show that we can use an ensemble of WaveCastNets to produce uncertainty estimates. This is a critical component for demonstrating and verifying the reliability of our proposed framework. Our work not only showcases WaveCastNet's improved forecasting accuracy, but it also shows its potential for improving warning times and thus advancing early warning systems.

2   Results

We evaluate our methodology by forecasting particle-velocity waveforms near the Hayward fault, simulating earthquake scenarios in the San Francisco Bay Area (SFBA), northern California, United States, as depicted in Figure 1(a-b). San Francisco, positioned approximately 20 km west of the Hayward fault, ranks among the most densely populated metropolitan regions in the United States. Given the heightened seismic risk associated with the Hayward fault — estimated by the United States Geological Survey (USGS) to exceed 30% — the enhancement of early warning sys-

tems is imperative for minimizing infrastructural damage and disruptions, as well as reducing human casualties. Our work focuses on the prediction of ground motion waveforms extending over 120 km along the X direction, parallel to, and 80 km along the Y direction, perpendicular to, the Hayward fault, as outlined by the black rectangle in Figure 1(a-b).

Metrics.   The performance of WaveCastNet is assessed by analyzing the intensity of the ground motions using peak ground velocity (PGV) values, which are defined as:

$$PGV(\mathcal{X}) = \max_t \sqrt{\mathcal{X}_t^2[c_X] + \mathcal{X}_t^2[c_Y]}, \qquad (1)$$

where $\mathcal{X}_t^2[c_X]$ and $\mathcal{X}_t^2[c_Y]$ represent the velocity data in the X and Y directions, respectively. Additionally, we examine the corresponding arrival time, T_{pgv}, determined by the equation:

$$T_{pgv}(\mathcal{X}) = \arg\max_t \sqrt{\mathcal{X}_t^2[c_X] + \mathcal{X}_t^2[c_Y]}, \qquad (2)$$

indicating the moment when the horizontal amplitude of the particle velocity reaches its peak.

Furthermore, to evaluate the accuracy of the predicted wavefield $\hat{\mathcal{X}}$ against the target ground truth $\mathcal{X}$, we use the accuracy (ACC) metric, expressed as:

$$ACC = \frac{\sum_{t,h,w} \hat{\mathcal{X}}_t[c,h,w] \cdot \mathcal{X}_t[c,h,w]}{\sqrt{\left(\sum_{t,h,w} \hat{\mathcal{X}}_t^2[c,h,w]\right) \cdot \left(\sum_{t,h,w} \mathcal{X}_t^2[c,h,w]\right)}},$$

and the relative Frobenius norm error (RFNE), defined as:

$$RFNE = \frac{\sqrt{\sum_{t,h,w} \left(\hat{\mathcal{X}}_t[c,h,w] - \mathcal{X}_t[c,h,w]\right)^2}}{\sqrt{\sum_{t,h,w} \mathcal{X}_t^2[c,h,w]}}.$$

2.1   Point-source Small Earthquakes

We first use WaveCastNet to predict ground motions from point-source earthquakes with magnitudes smaller than M4.5. The training dataset is generated using simulated waveforms at frequencies below 0.5 Hz, with a minimum S-wave velocity of 500 m/s. A total of 960 point sources are positioned at 1 km intervals along the white line shown in Figure 1a, with sources placed at depths between 2 and 15 km (note, the white line represents a rectangular plane extending 60 km horizontally and 13 km vertically). These simulations use a fourth-order finite-difference visco-elastic wave model provided by the open-source SW4 package [46, 47]. The subsurface elastic properties are derived from the San Francisco Bay region 3D seismic velocity model v21.1, developed by the USGS [1, 29]. The source wavelet, modeled as a delta function low-pass filtered at 0.5 Hz, assumes that the corner frequencies of small earthquakes exceed 0.5 Hz, maintaining relatively flat frequency spectra within our simulation bandwidth. A uniform double-couple source mechanism is used for all simulations. These simulated data are used as the ground truth throughout our study.

Our goal with WaveCastNet is to generate forecasts for future 100-second intervals based on data observed during the
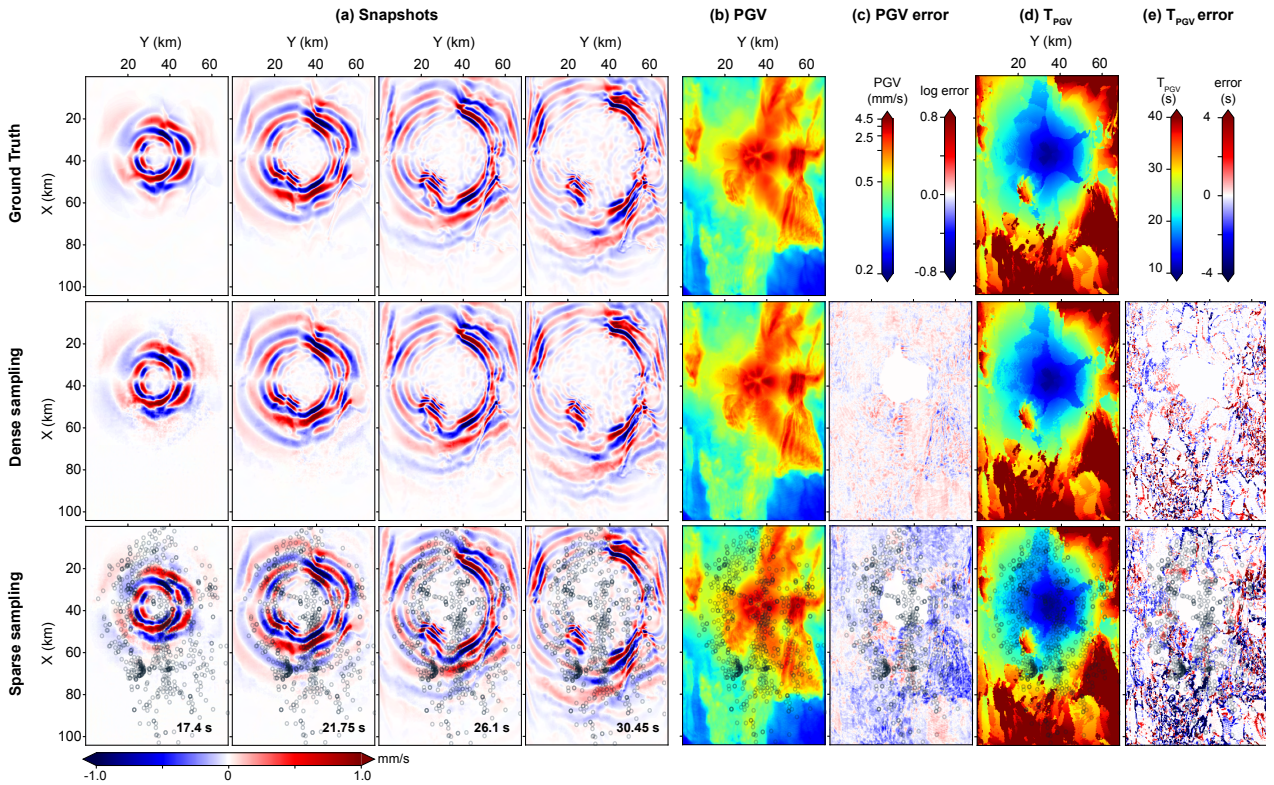
Figure 2: Point-source earthquake prediction: (a) Point-source earthquake wavefield snapshots for the Y component data at T=17.4, 21.75, 26.1, and 30.45 s from (top) ground truth, (middle) densely and regularly sampled predicted data, and (bottom) sparsely and irregularly sampled predicted data. (b) Map of PGV values, and (c) corresponding prediction errors; (d) map of T_{PGV}, and (e) corresponding prediction errors. The errors are calculated by subtracting the ground truth values from the predicted values; thus, positive values indicate over-prediction and negative values indicate under-prediction. Circles in the bottom figures indicate the location of stations.

initial 15.6 seconds post-rupture. The input sequence, denoted as $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_J$, comprises 60 elements, while the target sequence, $\mathcal{X}_{J+1}, \mathcal{X}_{J+2}, \ldots, \mathcal{X}_{J+K}$, includes 388 elements, with each timestep $\Delta t = 0.26$ seconds. The encoder component of WaveCastNet processes the input sequence to derive an encoding state, which subsequently is used by the decoder in generating the target sequence. Instead of forecasting the complete target sequence in one go, we consider iterative predictions of smaller subsequences, each spanning 60 elements. Specifically, we use the predicted subsequences as new inputs to forecast the next 60 elements, and so on. We obtain the entire target sequence in under one second.

In the following, we consider two scenarios: (i) input sequences consisting of densely sampled wavefields; and (ii) input sequences consisting of sparsely sampled wavefields.

Densely sampled input data. Here, we use densely sampled wavefields as inputs, where each element of the input and target sequences is a 3-D tensor, $\mathcal{X}_t$, with dimensions of 3 (components) x 344 (X direction) x 224 (Y direction).

Figure 2a displays a series of ground truth wavefield snapshots in the top row, while the middle row visualizes the wavefields predicted by WaveCastNet. Our model demonstrates exceptional capability in capturing the patterns of P- and S-wavefronts, as well as the scattered coda waves. Additionally, we assess WaveCastNet's performance by analyzing

the intensity and timing of the ground motions, focusing particularly on the PGV values.

Spatial distributions of PGV and its timing (T_{PGV}) are shown in Figure 2b-d. The results show accurate reproduction of large PGVs and their arrival times, notably near the earthquake hypocenter at $X = 40$, $Y = 38$ km, within the Livermore basin at $X = 60 - 80$ km, $Y = 40 - 60$ km, and in the northeast corner of the model at $X = 20 - 40$ km. The deviations in PGV values are minimal, less than 5% from the ground truth. Errors in T_{PGV} are generally negligible, although larger discrepancies are observed where T_{PGV} exhibits discontinuities, likely influenced by the underlying geological structures.

These findings show that WaveCastNet can capture complex kinematics and dynamics of wave propagation, and its capability to model phenomena such as amplitude decay — stemming from both geometrical spreading and intrinsic attenuation — and the amplification effects associated with wave reverberation within geological basins.

Sparsely sampled input data. Here, we simulate a scenario more representative of real-world conditions, where seismograph distributions are sparse and irregular, as depicted in Figure 1a. We derive sensor locations from waveforms recorded over the past decade, available through the Northern California Earthquake Data Center database [44].
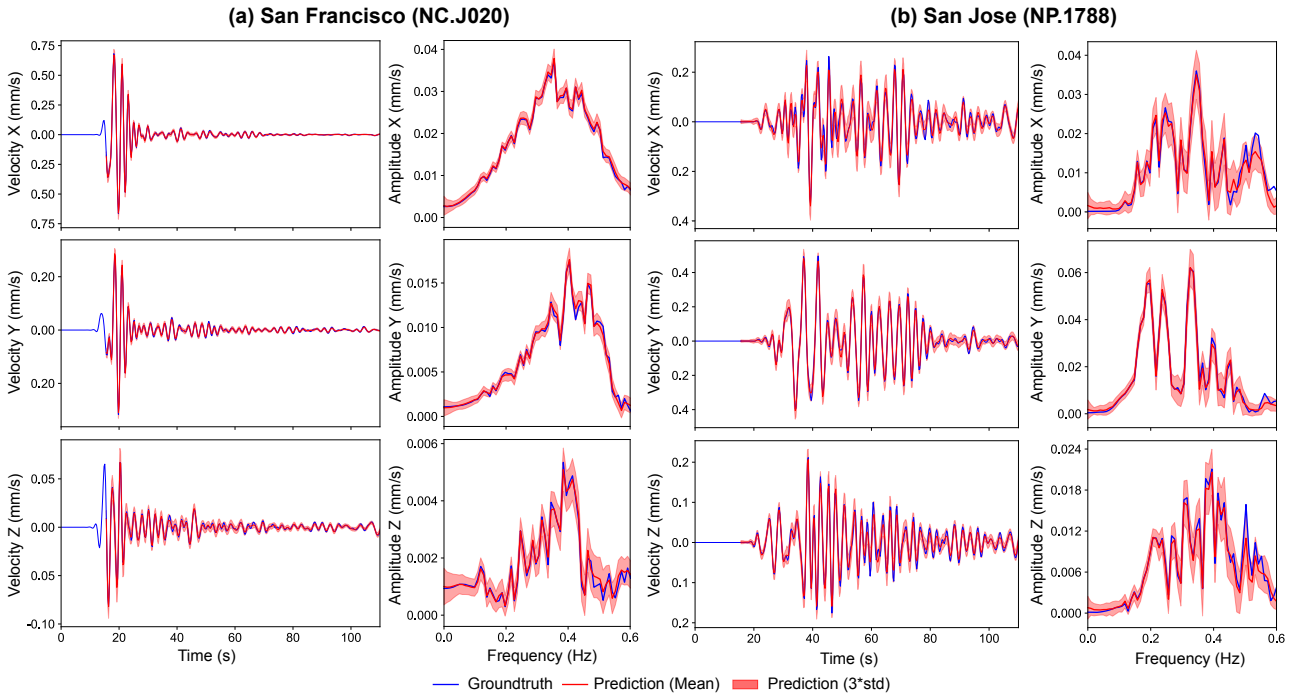
Figure 3: Waveforms from (a) San Francisco (NC.J020) and (b) San Jose (NP.1788) for a point-source earthquake. The blue lines indicate the ground truth, the red lines show the mean of the predicted waveforms, and the red shaded areas represent three times the standard deviation from the mean waveforms.

To obtain sparsely sampled data, we determine the row and column indices $[h, w]$ of each sensor on the wavefield snapshot $\mathcal{X}_t$. After eliminating sensors with overlapping indices, we retain data for 564 sensors, forming an input sequence where each element comprises a 2-D tensor with dimensions 3 (components) x 564 (sparse measurements). The corresponding target sequence consists of densely sampled wavefields, akin to the previous experimental setting. To handle the sparsely sampled input sequences, WaveCastNet incorporates a specialized embedding layer, while all other components of the model architecture remain unchanged.

The bottom row of Figure 2 demonstrates that WaveCastNet effectively predicts wave propagation, PGV, and T_{PGV} across the entire area, even when provided solely with sparsely sampled data. Although the errors are larger compared to those from the dense sampling scenario, sparse measurements are sufficient to capture the dynamics of wave propagation. Table 1 quantitatively evaluates WaveCastNet's performance across these two different scenarios, showcasing its adaptability to varied sampling conditions.

Uncertainty estimation. Quantifying uncertainty in ground motion forecasting is crucial. To address this, we em-

ploy an ensemble approach by training 50 instances of WaveCastNet with different seeds and bootstrapped training data. This approach allows us to calculate the mean and standard deviation of both time-series and their frequency-domain amplitude spectra for stations located in San Francisco and San Jose, as illustrated in Figure 3. Our ensemble successfully captures each waveform in detail, and the predicted amplitude spectra align closely with the ground truth.

Furthermore, WaveCastNet's performance remains reliable even in scenarios where no seismic waves reach a station
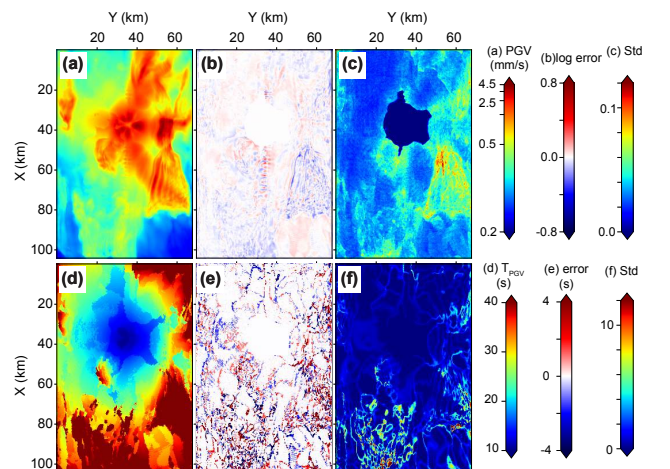

Figure 4: Uncertainty estimates for the dense point-source ground motion prediction. (a,d) Mean, (b,e) errors, and (c,f), standard deviation of (a-c) $\ln PGV$ and (d-f) T_{PGV}. A hole in (c), centered at X=40, Y=38 km, indicates the location where T_{PGV} is within our initial time window.

Input - setting	ACC	RFNE
Dense and regular sampling	0.98	0.20
Sparse and irregular sampling	0.96	0.27

Table 1: Performance Metrics for the dense and sparse sampling scenarios. Providing the model with more information (i.e., dense inputs) helps to improve performance.

5

within the initial input sequence, such as at station NC.J020 in San Jose (see Figure 3b). The mean values of PGVs and their arrival times (T_{PGV}) exhibit excellent agreement with the observed data, as demonstrated in Figures 4a, b, d, and e. The standard deviations for the logarithmic values of PGVs and T_{PGV} are consistently less than 1% of their mean values, indicating that WaveCastNet provides reliable predictions. Notably, slightly higher deviations are observed within the Livermore basin, potentially reflecting WaveCastNet's sensitivity to the complex interactions of multiple wavefronts arriving from different directions.

2.2 Generalization to Finite-fault Large Earthquakes

Earthquake early warning systems are designed to mitigate the hazards posed by large magnitude earthquakes. Unlike small earthquakes, which can be modeled as point sources, large earthquakes necessitate representation as finite-size rupture planes. An earthquake rupture initiates at the epicenter and propagates along the fault, emitting seismic waves from each point. This allows for modeling the effects of a finite-size fault as an aggregation of point sources, each initiating seismic activity at a predetermined time. By using a Green's function response for a point source along the entire fault, it is possible to compute the ground motion from a large magnitude earthquake by integrating the response of multiple point sources regularly distributed on the fault, following a physics-based kinematic rupture model [28, 25, 24, 48].

Inspired by this concept, we evaluate the capabilities of WaveCastNet to predict finite-fault earthquake waveforms using point-source simulations. For this, we employ kinematic rupture models, suitable for earthquakes ranging from M4.5 to M7, developed in accordance with [24]. These models allow us to generate synthetic waveforms using the same simulation method as that for the point sources. The rupture plane is designed as a vertical rectangle, strategically aligned with the locations of the point sources. The dimensions of the rupture planes are scaled in accordance with the earthquake magnitude to adequately release seismic energy (see Table 2), following the guidelines by [35]. Source parameters such as slip, slip rate, rupture initiation time, and local dip exhibit spatial variability and include stochastic fluctuations at minor scales, allowing the simulation to aggregate the linear responses of numerous point sources with varying parameters. WaveCastNet does not incorporate these parameters even during the inference time. Moreover, as the duration of energy release extends with increasing magnitude [56, 45] and the early waveforms remain similar across a range of magnitudes [43], predicting accurate ground motion waveforms presents a significant challenge.

Initially, we normalize the data for finite-fault earthquakes using the same pixel-wise mean and standard deviation tensors derived from the point-source training dataset. Subsequently, we scale the data by the standard deviation calculated from the initial 15.6 seconds of waveform data. WaveCastNet exhibits robust forecasting performance for earthquakes ranging from M4.5 to M5.5. However, as shown in Table 2, performance deteriorates for earthquakes of M6 and above.
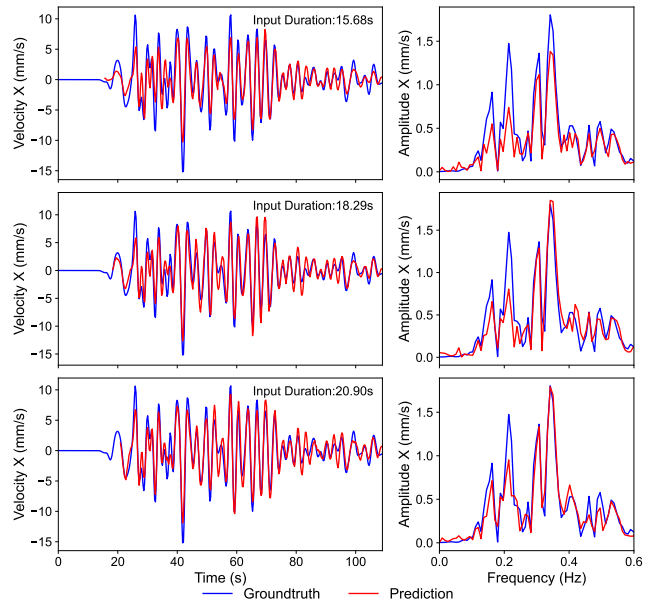

Figure 5: Evolution of M6 ground motion prediction at the station NP.1788 located in San Jose by using the time window of (a) 15.68, (b) 18.29, and (c) 20.90 s.

Mw	Fault size (km × km)	T_{rup} (s)	ACC	RFNE
4.5	1.8 × 1.8	3.5	0.95	0.35
5.0	3.4 × 3	3.7	0.95	0.37
5.5	8 × 4	6.0	0.95	0.42
6.0	12.5 × 8	9.6	0.88	0.52
6.5	26 × 12	13.2	0.66	0.84
7.0	66 × 15	26.6	0.53	0.86

Table 2: Fault size and performance metrics of finite-fault earthquake data predictions using 15.6-second input time window. T_{rup} indicates the end time of the rupture. See Figure D.1-D.6 for the rupture models.

This degradation in performance correlates with the duration of rupture, which extends up to 13.2 seconds for M6.5 and 26.2 seconds for M7 earthquakes, reaching or exceeding the length of the input time window. Consequently, the input waveforms fail to encompass the full extent of the excited energy. This leads to underestimations of amplitude, although the kinematics are reasonably well reproduced, as illustrated in Figure 5a.

To address this limitation, we extend the length of the input time window, a modification feasible in real-world applications. The extended results for the M6 earthquake are shown in Figures 5b-d and 6. As anticipated, the fidelity of waveform recovery is enhanced notably with the expansion of the time window, particularly evident in the low-frequency components. WaveCastNet forecasts phases of waveforms extremely well, but continues to slightly underestimate amplitudes, especially of early arrivals. Nonetheless, the errors in PGV remain within 1.5 log units, but the timing errors can be large. As shown in Figure 5, multiple wavelets show similar peak values challenging to be differentiated especially when the reverberations occur. These results affirm
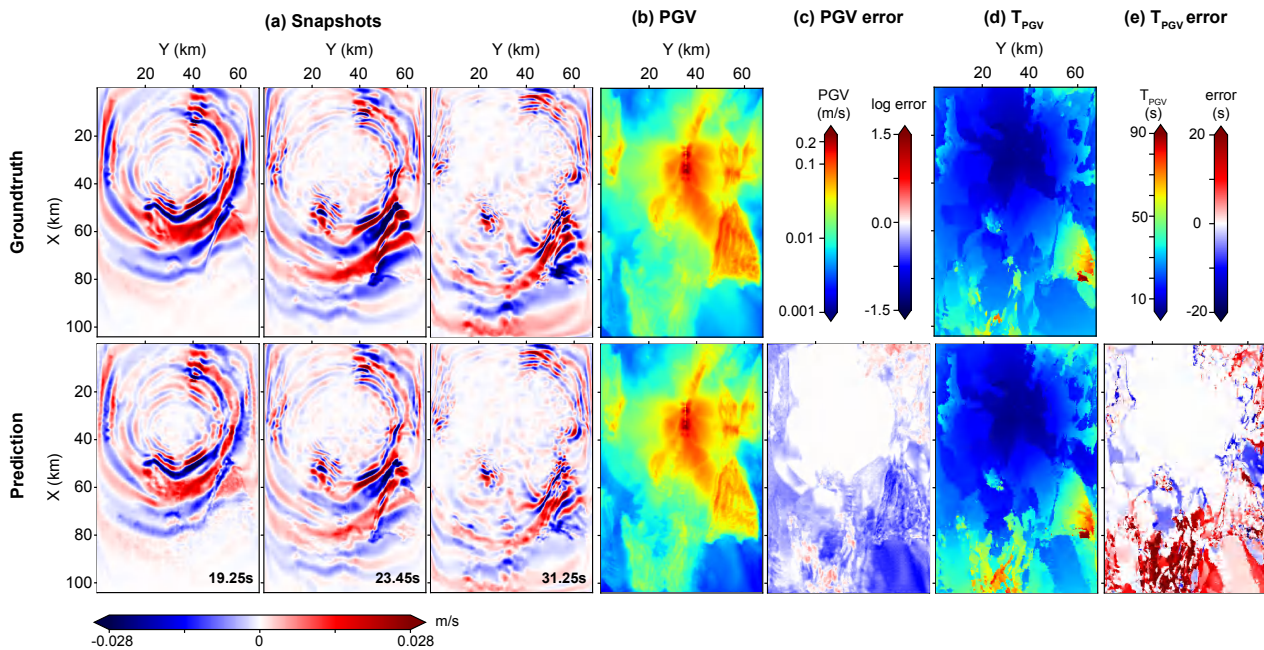
6

Figure 6: M6 earthquake ground motions for (top) ground truth and (bottom) prediction using 18.2-second input time window. (a) Snapshot waveforms for Y components, (b) PGV, (c) PGV error, (d) T_{PGV} and (e) T_{PGV} error.

WaveCastNet's substantial potential to generalize effectively to finite-fault earthquake simulations.

3 Discussion and Conclusions

Our experiments confirm that WaveCastNet holds considerable promise for accurately forecasting wavefields derived from both point-source and finite-fault simulations of large magnitude earthquakes. WaveCastNet shows, for both dense and (irregular) sparse sensor configurations, that it can reliably predict seismic wave propagation as well as that it can capture PGV values. The excellent fit from the first arrival to later coda waveforms is remarkable. We may interpret that this behavior as WaveCastNet captures the Huygens principle, i.e., each spatial point is represented as a new source point. Notably, WaveCastNet can process an entire 100-second sequence in just 0.56 seconds using a single NVIDIA A100 GPU. Moreover, we anticipate that inference time can be even further improved by optimizing both the model architecture, and inference pipeline.

These findings are particularly significant as they demonstrate the practicality of integrating WaveCastNet into earthquake early warning systems. This integration would significantly advance the systems' capabilities, facilitating a more rapid response during seismic events.

Generalization. WaveCastNet demonstrates robust generalization up to M5.5, and show that it effectively generalizes up to M6 when employing a sliding window approach. This modification, which accounts for energies released later in the earthquake rupture process, which was also used for data-assimilation-based earthquake early warning systems, is easy to implement and does not need additional training or alterations to the existing framework. Importantly, our AI-based forecasting approach does not require prior knowledge of earthquake magnitudes or epicenters. This suggests that WaveCastNet can be effectively trained on a limited dataset, while generalizing to different seismic scenarios, including higher magnitude earthquakes.

It is important to stress that applying a model trained on point-source earthquakes to a larger magnitude earthquake is challenging. This is because the physical representation of the rupture process changes from a point source to a finite-size fault, which is represented by a complex kinematic model. The amplitude of waveforms varies substantially between M4.5 and M7, with differences exceeding 80 times. Additionally, the spatial amplitude decay rate of ground motion intensities varies with magnitude due to changes in the fault size. Empirically, we observe that this can complicate the data normalization process, and lead to undesirable underprediction of amplitudes. Moreover, our results suggest that merely extending the input time window is insufficient. Thus, expanding the training set to include waveforms from finite-fault simulations is essential for overcoming these challenges.

Comparative study. To demonstrate the advantages of our proposed approach, we show performance comparisons with baseline models. We evaluate WaveCastNet against seq2seq frameworks which use ConvLSTM [52] and ConvGRU [7] as backbones. Results, presented in Table 3, show better accuracy and lower relative Frobenius norm error for WaveCastNet in predicting point-source earthquakes. These experiments use data that are spatially downsampled by a factor of four to ensure model convergence with less computational resources. Nevertheless, the setup mirrors that discussed in Sec. 2.1, with each $\mathcal{X}_t$ within the sequence reduced to $\mathbb{R}^{C \times \frac{H}{4} \times \frac{W}{4}}$. The findings demonstrate WaveCastNet effectiveness, achieving better accuracy and lower reconstruction errors, while using the same latent space dimensions.

Model	Parameters	Latent Space	Patch Size	Embed Dimension	ACC	RFNE
Seq2seq using ConvGRU[7]	4.99M	(144, 21, 14)	-	-	0.94	0.34
Seq2seq using ConvLSTM [52]	6.65M	(144, 21, 14)	-	-	0.95	0.32
WaveCastNet (ours)	8.15M	(144, 21, 14)	-	-	0.96	0.27
Swin Transformer [39]	13.72M	-	(3,4,4)	144	0.95	0.31
Time-S-Former [23]	10.21M	-	(1,8,8)	192	0.95	0.31
Swin Transformer*	24.27M	-	(4,4,4)	192	0.97	0.25
Time-S-Former*	33.82M	-	(1,8,8)	192	0.98	0.20

Table 3: Performance comparison between seq2seq frameworks using different recurrent cells, and state-of-the-art transformers for forecasting small point-source earthquakes. While larger vision transformers can perform better on this task, we show that these models fail to generalize to domain-shifted settings in Figure 7.

Additionally, we compare WaveCastNet to state-of-the-art transformer architectures designed for spatio-temporal modeling, including the Swin transformer [38] and the Time-S-Former [23]. Despite the good performance on the task of predicting point-source earthquakes, these transformers struggled with generalization in forecasting higher magnitude earthquakes, as indicated by large relative errors across magnitudes in Figure 7. The comparative study reveals that our WaveCastNet offers beneficial trade-offs: it requires fewer parameters than transformers, facilitates faster inference times, and introduces a regularization effect through its information bottleneck, aiding generalization.

Future Directions. Our experiments used synthetic, noise-free data at frequencies below 0.5 Hz. Moving forward, we plan to apply WaveCastNet to actual earthquake observations — a process we are currently preparing to undertake. The strong generalization capabilities observed suggest that it is sufficient to train WaveCastNet on a large number of real, small-magnitude earthquake recordings. We also expect that the model can be trained on both synthetic and real ground motion data, which may help to reduce uncertainties in visco-elastic earth models and earthquake source parameters.
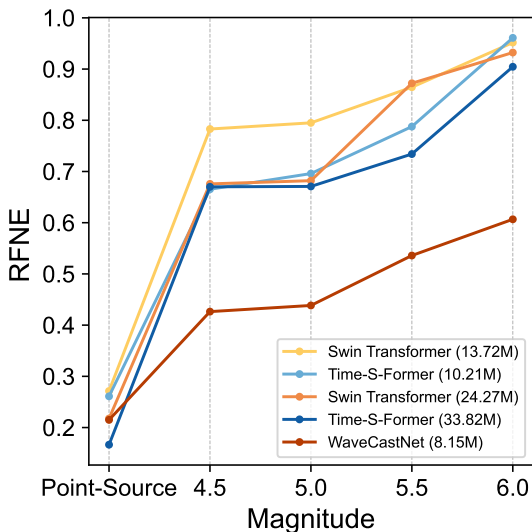

Figure 7: Generalization performance as a function of the earthquake magnitude. All models are trained on point-source earthquakes only, and it can be seen that WaveCastNet generalizes best to domain-shifted settings.

Future direction include also the exploration of data augmentation schemes to further improve the robustness to domain shifted settings [36, 37, 15], as well as recently proposed state-space models for modelling sequences [27, 59, 58].

4   Method

In this section, we outline the methodology behind WaveCastNet. We begin with an overview of the sequence-to-sequence (seq2seq) framework that serves as the basis for our forecasting model. We then explain the ConvLEM model, which is central to our approach. We then describe the data normalization and preprocessing strategies we employed, as well as the processes involved in generating the data used for our experiments.

4.1   Wavefield Forecasting Network

Our WaveCastNet is based on the sequence-to-sequence (seq2seq) framework, originally developed for natural language processing [54]. Similar to other seq2seq models, WaveCastNet comprises four primary components:

- **Embedding layer.** This layer maps input wavefields into a latent space. We employ two types of embedding layers: (i) convolutional layers enhanced with batch normalization and LeakyReLU activation, optimized for embedding densely sampled wavefields into a latent space; and (ii) fully connected layers, followed by convolutional layers, optimized for embedding sparsely sampled wavefields into a latent space.

- **Encoder.** The encoder processes the embedded sequence into a fixed-size encoder state that provides a compressed summary of the input sequence necessary for generating the target sequence.

- **Decoder.** Operating sequentially, the decoder predicts each element of the target sequence one at a time. It uses the previously predicted output combined with the encoder state to forecast the next element.

- **Reconstruction layer.** The reconstruction layer allows us to recover detailed spatial information from the predicted latent sequences by using transposed convolutional layers alongside pixel-shuffle techniques to reconstruct the high-resolution wavefield.

Both the encoder and decoder use our novel ConvLEM cell (see Section 4.2 for details), which is designed to capture complex multi-scale patterns in both spatial and temporal dimensions. Additional technical details of the embedding and reconstruction layers are discussed in the appendix.

The seq2seq framework seeks to find a target sequence $\mathcal{Y} := \mathcal{X}_{J+1}, \ldots, \mathcal{X}_{J+K}$, from a given input sequence $\mathcal{X} := \mathcal{X}_1, \ldots, \mathcal{X}_J$. The objective is to optimize the conditional probability:

$$\tilde{\mathcal{Y}} = \arg\max_{\mathcal{Y}} p(\mathcal{Y}|\mathcal{X}) \approx \mathcal{D}_{\text{decoder}}(\mathcal{E}_{\text{encoder}}(\mathcal{X})). \quad (3)$$

While it is challenging to compute the conditional probability directly, an encoder-decoder framework can be used to generate an approximate target sequence [54]. In this process, an encoder, denoted as $\mathcal{E}_{\text{encoder}}$, compresses the embedded input sequence $\mathcal{X}$ into a concise encoder state. This state is subsequently used by a decoder, $\mathcal{D}_{\text{decoder}}$, to generate the predicted latent sequence $\tilde{\mathcal{Y}}$, which can then be mapped to desired output space by a reconstruction layer. This approach effectively leverages the encoded information to produce a sequence that approximates the target sequence.

WaveCastNet tailors this seq2seq framework specifically for the task of forecasting ground motions, treating the prediction challenge as a regression problem. We aim to minimize the sum of all the squared differences between the predicted wavefields $\hat{\mathcal{X}}$ and the actual wavefields $\mathcal{X}$:

$$\mathcal{L}_2 = \frac{1}{T} \sum_{t=1}^{T} \|\hat{\mathcal{X}}_t - \mathcal{X}_t\|_F^2, \quad (4)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Under the assumption that prediction errors follow a normal distribution, minimizing the $\mathcal{L}_2$ loss corresponds to maximizing the likelihood of the data given the model. This approach guides the learning of the model parameters through the minimization of the loss across all forecasted and actual sequences. During inference, the model uses these learned parameters to generate target sequences for new input sequences.

To further enhance the model's performance, we adopt the Huber loss during training, defined as follows:

$$\mathcal{L}_{\text{Huber}} = \frac{\sum_{t,c,h,w} L_\delta \left(\hat{\mathcal{X}}_t[c,h,w], \mathcal{X}_t[c,h,w] \right)}{TCHW}, \quad (5)$$

with the loss function L_δ given by:

$$L_\delta(\hat{x}, x) = \begin{cases} \frac{1}{2}(\hat{x} - x)^2 & \text{for } |\hat{x} - x| \leq \delta, \\ \delta \cdot \left(|\hat{x} - x| - \frac{1}{2}\delta \right) & \text{otherwise.} \end{cases}$$

$$(6)$$

The Huber loss effectively balances the $L1$ and $L2$ norms, which supports more robust fitting across various earthquake conditions and depths during training. Specifically, we find that the Huber loss improves WaveCastNet's capability to better capture the challenging PGV patterns. Moreover, we observe that using this loss enables our model to better generalize across different earthquake magnitudes and conditions, while also ensuring faster convergence during training.

4.2 Convolutional Long Expressive Memory

We propose Convolutional Long Expressive Memory (ConvLEM) to overcome the limitations of traditional recurrent units in modeling complex multi-scale structures across spatial and temporal dimensions. These limitations are highlighted when recurrent units are viewed as dynamical systems [12, 16], where the evolution over time is governed by a system of input-dependent ordinary differential equations:

$$\frac{d\mathbf{h}}{dt} = \tau \cdot f_\theta(\mathbf{h}(t), \mathbf{x}(t)), \quad (7)$$

where inputs $\mathbf{x}(t) \in \mathbb{R}^d$ and hidden states $\mathbf{h}(t) \in \mathbb{R}^l$ are modeled as continuous functions over time $t \in [0, T]$. However, this model is limited to modeling dynamics at a fixed temporal scale τ. An intuitive approach to address this issue involves integrating a high-dimensional gating function to replace τ, aiming to model dynamics occurring across various time scales [13, 17]. Nevertheless, employing a single gating mechanism often falls short of adequately capturing the complexities found in more challenging dynamical systems.

In this work, we enhance the modeling of multi-scale temporal structures by extending the recently introduced Long Expressive Memory (LEM) unit [50]. This approach is based on the following coupled differential equations:

$$\begin{aligned} \frac{d\mathbf{c}(t)}{dt} &= \mathbf{g}_c \odot \left[f_{\theta_c}^c(\mathbf{h}(t), \mathbf{x}(t)) - \mathbf{c}(t) \right], \\ \frac{d\mathbf{h}(t)}{dt} &= \mathbf{g}_h \odot \left[f_{\theta_h}^h(\mathbf{c}(t), \mathbf{x}(t)) - \mathbf{h}(t) \right], \end{aligned} \quad (8)$$

where $\mathbf{h}(t) \in \mathbb{R}^l$ and $\mathbf{c}(t) \in \mathbb{R}^l$ represent the slow and fast evolving hidden states, respectively. The gating functions $\mathbf{g}_c$ and $\mathbf{g}_h$, which are dependent on both the input and the states, introduce variability in temporal scales into the dynamics of the model. Here, $\odot$ signifies the Hadamard product, ensuring element-wise multiplication.

We advance the basic LEM unit by incorporating convolutional operations that facilitate modeling of both input-to-state and state-to-state transitions, akin to those used in the ConvLSTM model [52]. By representing the hidden states and inputs as tensors, we are better able to preserve and model critical multi-scale spatial patterns. The ConvLEM is thus formulated as:

$$\frac{d\mathcal{C}(t)}{dt} = \mathbf{g}_c \odot \left[f_{\theta_c}^c(\mathcal{H}(t), \mathcal{X}(t)) - \mathcal{C}(t) \right], \quad (9)$$

$$\frac{d\mathcal{H}(t)}{dt} = \mathbf{g}_h \odot \left[f_{\theta_h}^h(\mathcal{C}(t), \mathcal{X}(t)) - \mathcal{H}(t) \right], \quad (10)$$

In this equation, $\mathcal{H}(t) \in \mathbb{R}^{r \times p \times q}$ and $\mathcal{C}(t) \in \mathbb{R}^{r \times p \times q}$ denote the slow and fast evolving hidden states, respectively. The input $\mathcal{X}(t) \in \mathbb{R}^{c \times h \times w}$ is a three-dimensional tensor.

To effectively train this model, using an appropriate discretization scheme is essential, as it enables the learning of model weights through backpropagation over time. Following the methodology presented in [50], we consider a positive timestep Δt and use the Implicit-Explicit (IMEX) time-stepping scheme. This approach aids in formulating

the discretized version of the ConvLEM unit as follows:

$$\Delta\mathbf{t}_n = \Delta t\,\mathbf{g}_c \tag{11}$$

$$\overline{\Delta\mathbf{t}_n} = \Delta t\,\mathbf{g}_h \tag{12}$$

$$\mathcal{C}_n = (\mathbb{1} - \Delta\mathbf{t}_n) \odot \mathcal{C}_{n-1} + \Delta\mathbf{t}_n \odot f_{\theta_c}^c \tag{13}$$

$$\mathcal{H}_n = \left(\mathbb{1} - \overline{\Delta\mathbf{t}_n}\right) \odot \mathcal{H}_{n-1} + \overline{\Delta\mathbf{t}_n} \odot f_{\theta_h}^h \tag{14}$$

with update functions

$$f_{\theta_c}^c = tanh\left(\mathbf{W}_{hc} * \mathcal{H}_{n-1} + \mathbf{W}_{xc} * \mathcal{X}_n\right), \tag{15}$$

$$f_{\theta_h}^h = tanh\left(\mathbf{W}_{ch} * \mathcal{C}_n + \mathbf{W}_{xh} * \mathcal{X}_n\right), \tag{16}$$

and gating functions

$$\mathbf{g}_h = \sigma\left(\mathbf{W}_{x\bar{t}} * \mathcal{X}_n + \mathbf{W}_{h\bar{t}} * \mathcal{H}_{n-1}\right), \tag{17}$$

$$\mathbf{g}_c = \sigma\left(\mathbf{W}_{xt} * \mathcal{X}_n + \mathbf{W}_{ht} * \mathcal{H}_{n-1}\right). \tag{18}$$

In this notation, $\mathbf{W}_{.,.}$ denotes the weight tensors, $\odot$ represents the Hadamard product, and $*$ indicates the convolutional operator, with subscript n marking a discrete time step ranging from 1 to N. The matrix of ones, denoted as $\mathbb{1}$, matches the shape of the hidden states. The sigmoid function σ, used in the gating functions, maps activations to a range between 0 and 1. Note, for brevity, bias vectors are omitted from the update and gating function.

Based on the model structures outlined above, we further introduce a reset gate $\mathbf{g}_{reset}$ to refine the modeling of the correlation between fast and slow hidden states:

$$\mathbf{g}_{reset} = \sigma\left(\mathbf{W}_{xr} * \mathcal{X}_n + \mathbf{W}_{hr} * \mathcal{H}_{n-1}\right). \tag{19}$$

The reset gate is integrated into the update function for the slow hidden states as follows:

$$f_{\theta_h}^h = tanh\left(\mathbf{g}_{reset} \odot \left(\mathbf{W}_{ch} * \mathcal{C}_n\right) + \mathbf{W}_{xh} * \mathcal{X}_n\right). \tag{20}$$

Intuitively, this additional gate helps to improve the flow of relevant information from the updated fast hidden states into updating the slow hidden states.

Enhancing the gating functions proves beneficial for modeling complex spatio-temporal problems in practice. Leveraging the concept of "peephole connections" [53], we further enhance the gates by injecting information about the fast hidden states. We define these gates as follows:

$$\mathbf{g}_h = \sigma\left(\mathbf{W}_{x\bar{t}} * \mathcal{X}_n + \mathbf{W}_{h\bar{t}} * \mathcal{H}_{n-1} + \mathbf{W}_{c\bar{t}} \odot \mathcal{C}_{n-1}\right),$$
$$\mathbf{g}_c = \sigma\left(\mathbf{W}_{xt} * \mathcal{X}_n + \mathbf{W}_{ht} * \mathcal{H}_{n-1} + \mathbf{W}_{ct} \odot \mathcal{C}_n\right),$$
$$\mathbf{g}_{reset} = \sigma\left(\mathbf{W}_{xr} * \mathcal{X}_n + \mathbf{W}_{hr} * \mathcal{H}_{n-1} + \mathbf{W}_{cr} \odot \mathcal{C}_n\right).$$

These modified gates show an improved ability to process longer sequences more accurately. Intuitively, by incorporating additional contextual information, these gates are better suited to model complex multi-scale dynamics, which in turn improves the model's expressiveness.

Figure 8 illustrates the discretized ConvLEM unit.

4.3 Normalization

Seismic waves exhibit varying residence times as they travel through different geographic locations, leading to significantly greater ground motion variance in certain regions.
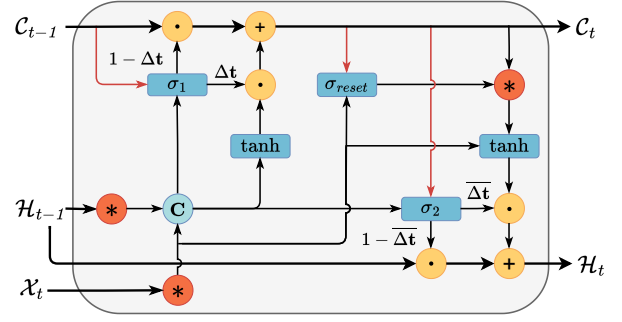

Figure 8: Schematic of the ConvLEM cell. Here, σ_1 and σ_2 represent $\mathbf{g}_c$ and $\mathbf{g}_h$ respectively. Update function f is set to tanh. Red links indicate peephole connections.

Therefore, normalizing is crucial in order to obtain a good forecasting performance. In this work, we use a particle velocity-wise normalization scheme for each snapshot.

Consider all Q sequences in the training set, whereas each sequence is composed of T snapshots $\{\mathcal{X}_t^q\}$. For each particle velocity $\mathcal{X}[c, h, w]$, we compute the mean and standard deviation values across all snapshots in the training set:

$$\{\mathcal{X}_t^q[c, h, w] | q = 0, 1, 2, \ldots Q - 1; t = 0, 1, 2, \ldots T - 1\}.$$

The resulting mean and standard deviation tensors have the same shape as the snapshot $\mathcal{X}_t$, denoted as $\mathcal{X}_{\text{mean}}$, $\mathcal{X}_{\text{std}}$, respectively. During the data preprocessing stage, for each snapshot $\mathcal{X}_t$, we apply particle velocity-wise normalization as follows:

$$\bar{\mathcal{X}}_t = \frac{\mathcal{X}_t[c, h, w] - \mathcal{X}_{\text{mean}}[c, h, w]}{\mathcal{X}_{\text{std}}[c, h, w]}.$$

Particle velocity-wise normalization also prevents potential spatial information leakage during the normalization process for our sparse sampling scenario.

Normalization for domain-shifted settings. The ground motion of earthquakes with higher magnitudes (e.g., M4.5-M7), once normalized, exhibits a considerably wider range compared to the normalized M4 data. Thus, we need to normalize the ground motions again to obtain a reasonable range using the information present in the input window. Given the input window from time step t_1 to t_2, we conduct a channel-wise normalization for each input snapshot $\mathcal{X}_t$ based on the standard deviation values computed for the following set:

$$\{\bar{\mathcal{X}}_t[c, h, w] | t = t_1, t_1 + 1, \ldots, t_2; h = 0, 1, 2, \ldots H - 1;$$
$$w = 0, 1, 2, \ldots W - 1\}.$$

The reasons for not using the particle velocity-wise normalization here are twofold. Firstly, the initial particle velocity-wise normalization has already introduced varying standard deviations for different spatial locations. Secondly, since ground motion in the early warning area is observed to be zero within the input window, the particle velocity-wise standard deviation tensor would consist mostly of zeros, making the normalization process infeasible.

4.4 Data Generation

We simulate point-source and finite-fault earthquake ground motions up to 0.5 Hz within a three-dimensional (3D) volume extending 120 km in the fault parallel (FP) direction (X direction), 80 km in the fault normal (FN) direction (Y direction), and 30 km in depth. These simulations are conducted using the USGS San Francisco Bay region 3D seismic velocity model (SFVM) v21.1 [29]. Material properties, including the Vp-Vs relationships, are defined for each geological unit based on laboratory and well-log measurements, which include parameters such as P- and S-wave velocities [29, 10, 2]. Simulations are initiated with a minimum S-wave velocity of 500 m/s. We generate visco-elastic wave fields using the open-source SW4 package, which computes the 4th order finite-difference solution of the visco-elastic wave equations [47]. This software package is well-established, with its accuracy validated through numerous ground motion simulations [40, 41, 49].

The surface of the Earth is modeled with a free surface condition, while the outer boundaries use absorbing boundary conditions through a super grid approach spanning 30 grids. We consider a flat surface, and to avoid numerical dispersion, we consider a simulation grid with a mesh size of 150 m^3 at the surface, designed to ensure a minimum of six grids per wavelength. To optimize computational resources, the mesh size is doubled at depths of 2.2 km and 6.6 km. The largest grid size employed is 600 m^3, covering a total of approximately 9.59 million grid points. The attenuation and velocity dispersion are modeled using three standard linear solid models, assuming a constant Q over the simulated frequency range. Each simulation runs for 120 seconds with a time step of 0.0260134 seconds, resulting in 4,613 time steps. The three component particle velocity motions are recorded every 10 steps (i.e., 0.26014 sec) at 150 m x 150 m grids and then are downsampled to 300 m $\times$ 300 m grids for the training and testing WaveCastNet. These simulations are carried out on 12 nodes equipped with INTEL XEON Gold 5218/6230 CPUs within the Lawrencium cluster at Lawrence Berkeley National Laboratory.

Acknowledgements

References

[1] B. T. Aagaard, T. M. Brocher, D. Dolenc, D. Dreger, R. W. Graves, S. Harmsen, S. Hartzell, S. Larsen, and M. L. Zoback. Ground-Motion Modeling of the 1906 San Francisco Earthquake, Part I: Validation Using the 1989 Loma Prieta Earthquake. *Bulletin of the Seismological Society of America*, 98(2):989–1011, 2008.

[2] B. T. Aagaard, R. W. Graves, A. Rodgers, T. M. Brocher, R. W. Simpson, D. Dreger, N. A. Petersson, S. C. Larsen, S. Ma, and R. C. Jachens. Ground-Motion Modeling of Hayward Fault Scenario Earthquakes, Part II: Simulation of Long-Period and Broadband Ground Motions. *Bulletin of the Seismological Society of America*, 100(6):2945–2977, 2010.

[3] R. M. Allen and D. Melgar. Earthquake Early Warning: Advances, Scientific Challenges, and Societal Needs. *Annual Review of Earth and Planetary Sciences*, 47(1):1–28, 2019.

[4] R. M. Allen and M. Stogaitis. Global Growth of Earthquake Early Warning. *Science*, 375(6582):717–718, 2022.

[5] M. Arjovsky, A. Shah, and Y. Bengio. Unitary Evolution Recurrent Neural Networks. In *International Conference on Machine Learning*, pages 1120–1128, 2016.

[6] G. M. Atkinson and D. M. Boore. Modifications to Existing Ground-Motion Prediction Equations in Light of New DataModifications to Existing Ground-Motion Prediction Equations in Light of New Data. *Bulletin of the Seismological Society of America*, 101(3):1121–1135, 2011.

[7] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving Deeper Into Convolutional Networks for Learning Video Representations. *International Conference on Learning Representations*, 2016.

[8] J. Bayless and N. A. Abrahamson. Summary of the BA18 Ground-Motion Model for Fourier Amplitude Spectra for Crustal Earthquakes in CaliforniaSummary of the BA18 Ground-Motion Model for Fourier Amplitude Spectra for Crustal Earthquakes. *Bulletin of the Seismological Society of America*, 109(5):2088–2105, 2019.

[9] Y. Bozorgnia, N. A. Abrahamson, L. A. Atik, T. D. Ancheta, G. M. Atkinson, J. W. Baker, A. Baltay, D. M. Boore, K. W. Campbell, B. S.-J. Chiou, R. Darragh, S. Day, J. Donahue, R. W. Graves, N. Gregor, T. Hanks, I. Idriss, R. Kamai, T. Kishida, A. Kottke, S. A. Mahin, S. Rezaeian, B. Rowshandel, E. Seyhan, S. Shahi, T. Shantz, W. Silva, P. Spudich, J. P. Stewart, J. Watson-Lamprey, K. Wooddell, and R. Youngs. NGA-West2 Research Project. *Earthquake Spectra*, 30(3):973–987, 2014.

[10] T. M. Brocher. Compressional and Shear-Wave Velocity versus Depth Relations for Common Rock Types in Northern CaliforniaCompressional and Shear-Wave Velocity versus Depth Relations for Common Rock

Types in Northern CA. *Bulletin of the Seismological Society of America*, 98(2):950–968, 2008.

[11] C. Chai, M. Maceira, H. J. Santos-Villalobos, S. V. Venkatakrishnan, M. Schoenball, W. Zhu, G. C. Beroza, C. Thurber, and E. C. Team. Using a Deep Neural Network and Transfer Learning to Bridge Scales for Seismic Phase Picking. *Geophysical Research Letters*, 47(16), 2020.

[12] B. Chang, M. Chen, E. Haber, and E. H. Chi. AntisymmetricRNN: A dynamical system view on recurrent neural networks. In *International Conference on Learning Representations*, 2018.

[13] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

[14] J. Douglas, S. Akkar, G. Ameri, P.-Y. Bard, D. Bindi, J. J. Bommer, S. S. Bora, F. Cotton, B. Derras, M. Hermkes, N. M. Kuehn, L. Luzi, M. Massa, F. Pacor, C. Riggelsen, M. A. Sandıkkaya, F. Scherbaum, P. J. Stafford, and P. Traversa. Comparisons Among the Five Ground-Motion Models Developed Using RESORCE for the Prediction of Response Spectral Accelerations due to Earthquakes in Europe and the Middle East. *Bulletin of Earthquake Engineering*, 12(1):341–358, 2014.

[15] B. Erichson, S. H. Lim, W. Xu, F. Utrera, Z. Cao, and M. Mahoney. NoisyMix: boosting model robustness to common corruptions. In *International Conference on Artificial Intelligence and Statistics*, pages 4033–4041. PMLR, 2024.

[16] N. B. Erichson, O. Azencot, A. Queiruga, L. Hodgkinson, and M. W. Mahoney. Lipschitz recurrent neural networks. In *International Conference on Learning Representations*, 2021.

[17] N. B. Erichson, S. H. Lim, and M. W. Mahoney. Gated recurrent neural networks with weighted time-delay feedback. *arXiv preprint arXiv:2212.00228*, 2022.

[18] N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz. Shallow Neural Networks for Fluid Flow Reconstruction With Limited Sensors. *Proceedings of the Royal Society A*, 476(2238):20200097, 2020.

[19] R. D. D. Esfahani, F. Cotton, M. Ohrnberger, and F. Scherbaum. TFCGAN: Nonstationary Ground-Motion Simulation in the Time–Frequency Domain Using Conditional Generative Adversarial Network (CGAN) and Phase Retrieval Methods. *Bulletin of the Seismological Society of America*, 113(1):453–467, 2022.

[20] M. A. Florez, M. Caporale, P. Buabthong, Z. E. Ross, D. Asimaki, and M.-A. Meier. Data-Driven Synthesis of Broadband Earthquake Ground Motions Using

Artificial Intelligence. *Bulletin of the Seismological Society of America*, 112(4):1979–1996, 2022.

[21] T. Furumura, T. Maeda, and A. Oba. Early Forecast of Long-Period Ground Motions via Data Assimilation of Observed Ground Motions and Wave Propagation Simulations. *Geophysical Research Letters*, 46(1):138–147, 2019.

[22] T. Furumura and Y. Oishi. An Early Forecast of Long-Period Ground Motions of Large Earthquakes Based on Deep Learning. *Geophysical Research Letters*, 50(6), 2023.

[23] Gedas Bertasius and Heng Wang and Lorenzo Torresani. Is Space-Time Attention All You Need for Video Understanding? In *International Conference on Machine Learning*, 2021.

[24] R. Graves and A. Pitarka. Kinematic Ground-Motion Simulations on Rough Faults Including Effects of 3D Stochastic Velocity Perturbations. *Bulletin of the Seismological Society of America*, 106(5):2136–2153, 2016.

[25] R. W. Graves and A. Pitarka. Broadband Ground-Motion Simulation Using a Hybrid Approach. *Bulletin of the Seismological Society of America*, 100(5A):2095–2123, 2010.

[26] N. Gregor, N. A. Abrahamson, G. M. Atkinson, D. M. Boore, Y. Bozorgnia, K. W. Campbell, B. S.-J. Chiou, I. Idriss, R. Kamai, E. Seyhan, W. Silva, J. P. Stewart, and R. Youngs. Comparison of NGA-West2 GMPEs. *Earthquake Spectra*, 30(3):1179–1197, 2014.

[27] A. Gu, K. Goel, and C. Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[28] S. Hartzell, S. Harmsen, A. Frankel, and S. Larsen. Calculation of Broadband Time Histories of Ground Motion: Comparison of Methods and Validation Using Strong-Ground Motion From the 1994 Northridge Earthquake. *Bulletin of the Seismological Society of America*, 89(6):1484–1504, 1999.

[29] E. Hirakawa and B. Aagaard. Evaluation and Updates for the USGS San Francisco Bay Region 3D Seismic Velocity Model in the East and North Bay Portions. *Bulletin of the Seismological Society of America*, 2022.

[30] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[31] M. Hoshiba and S. Aoki. Numerical Shake Prediction for Earthquake Early Warning: Data Assimilation, Real-Time Shake Mapping, and Simulation of Wave PropagationNumerical Shake Prediction for Earthquake Early Warning. *Bulletin of the Seismological Society of America*, 105(3):1324–1338, 2015.

[32] E. Kalnay. *Atmospheric Modeling, Data Assimilation, and Predictability*. Cambridge University Press, 2003.

[33] M. D. Kohler, E. S. Cochran, D. Given, S. Guiwits, D. Neuhauser, I. Henson, R. Hartog, P. Bodin,

V. Kress, S. Thompson, et al. Earthquake Early Warning ShakeAlert System: West Coast Wide Production Prototype. *Seismological Research Letters*, 89(1):99–107, 2018.

[34] M. D. Kohler, D. E. Smith, J. Andrews, A. I. Chung, R. Hartog, I. Henson, D. D. Given, R. de Groot, and S. Guiwits. Earthquake Early Warning ShakeAlert 2.0: Public Rollout. *Seismological Research Letters*, 91(3):1763–1775, 2020.

[35] M. Leonard. Earthquake Fault Scaling: Self-Consistent Relating of Rupture Length, Width, Average Displacement, and Moment Release. *Bulletin of the Seismological Society of America*, 100(5A):1971–1988, 2010.

[36] S. H. Lim, N. B. Erichson, L. Hodgkinson, and M. W. Mahoney. Noisy recurrent neural networks. *Advances in Neural Information Processing Systems*, 34:5124–5137, 2021.

[37] S. H. Lim, N. B. Erichson, F. Utrera, W. Xu, and M. W. Mahoney. Noisy feature mixup. In *International Conference on Learning Representations*, 2021.

[38] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, pages 10012–10022, 2021.

[39] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video Swin Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.

[40] D. McCallen, A. Petersson, A. Rodgers, A. Pitarka, M. Miah, F. Petrone, B. Sjogreen, N. Abrahamson, and H. Tang. EQSIM—A Multidisciplinary Framework for Fault-to-Structure Earthquake Simulations on Exascale Computers Part I: Computational Models and Workflow. *Earthquake Spectra*, page 875529302097098, 2020.

[41] D. McCallen, F. Petrone, M. Miah, A. Pitarka, A. Rodgers, and N. Abrahamson. EQSIM—A Multidisciplinary Framework for Fault-to-Structure Earthquake Simulations on Exascale Computers, Part II: Regional Simulations of Building Response. *Earthquake Spectra*, page 875529302097098, 2020.

[42] M. Meier, T. Heaton, and J. Clinton. The Gutenberg Algorithm: Evolutionary Bayesian Magnitude Estimates for Earthquake Early Warning with a Filter BankThe Gutenberg Algorithm: Evolutionary Bayesian Magnitude Estimates for EEW. *Bulletin of the Seismological Society of America*, 105(5):2774–2786, 2015.

[43] M.-A. Meier, J. P. Ampuero, and T. H. Heaton. The Hidden Simplicity of Subduction Megathrust Earthquakes. *Science*, 357(6357):1277–1281, 2017.

[44] NCEDC. Northern California Earthquake Data Center. UC Berkeley Seismological Laboratory.

[45] S. Noda and W. L. Ellsworth. Scaling Relation Between Earthquake Magnitude and the Departure Time from P Wave Similar Growth. *Geophysical Research Letters*, 43(17):9053–9060, 2016.

[46] A. Petersson, B. Sjogreen, and H. Tang. SW4: User's Guide, version 3.0. *Technical Report LLNL-SM-741439*, Lawrence Livermore National Laboratory, 2023.

[47] N. A. Petersson and B. Sjögreen. Stable Grid Refinement and Singular Source Discretization for Seismic Wave Simulations. *Comm. Comput. Phys.*, 8(5):1074–1110, 2010.

[48] A. Pitarka, R. Graves, K. Irikura, K. Miyakoshi, C. Wu, H. Kawase, A. Rodgers, and D. McCallen. Refinements to the Graves–Pitarka Kinematic Rupture Generator, Including a Dynamically Consistent Slip-Rate Function, Applied to the 2019 Mw 7.1 Ridgecrest Earthquake. *Bulletin of the Seismological Society of America*, 2021.

[49] A. J. Rodgers, A. Pitarka, N. A. Petersson, B. Sjögreen, and D. B. McCallen. Broadband (0–4 Hz) Ground Motions for a Magnitude 7.0 Hayward Fault Earthquake With Three-Dimensional Structure and Topography. *Geophysical Research Letters*, 45(2):739–747, 2018.

[50] T. K. Rusch, S. Mishra, N. B. Erichson, and M. W. Mahoney. Long Expressive Memory for Sequence Modeling. In *International Conference on Learning Representations*, 2021.

[51] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[52] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Conference on Neural Information Processing Systems*, 28, 2015.

[53] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised Learning of Video Representations Using LSTMs. In *International Conference on Machine Learning*, pages 843–852, 2015.

[54] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning With Neural Networks. *Conference on Neural Information Processing Systems*, 27, 2014.

[55] D. T. Trugman, M. T. Page, S. E. Minson, and E. S. Cochran. Peak Ground Displacement Saturates Exactly When Expected: Implications for Earthquake Early Warning. *Journal of Geophysical Research: Solid Earth*, 124(5):4642–4653, 2019.

[56] T. Uchide and S. Ide. Scaling of Earthquake Rupture Growth in the Parkfield Area: Self-Similar Growth and Suppression by the Finite Seismogenic Layer. *Journal of Geophysical Research: Solid Earth*, 115(B11), 2010.

[57] Y. Yang, A. F. Gao, K. Azizzadenesheli, R. W. Clayton, Z. E. Ross, and Y. Yang. Rapid Seismic Waveform Modeling and Inversion With Neural Operators. *IEEE*

Transactions on Geoscience and Remote Sensing, 61:1–12, 2023.

[58] A. Yu, M. W. Mahoney, and N. B. Erichson. There is HOPE to avoid HiPPOs for long-memory state space models. *arXiv preprint arXiv:2405.13975*, 2024.

[59] A. Yu, A. Nigmetov, D. Morozov, M. W. Mahoney, and N. B. Erichson. Robustifying state-space models for long sequences via approximate diagonalization. In *International Conference on Learning Representations*, 2023.

[60] X. Zhang, M. Zhang, and X. Tian. Real-Time Earthquake Early Warning With Deep Learning: Application to the 2016 M 6.0 Central Apennines, Italy Earthquake. *Geophysical Research Letters*, 48(5), 2021.

[61] W. Zhu and G. C. Beroza. PhaseNet: A Deep-Neural-Network-Based Seismic Arrival Time Picking Method. *Geophysical Journal International*, 2018.

A Notation

Terms	Definition
WaveCastNet	Wavefield forecasting network (WaveCastNet) based on a seq2seq model.
Seq2Seq	AI-enabled sequence to sequence (seq2seq) modelling framework.
ConvLEM	Convolutional long expressive memory (ConvLEM) recurrent unit.
time window	Sequence length in temporal dimension.
arrival time	Time step at which maximal waveform arrives.
particle velocity	Single pixel $\mathcal{X}_t[c, h, w]$ in the snapshot.
waveform	Time series recorded for a single particle velocity.
wavefield	Snapshot $\mathcal{X}_t$ at a certain time step.
t	Temporal coordinate (time point).
h, w	XY-index of each input snapshot.
c	Channel index for velocity in a certain direction.
$\mathcal{X}_t$	Snapshot of shape $C \times H \times W$ at time step t.
$\mathcal{C}_n$	Fast hidden state in latent space.
$\mathcal{H}_n$	Slow hidden state in latent space.
X (NS)	North-South direction.
Y (EW)	East-West direction.
Z (UP)	Vertical direction, positive values signify upward movement.

Table A.1: Terms and Definitions

B Technical Details

B.1 Discretized ConvLEM

Here we derive the discretized formula of ConvLEM from the following time-dependent ODEs:

$$\frac{d\mathcal{C}(t)}{dt} = \psi_\mathcal{C}\left(\mathcal{C}(t), \mathcal{H}(t), \mathcal{X}(t)\right) = \mathbf{g}_c\left(\mathcal{H}(t), \mathcal{X}(t)\right) \odot \left[f_{\theta_c}^c(\mathcal{H}(t), \mathcal{X}(t)) - \mathcal{C}(t)\right],$$
$$\frac{d\mathcal{H}(t)}{dt} = \psi_\mathcal{H}\left(\mathcal{C}(t), \mathcal{H}(t), \mathcal{X}(t)\right) = \mathbf{g}_h\left(\mathcal{H}(t), \mathcal{X}(t)\right) \odot \left[f_{\theta_h}^h(\mathcal{C}(t), \mathcal{X}(t)) - \mathcal{H}(t)\right].$$

(21)

The gating functions $\mathbf{g}_c$, $\mathbf{g}_h$, and update functions $f_{\theta_c}^c$, $f_{\theta_h}^h$ are defined based on convolutional operation:

$$\mathbf{g}_c\left(\mathcal{H}, \mathcal{X}\right) = \sigma\left(\mathbf{W}_{xt} * \mathcal{X} + \mathbf{W}_{ht} * \mathcal{H}\right),$$
$$\mathbf{g}_h\left(\mathcal{H}, \mathcal{X}\right) = \sigma\left(\mathbf{W}_{x\bar{t}} * \mathcal{X} + \mathbf{W}_{h\bar{t}} * \mathcal{H}\right),$$
$$f_{\theta_c}^c(\mathcal{H}, \mathcal{X}) = tanh\left(\mathbf{W}_{hc} * \mathcal{H} + \mathbf{W}_{xc} * \mathcal{X}\right),$$
$$f_{\theta_h}^h(\mathcal{C}, \mathcal{X}) = tanh\left(\mathbf{W}_{ch} * \mathcal{C} + \mathbf{W}_{xh} * \mathcal{X}\right).$$

(22)

In this notation, $\mathcal{H}(t)$ and $\mathcal{C}(t)$ denote the slow and fast evolving hidden states in latent space $\mathbb{R}^{r \times p \times q}$ respectively. $\mathcal{X}(t) \in \mathbb{R}^{c \times h \times w}$ represents a three-dimensional input tensor. $\mathbf{W}_{\cdot\cdot}$ denotes the convolutional kernels, $\odot$ represents the Hadamard product, and $*$ indicates the convolutional operator. For brevity, bias vectors are omitted in gating and updated functions defined in 22.

We utilize the Implicit-Explicit (IMEX) time-stepping scheme to write the ODEs in Eq. (21) in a discretized formula, with subscript n as time steps index ranging from 1 to N. Given $\Delta t > 0$:

$$\frac{\mathcal{C}_n - \mathcal{C}_{n-1}}{\Delta t} = \psi_\mathcal{C}\left(\mathcal{C}_{n-1}, \mathcal{H}_{n-1}, \mathcal{X}_n\right) = \mathbf{g}_c\left(\mathcal{H}_{n-1}, \mathcal{X}_n\right) \odot \left[f_{\theta_c}^c(\mathcal{H}_{n-1}, \mathcal{X}_n) - \mathcal{C}_{n-1}\right],$$
$$\frac{\mathcal{H}_n - \mathcal{H}_{n-1}}{\Delta t} = \psi_\mathcal{H}\left(\mathcal{C}_n, \mathcal{H}_{n-1}, \mathcal{X}_n\right) = \mathbf{g}_h\left(\mathcal{H}_{n-1}, \mathcal{X}_n\right) \odot \left[f_{\theta_h}^h(\mathcal{C}_n, \mathcal{X}_n) - \mathcal{H}_{n-1}\right].$$

(23)

For discretized fast hidden state $\mathcal{C}_n$, we have:

$$\mathcal{C}_n - \mathcal{C}_{n-1} = \Delta t \cdot \mathbf{g}_c \odot (f_{\theta_c}^c - \mathcal{C}_{n-1});$$
$$\mathcal{C}_n = (\Delta t \cdot \mathbf{g}_c) \odot f_{\theta_c}^c + \mathcal{C}_{n-1} - (\Delta t \cdot \mathbf{g}_c) \odot \mathcal{C}_{n-1}$$
$$= (\Delta t \cdot \mathbf{g}_c) \odot f_{\theta_c}^c + \mathbb{1} \odot \mathcal{C}_{n-1} - (\Delta t \cdot \mathbf{g}_c) \odot \mathcal{C}_{n-1}$$
$$= (\Delta t \cdot \mathbf{g}_c) \odot f_{\theta_c}^c + (\mathbb{1} - \Delta t \cdot \mathbf{g}_c) \odot \mathcal{C}_{n-1},$$

where $\mathbb{1}$ is the matrix of ones that matches the shape of hidden state $\mathcal{C}_n$ and $\mathcal{H}_n$.

Similarly, we have for $\mathcal{H}_n$:

$$\mathcal{H}_n = (\Delta t \cdot \mathbf{g}_h) \odot f_{\theta_h}^h + (\mathbb{1} - \Delta t \cdot \mathbf{g}_h) \odot \mathcal{H}_{n-1}.$$

Define $\boldsymbol{\Delta t}_n = \Delta t \, \mathbf{g}_c$, $\overline{\boldsymbol{\Delta t}_n} = \Delta t \, \mathbf{g}_h$. By plugging in 22 and 23, we derive the discretized formula for ConvLEM:

$$
\begin{aligned}
\boldsymbol{\Delta t}_n &= \Delta t \, \mathbf{g}_c(\mathcal{H}_{n-1}, \mathcal{X}_n) \\
\overline{\boldsymbol{\Delta t}_n} &= \Delta t \, \mathbf{g}_h(\mathcal{H}_{n-1}, \mathcal{X}_n) \\
\mathcal{C}_n &= (\mathbb{1} - \boldsymbol{\Delta t}_n) \odot \mathcal{C}_{n-1} + \boldsymbol{\Delta t}_n \odot f_{\theta_c}^c(\mathcal{H}_{n-1}, \mathcal{X}_n) \\
\mathcal{H}_n &= \left(\mathbb{1} - \overline{\boldsymbol{\Delta t}_n}\right) \odot \mathcal{H}_{n-1} + \overline{\boldsymbol{\Delta t}_n} \odot f_{\theta_h}^h(\mathcal{C}_n, \mathcal{X}_n). \quad \square
\end{aligned}
\tag{24}
$$

B.2 Gating Function Distribution

We visualize the distribution of $\boldsymbol{\Delta t}$ and $\overline{\boldsymbol{\Delta t}}$ for the encoder ConvLEM cells in WaveCastNet on point-source small earthquakes in Figure B.1. Here, we set the time step factor Δt in 24 to 1, so $\boldsymbol{\Delta t}$ and $\overline{\boldsymbol{\Delta t}}$ equal to the gating functions $\mathbf{g}_c$ and $\mathbf{g}_h$ for the fast and slow hidden states $\mathcal{C}(t)$ and $\mathcal{H}(t)$, respectively.

As shown in Figure B.1, the observed occurrences of $\boldsymbol{\Delta t}$ and $\overline{\boldsymbol{\Delta t}}$ at each scale decays as a power law with respect to scale amplitude [50].

Figure B.1: Histogram of $\boldsymbol{\Delta t}$ and $\overline{\boldsymbol{\Delta t}}$ for the encoder ConvLEM cells in WaveCastNet.

By setting all axes to log scale, we can observe the different linear slopes and amplitude ranges for $\boldsymbol{\Delta t}$ and $\overline{\boldsymbol{\Delta t}}$. $\overline{\boldsymbol{\Delta t}}$ exhibits a smaller linear slope and longer trailing tail, with distribution at the amplitude closer to 0 compared to $\boldsymbol{\Delta t}$, enabling $\mathcal{H}(t)$ to better capture low-frequency features. In contrast, $\boldsymbol{\Delta t}$ is more centrally distributed near 1, showing a smaller amplitude range and a larger linear slope, reflecting the rapid change of hidden state $\mathcal{C}(t)$. These observations prove that the temporal multiscale resolution structure of ConvLEM is essential for modeling the fast-slow dynamical pattern in ground motion data.

B.3 Structure of Embedding and Reconstruction Layers

Here we discuss the embedding and reconstruction layers.

The embedding layer for densely and regularly sampled inputs $x \in \mathbb{R}^{3 \times 344 \times 224}$ is composed of three cascaded encoder layers. A standard encoder layer comprises a convolutional layer, with kernel size $=(4, 4)$, stride$=2$, padding$=1$, followed by a LeakyRelu activation layer and BatchNorm layer. Each encoder layer reduces the input spatial dimensions by a factor of 2. After the input signal is passed through three encoder layers, the dimensions change from $3 \times 344 \times 224$ to a fixed-size latent space of $144 \times 43 \times 28$. The channel transformation process is illustrated in Figure B.2.

The embedding layer for sparsely and irregularly sampled data maps the inputs $x \in \mathbb{R}^{3 \times 564}$ to a latent space of dimension $144 \times 43 \times 28$. Specifically, this embedding layer uses a shallow multi-layer feed forward network [18], followed by two convolutional layers, as illustrated in Figure B.2.

The reconstruction layer retrieves the predicted wavefield snapshot, shaped $144 \times 43 \times 28$, from the latent space. This process involves increasing the spatial dimensions by a factor of 2 through transposed convolution, followed by a PixelShuffle layer [51] to further upscale the output by a factor of 4. The dimensional transformation process is depicted in Figure B.2.
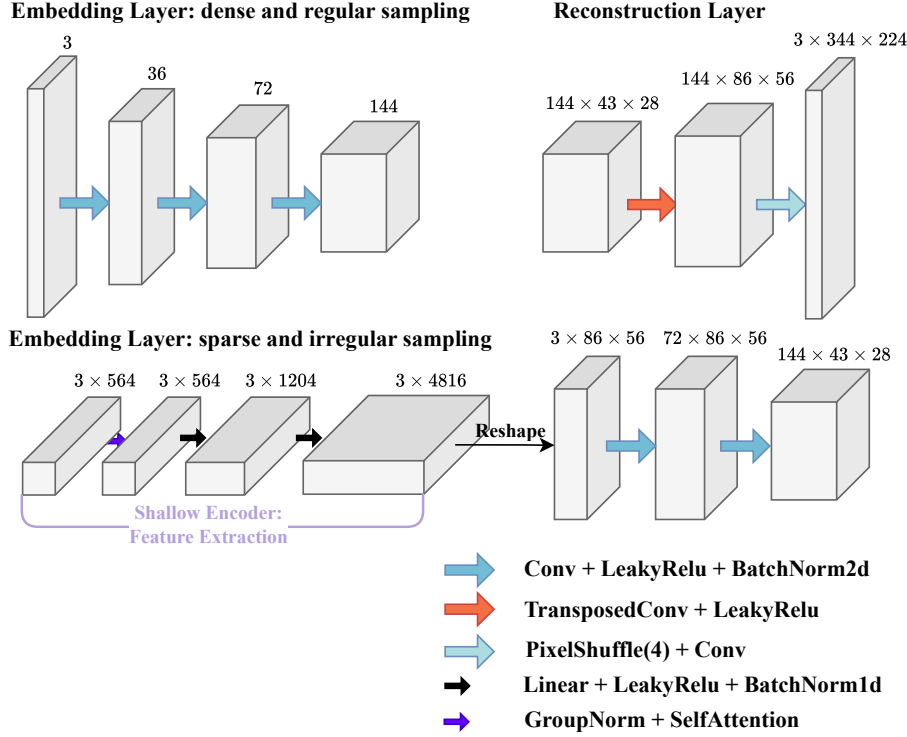

Embedding Layer: dense and regular sampling

Reconstruction Layer

Embedding Layer: sparse and irregular sampling

Shallow Encoder: Feature Extraction

Reshape

- Conv + LeakyRelu + BatchNorm2d
- TransposedConv + LeakyRelu
- PixelShuffle(4) + Conv
- Linear + LeakyRelu + BatchNorm1d
- GroupNorm + SelfAttention

Figure B.2: Detailed structure for the embedding layers and reconstruction layer in dense and sparse sampling scenarios.

B.4 Other Related Methods

We implemented ConvLSTM and ConvGRU with peephole connections as follows. For brevity, bias vectors are omitted from the activation and gating functions.

ConvLSTM

$$
\begin{aligned}
\mathbf{i}_n &= \sigma \left(\mathbf{W}_{xi} * \mathcal{X}_n + \mathbf{W}_{hi} * \mathcal{H}_{n-1} + W_{ci} \odot \mathcal{C}_{n-1} \right), \\
\mathbf{f}_n &= \sigma \left(\mathbf{W}_{xf} * \mathcal{X}_n + \mathbf{W}_{hf} * \mathcal{H}_{n-1} + \mathbf{W}_{cf} \odot \mathcal{C}_{n-1} \right), \\
\mathcal{C}_n &= \mathbf{f}_n \odot \mathcal{C}_{n-1} + \mathbf{i}_n \odot f \left(\mathbf{W}_{xc} * \mathcal{X}_n + \mathbf{W}_{hc} * \mathcal{H}_{n-1} \right), \\
\mathbf{o}_n &= \sigma \left(\mathbf{W}_{xo} * \mathcal{X}_n + \mathbf{W}_{ho} * \mathcal{H}_{n-1} + \mathbf{W}_{co} \odot \mathcal{C}_n \right), \\
\mathcal{H}_n &= \mathbf{o}_n \odot f \left(\mathcal{C}_n \right).
\end{aligned}
$$

ConvGRU

$$
\begin{aligned}
\mathbf{Z}_n &= \sigma \left(\mathbf{W}_{xz} * \mathcal{X}_n + \mathbf{W}_{hz} * \mathcal{H}_{n-1} \right), \\
\mathbf{R}_n &= \sigma \left(\mathbf{W}_{xr} * \mathcal{X}_n + \mathbf{W}_{hr} * \mathcal{H}_{n-1} \right), \\
\mathbf{o}_n &= f \left(\mathbf{W}_{xo} * \mathcal{X}_n + \mathbf{R}_n \odot \left(\mathbf{W}_{ho} * \mathcal{H}_{n-1} \right) \right), \\
\mathcal{H}_n &= (1 - \mathbf{Z}_n) \odot \mathcal{H}_{n-1} + \mathbf{Z}_n \odot \mathbf{o}_n.
\end{aligned}
$$

C Additional Results

C.1 Moving MNIST

Here, we show experiments for the MovingMNIST dataset to further demonstrate the ConvLEM's performance in spatio-temporal forecasting. The MovingMNIST dataset [53] is a well-established benchmark for video prediction and spatiotemporal modeling tasks. This dataset consists of a total of $10,000$ videos, each comprising 20 fixed-size frames with dimensions of $1 \times 64 \times 64$ pixels. Each video sequence features two handwritten digits selected from the original MNIST dataset, which move within the frame at various speeds and directions. The digits exhibit diverse velocities and trajectories, including linear motion, bouncing off the frame edges, and occasional overlap, presenting a complex and

challenging scenario for spatio-temporal forecasting models. The diversity in motion patterns makes the MovingMNIST dataset an ideal benchmark for evaluating the ability of models to capture and predict dynamic changes over time.

We use the first 10 frames as input to predict the subsequent 10 frames. The models consist of 3 stacked recurrent layers with no embedding or reconstruction layers involved. Table C.1 shows the results for this task. While using fewer parameters, ConvLEM is able to outperform the prediction performance of the ConvLSTM model. This further demonstrates ConvLEM's potential for spatio-temporal forecasting tasks.

Figure C.1: An example on MovingMnist dataset.

Model	Parameters	Latent Space	Layers	BCELoss $\downarrow$
Stacked ConvLSTM	3.10M	(64, 64, 64)	3	206.13
Stacked ConvLEM	2.31M	(64, 64, 64)	3	166.75

Table C.1: Results for Moving Mnist. The ConvLEM demonstrates improved forecasting capabilities while requiring fewer parameters than ConvLSTM.

D Supplementary figures

- Figure D.1: Kinematic rupture model of the M4.5 earthquake

- Figure D.2: Kinematic rupture model of the M5 earthquake

- Figure D.3: Kinematic rupture model of the M5.5 earthquake

- Figure D.4: Kinematic rupture model of the M6 earthquake

- Figure D.5: Kinematic rupture model of the M6.5 earthquake

- Figure D.6: Kinematic rupture model of the M7earthquake

m5.00–3.4x3.0_s500–Hayward_scor0.94_vr0.8_dh1.0

m4.50–1.8x1.8_s500–Hayward_scor0.94_vr0.8_dh0.4
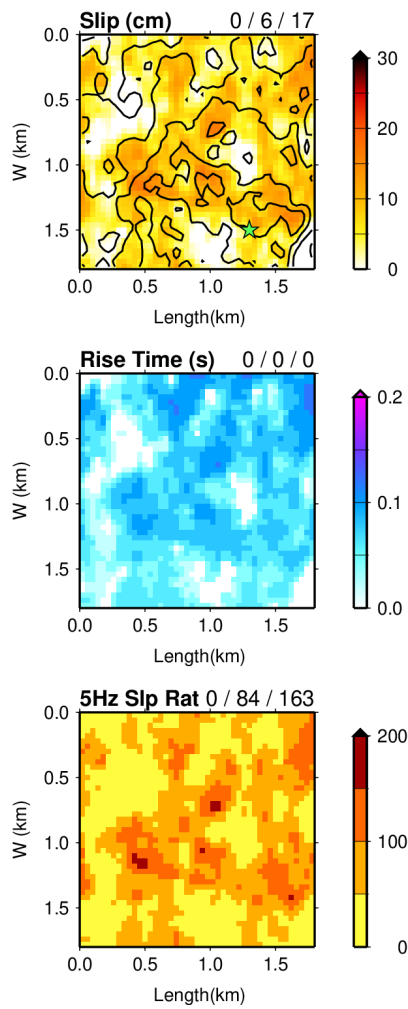

Figure D.1: Kinematic rupture models for the M4.5 earthquake. (Top) slip (middle) rise time and (bottom) 5 Hz slip rate distributions.
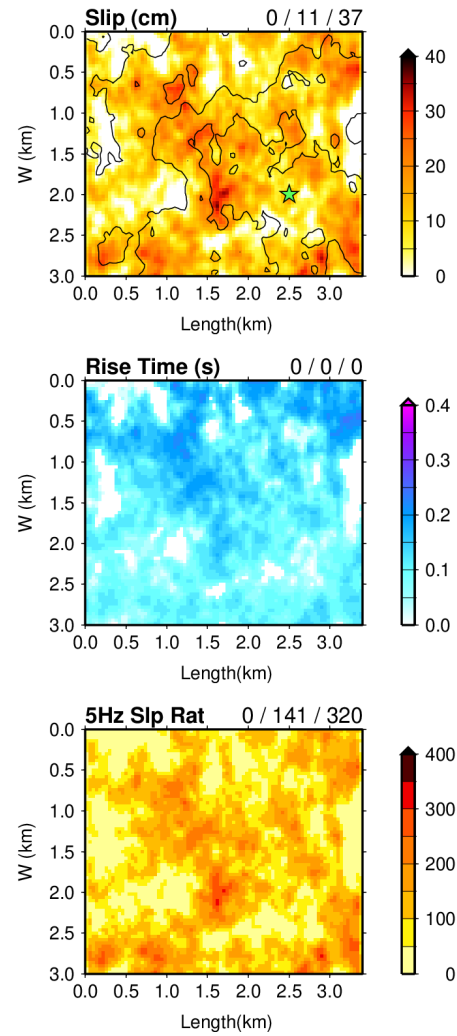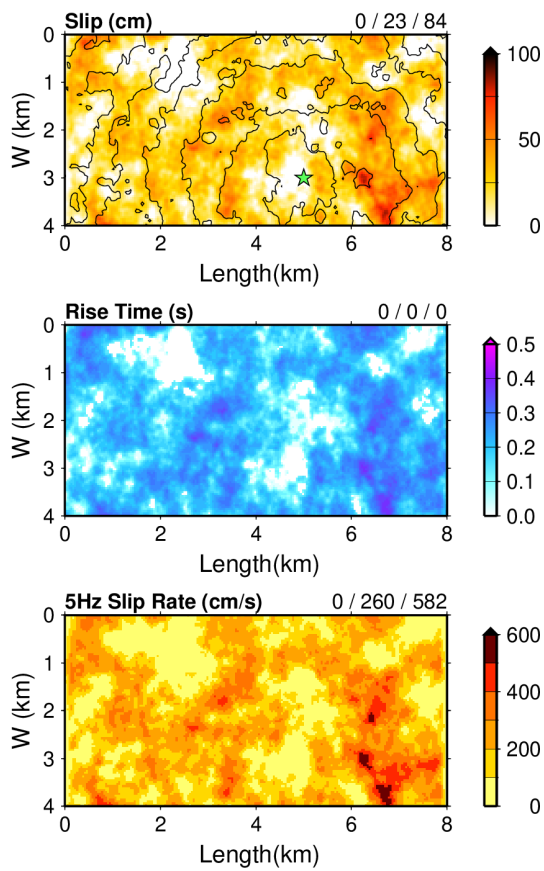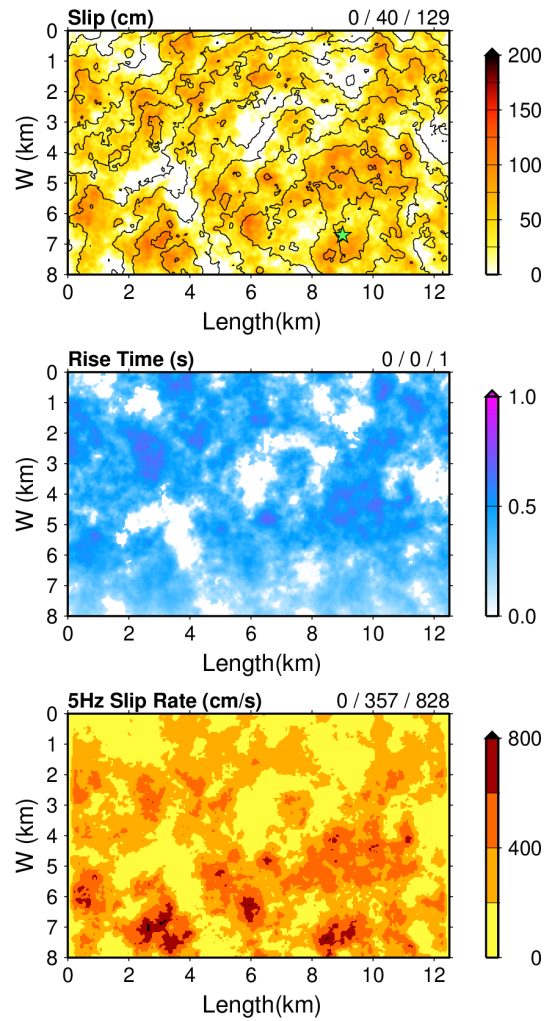
Figure D.2: Kinematic rupture models for the M5.0 earthquake. (Top) slip (middle) rise time and (bottom) 5 Hz slip rate distributions.

m6.00–12.5x8.0_s100–Hayward_scor0.95_vr0.8_dh3.0
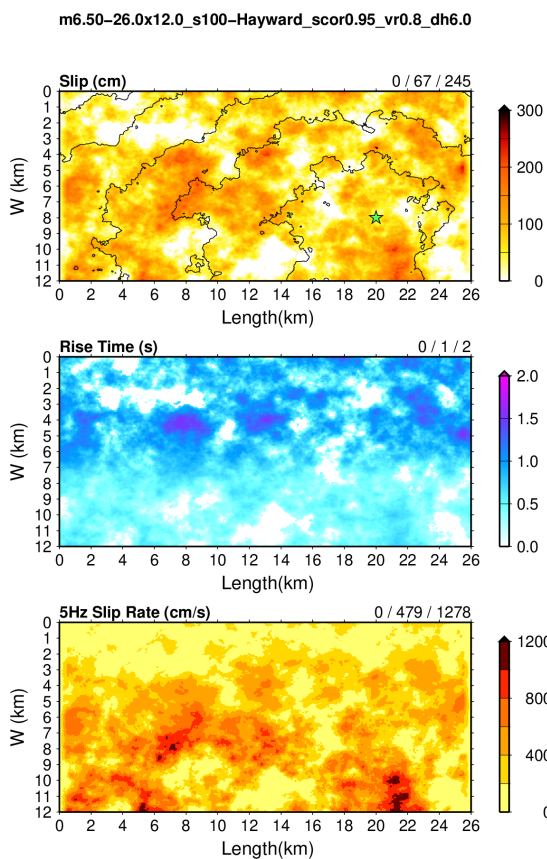
m5.50–8.0x4.0_s600–Hayward_scor0.94_vr0.8_dh1.0

Figure D.3: Kinematic rupture models for the M5.5 earthquake. (Top) slip (middle) rise time and (bottom) 5 Hz slip rate distributions.

Figure D.4: Kinematic rupture models for the M6 earthquake. (Top) slip (middle) rise time and (bottom) 5 Hz slip rate distributions.

Figure D.5: Kinematic rupture models for the M6.5 earthquake. (Top) slip (middle) rise time and (bottom) 5 Hz slip rate distributions.
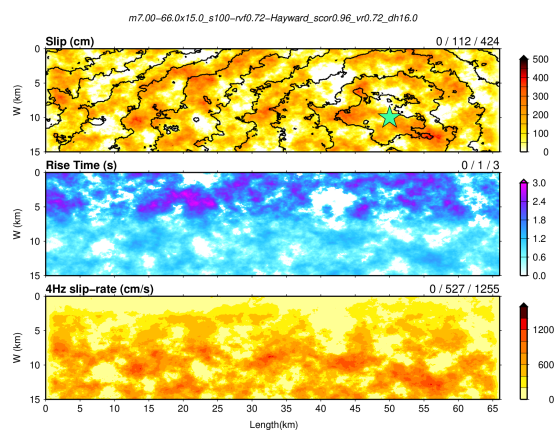
Figure D.6: Kinematic rupture models for the M7 earthquake.(Top) slip (middle) rise time and (bottom) 5 Hz slip rate distributions.

21